

```
In [1]: import pandas as pd
# Load the dataset
work = pd.read_csv("Impact_of_Remote_Work_on_Mental_Health.csv")
```

pip install pandas openpyxl

import pandas as pd

Load the Excel dataset work = pd.read_excel("Nithi_internship_Companies.xlsx")

Display the first few rows of the DataFrame to verify loading print(work.head())

```
In [2]: work.head()
```

```
Out[2]:
```

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

```
In [3]: work.head(2)
```

```
Out[3]:
```

	Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location
0	EMP0001	32	Non-binary	HR	Healthcare	13	Hybrid
1	EMP0002	40	Female	Data Scientist	IT	3	Remote

```
In [4]: print(work.columns)
```

```
Index(['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',
      'Company_Support_for_Remote_Work', 'Physical_Activity', 'Sleep_Quality',
      'Region'],
      dtype='object')
```

```
In [5]: # Display the shape of the DataFrame (rows, columns)
print(work.shape)

(5000, 20)
```

```
In [6]: #check null value
work.isnull().sum()
```

```
Out[6]: 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
```

```
In [7]: # Fill missing values with mode (most frequent value) in 'Mental_Health_Condition'
work['Mental_Health_Condition'].fillna(work['Mental_Health_Condition'].mode()[0], inplace=True)
work['Physical_Activity'].fillna(work['Physical_Activity'].mode()[0], inplace=True)
```

```
In [8]: #check null value
work.isnull().sum()
```

```
Out[8]: 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 [ ]:
```

```
pip install matplotlib seaborn
```

```
In [9]: import matplotlib.pyplot as plt
import seaborn as sns
```

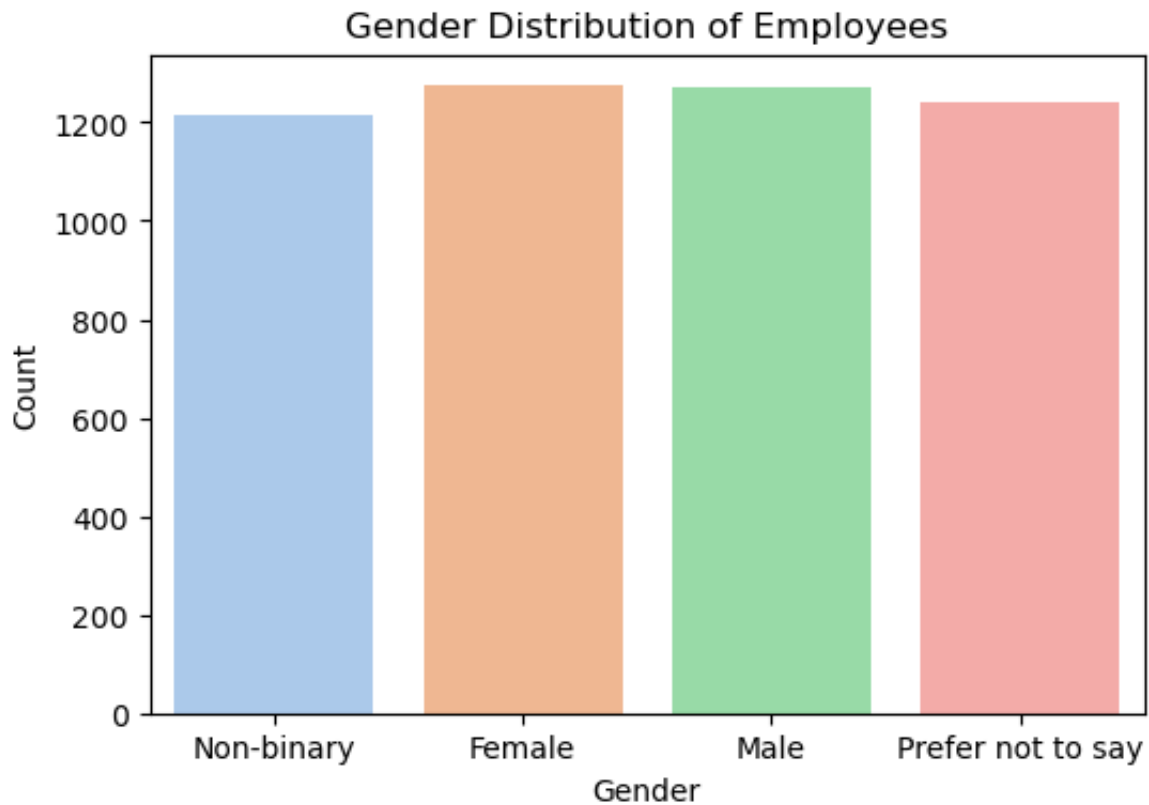
```
In [ ]:
```

```
In [10]: import warnings
warnings.filterwarnings('ignore')
```

Bar Chart – Gender Distribution

```
In [11]: # Count of employees by gender
plt.figure(figsize=(6, 4))
sns.countplot(x='Gender', data=work, palette='pastel')

# Title and labels
plt.title('Gender Distribution of Employees')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



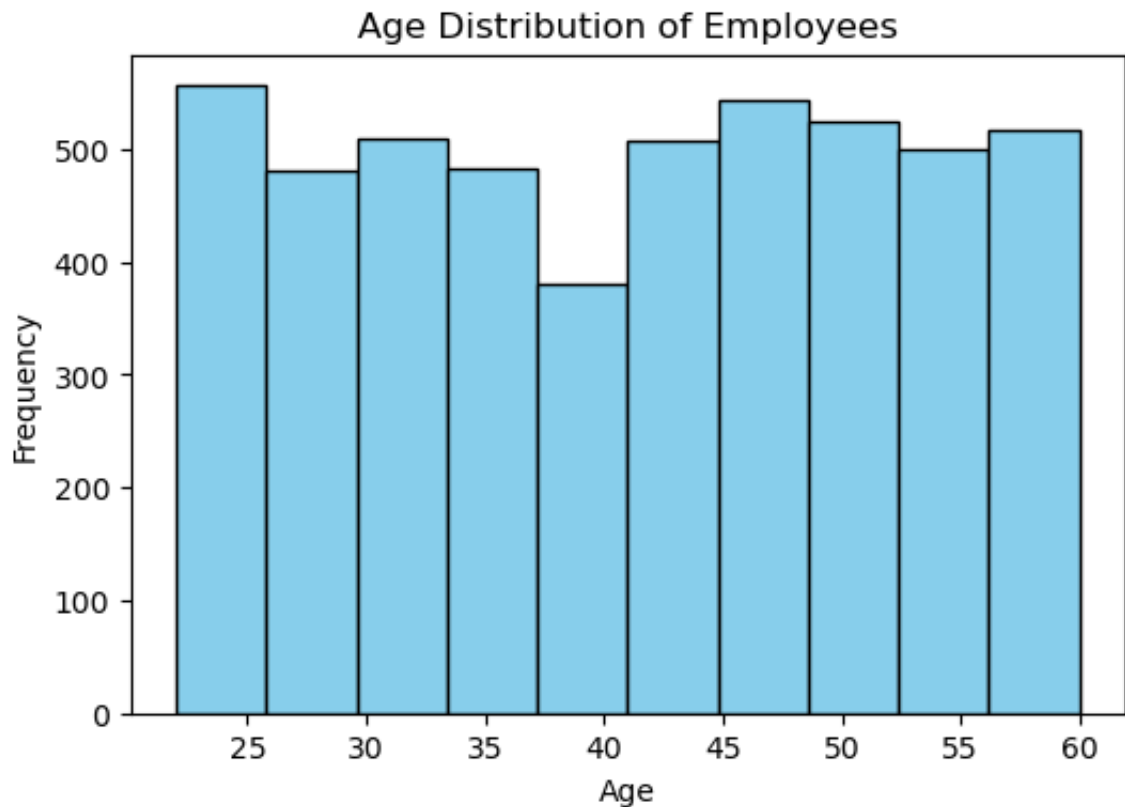
This bar chart visualizes the distribution of employees by gender. It helps to see the number of employees in each gender category

In []:

Histogram – Age Distribution

```
In [12]: # Histogram of age distribution
plt.figure(figsize=(6, 4))
plt.hist(work['Age'], bins=10, color='skyblue', edgecolor='black')

# Title and labels
plt.title('Age Distribution of Employees')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



This histogram shows the distribution of employees' ages, making it easy to spot common age ranges among employees.

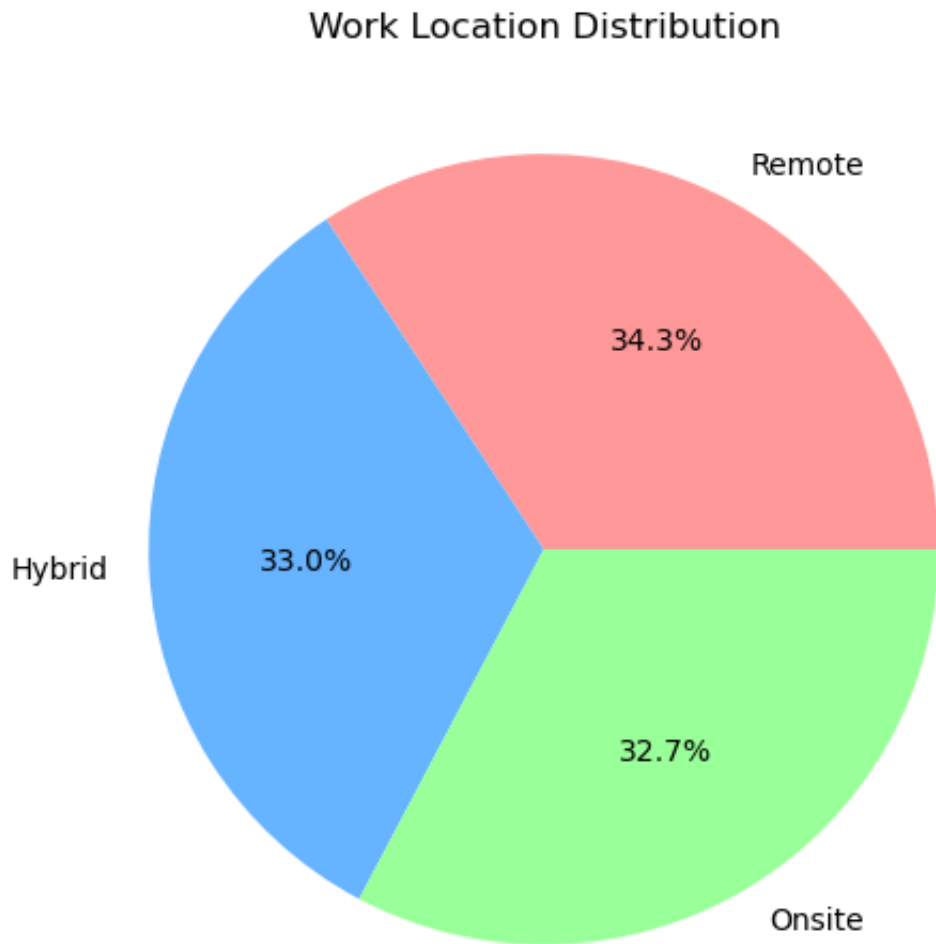
In []:

Pie Chart – Work Location Distribution

```
In [13]: # Pie chart of work location distribution
work_location_counts = work['Work_Location'].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(work_location_counts, labels=work_location_counts.index, autopct=

# Title
plt.title('Work Location Distribution')
plt.show()
```



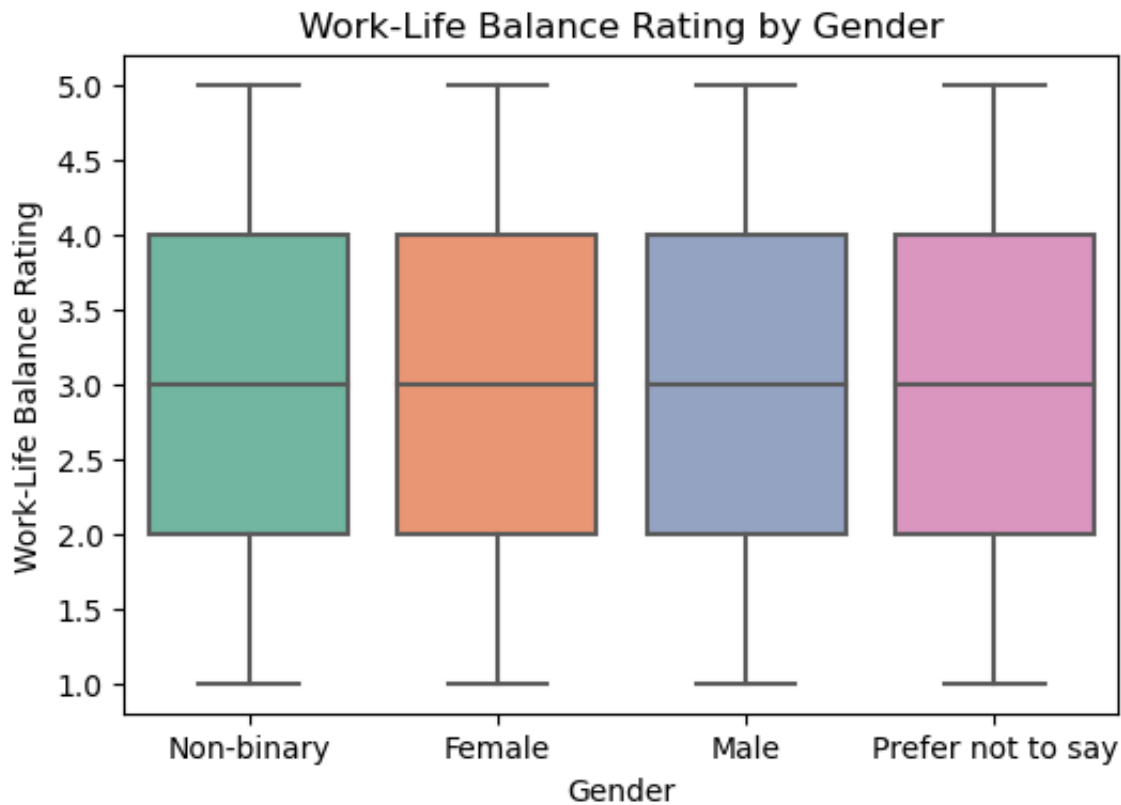
The pie chart shows the percentage of employees working remotely, in a hybrid setup, or on-site. This gives a quick glance at how work is distributed.

In []:

Boxplot – Work-Life Balance Rating by Gender

```
In [14]: # Boxplot of work-life balance rating by gender
plt.figure(figsize=(6, 4))
sns.boxplot(x='Gender', y='Work_Life_Balance_Rating', data=work, palette=

# Title and labels
plt.title('Work-Life Balance Rating by Gender')
plt.xlabel('Gender')
plt.ylabel('Work-Life Balance Rating')
plt.show()
```



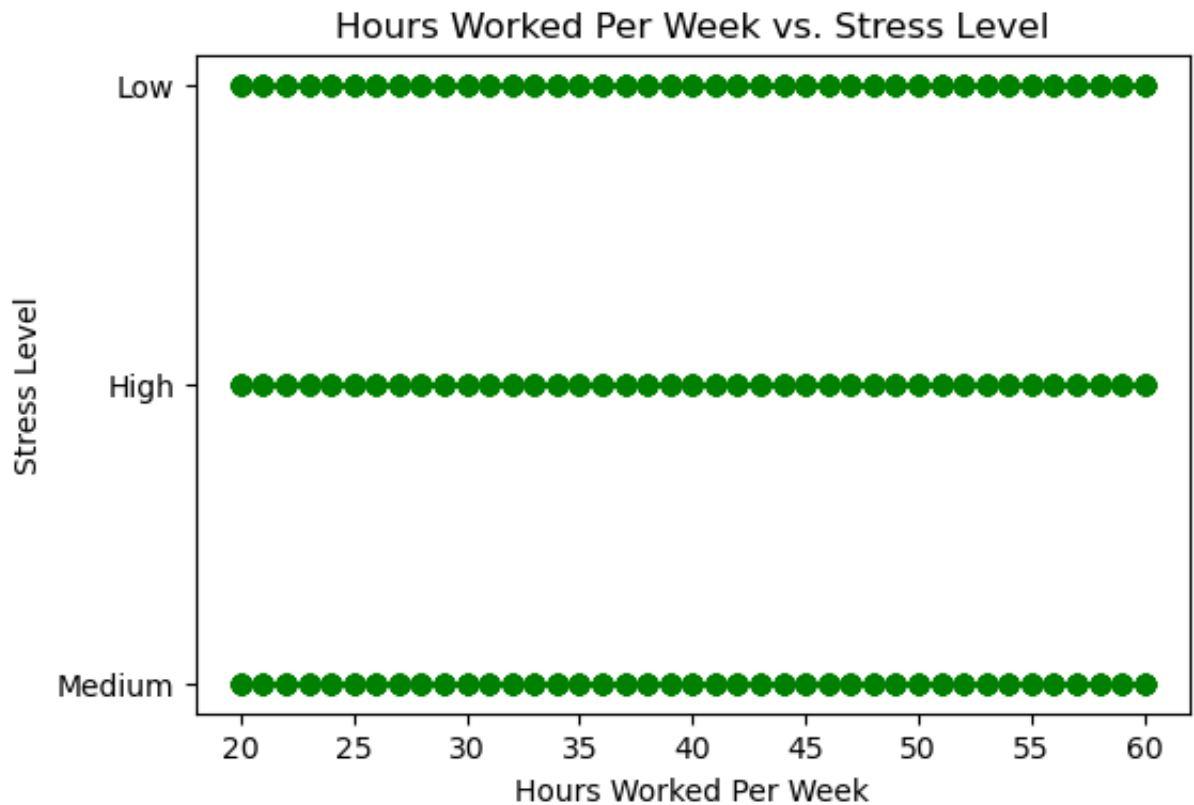
The boxplot shows how work-life balance ratings vary across different genders. The median, quartiles, and potential outliers are visualized here.

In []:

Scatter Plot – Hours Worked vs. Stress Level

```
In [15]: # Scatter plot of hours worked per week vs stress level
plt.figure(figsize=(6, 4))
plt.scatter(work['Hours_Worked_Per_Week'], work['Stress_Level'], color='g')

# Title and labels
plt.title('Hours Worked Per Week vs. Stress Level')
plt.xlabel('Hours Worked Per Week')
plt.ylabel('Stress Level')
plt.show()
```



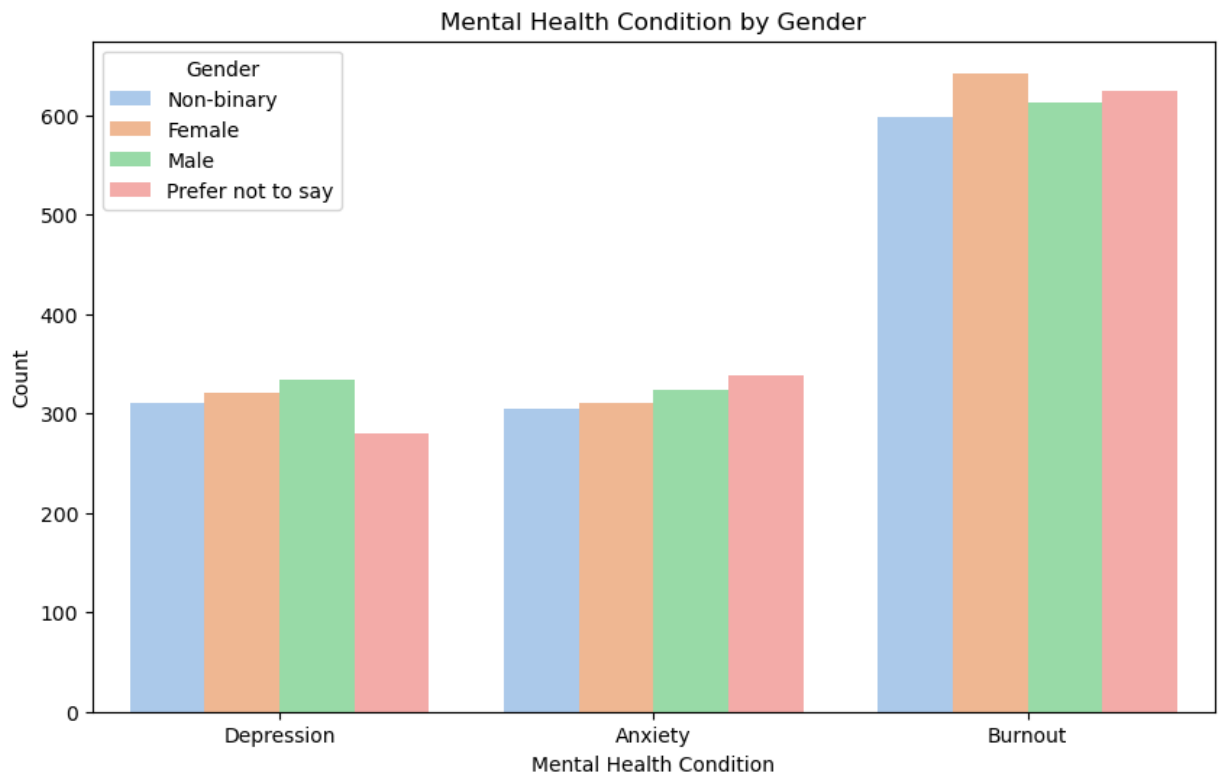
This scatter plot shows the relationship between hours worked and stress levels. It helps in understanding if working more hours tends to increase stress.

In []:

Count Plot – Mental Health Condition by Gender

```
In [16]: # Count plot of mental health condition by gender
plt.figure(figsize=(10, 6))
sns.countplot(x='Mental_Health_Condition', hue='Gender', data=work, palette='magma')

# Title and labels
plt.title('Mental Health Condition by Gender')
plt.xlabel('Mental Health Condition')
plt.ylabel('Count')
plt.legend(title='Gender')
plt.show()
```

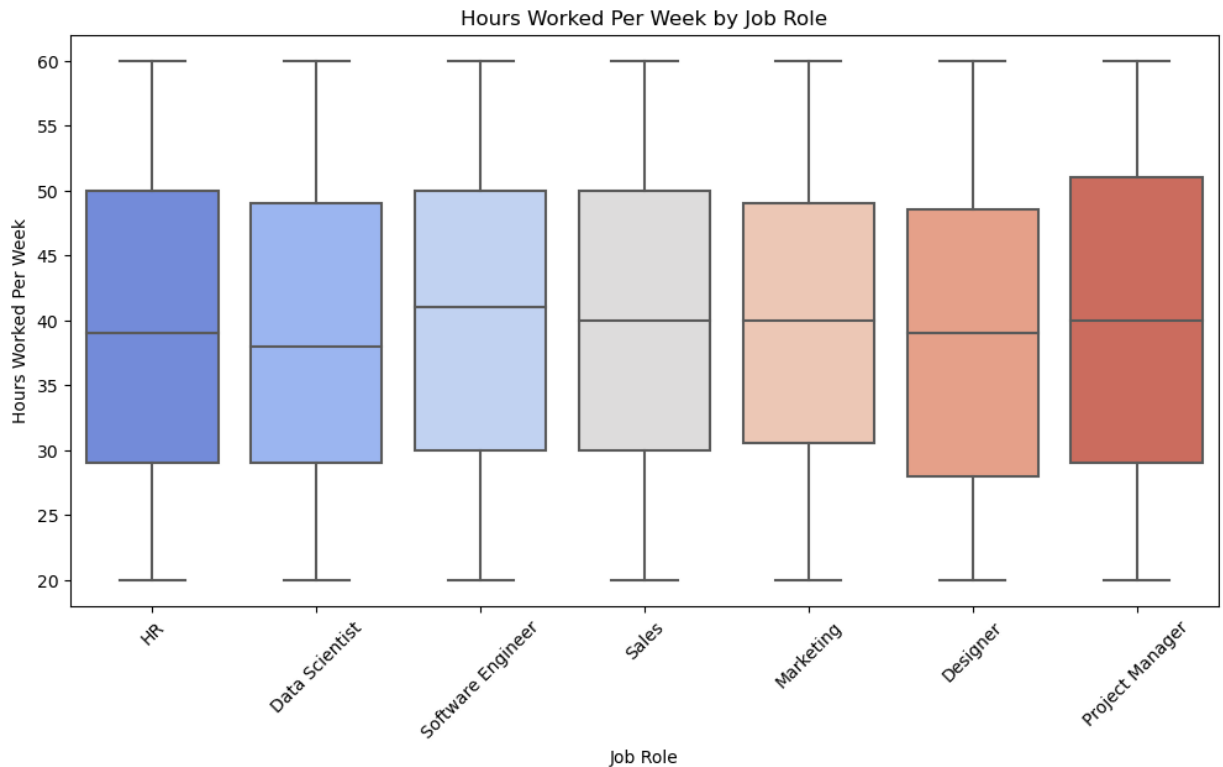
This count plot helps visualize how different mental health conditions are distributed among various genders, allowing for comparisons between groups.

In []:

Boxplot – Hours Worked per Week by Job Role

```
In [17]: # Boxplot of hours worked per week by job role
plt.figure(figsize=(12, 6))
sns.boxplot(x='Job_Role', y='Hours_Worked_Per_Week', data=work, palette='

# Title and labels
plt.title('Hours Worked Per Week by Job Role')
plt.xlabel('Job Role')
plt.ylabel('Hours Worked Per Week')
plt.xticks(rotation=45)
plt.show()
```

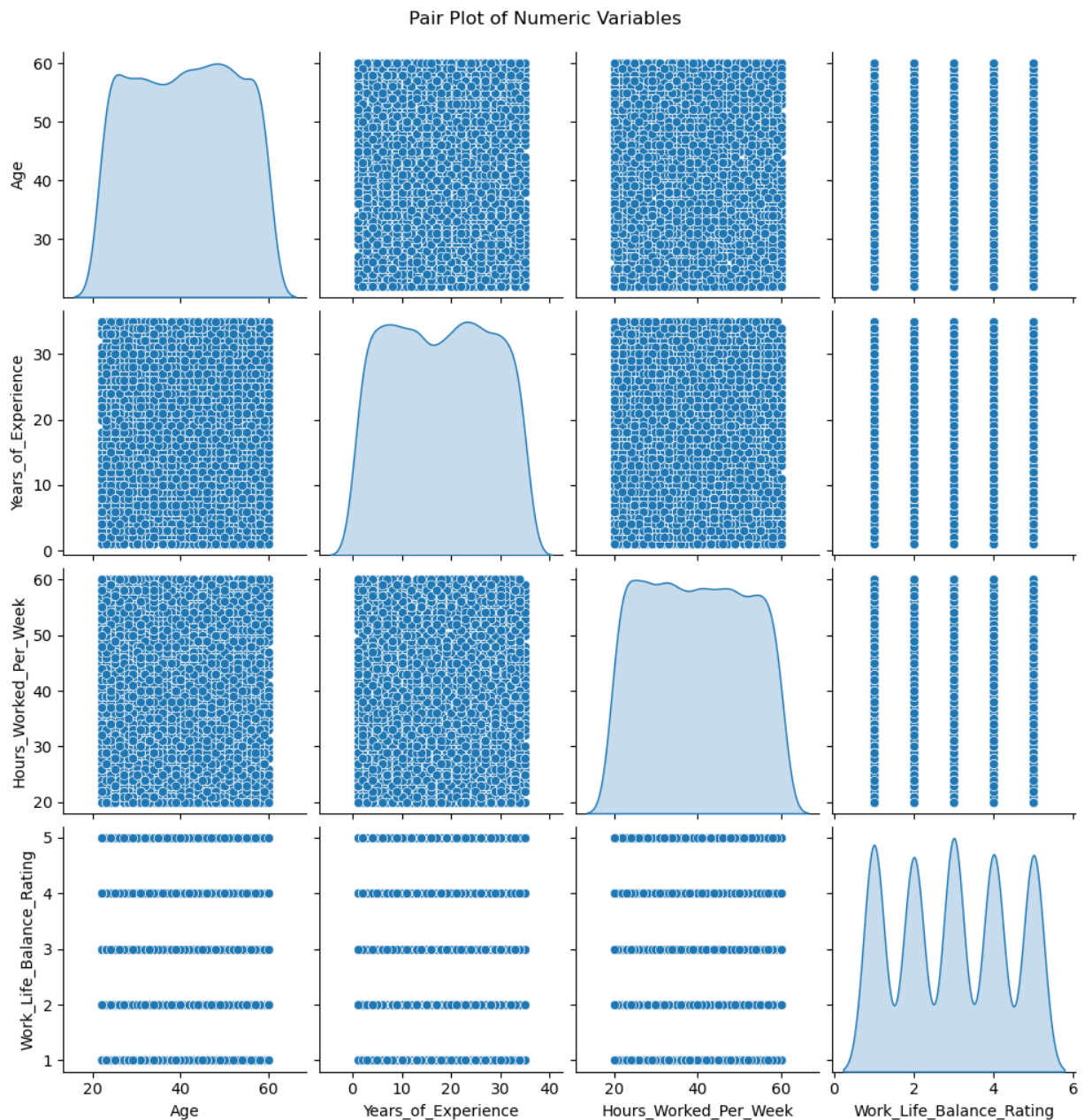


The boxplot shows the variation in hours worked per week for different job roles. It can help identify which roles are more demanding in terms of working hours.

In []:

Pair Plot – Relationships Between Multiple Numeric Variables

```
In [18]: # Pair plot of selected numeric variables
sns.pairplot(work[['Age', 'Years_of_Experience', 'Hours_Worked_Per_Week']],
plt.suptitle('Pair Plot of Numeric Variables', y=1.02) # Adjust title po
plt.show())
```



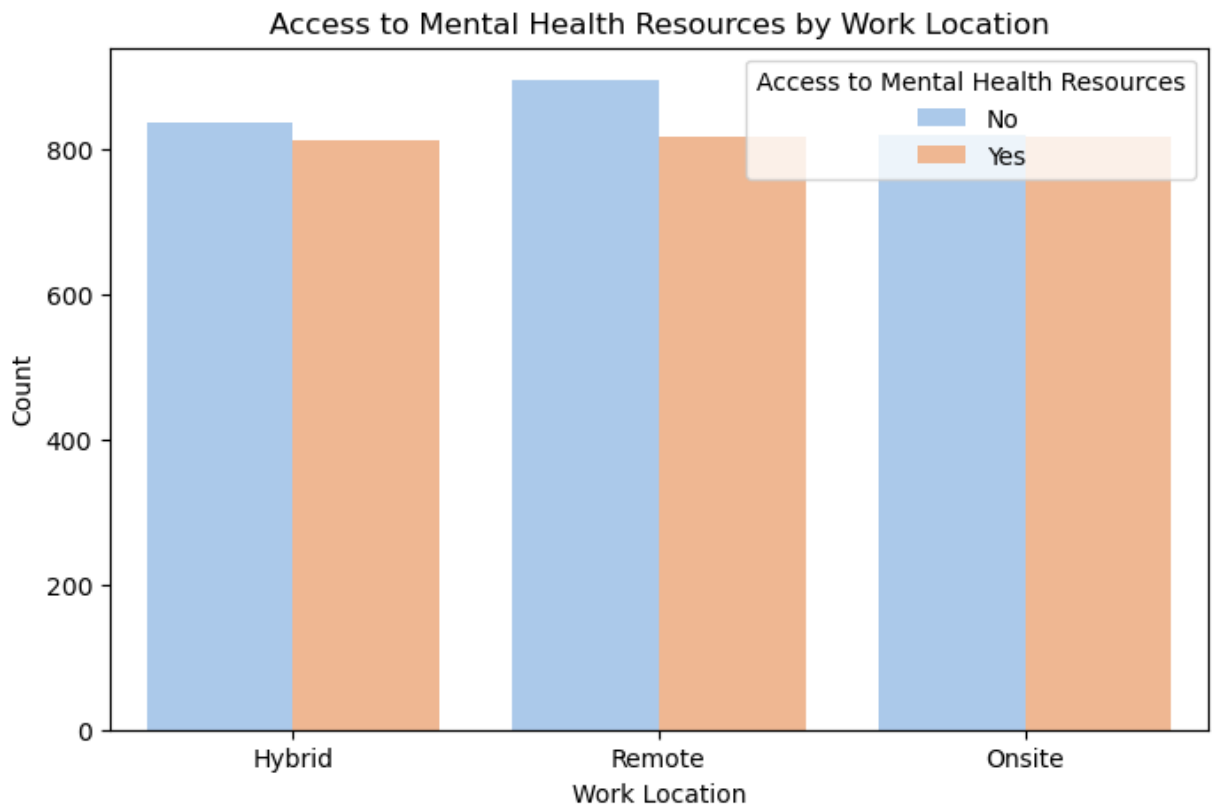
The pair plot provides a comprehensive view of relationships between multiple numeric variables. It allows you to easily identify trends and correlations among them.

In []:

Count Plot – Access to Mental Health Resources by Work Location

```
In [19]: # Count plot of access to mental health resources by work location
plt.figure(figsize=(8, 5))
sns.countplot(x='Work_Location', hue='Access_to_Mental_Health_Resources',

# Title and labels
plt.title('Access to Mental Health Resources by Work Location')
plt.xlabel('Work Location')
plt.ylabel('Count')
plt.legend(title='Access to Mental Health Resources', loc='upper right')
plt.show()
```



This count plot helps visualize how access to mental health resources varies by work location, highlighting potential areas for improvement.

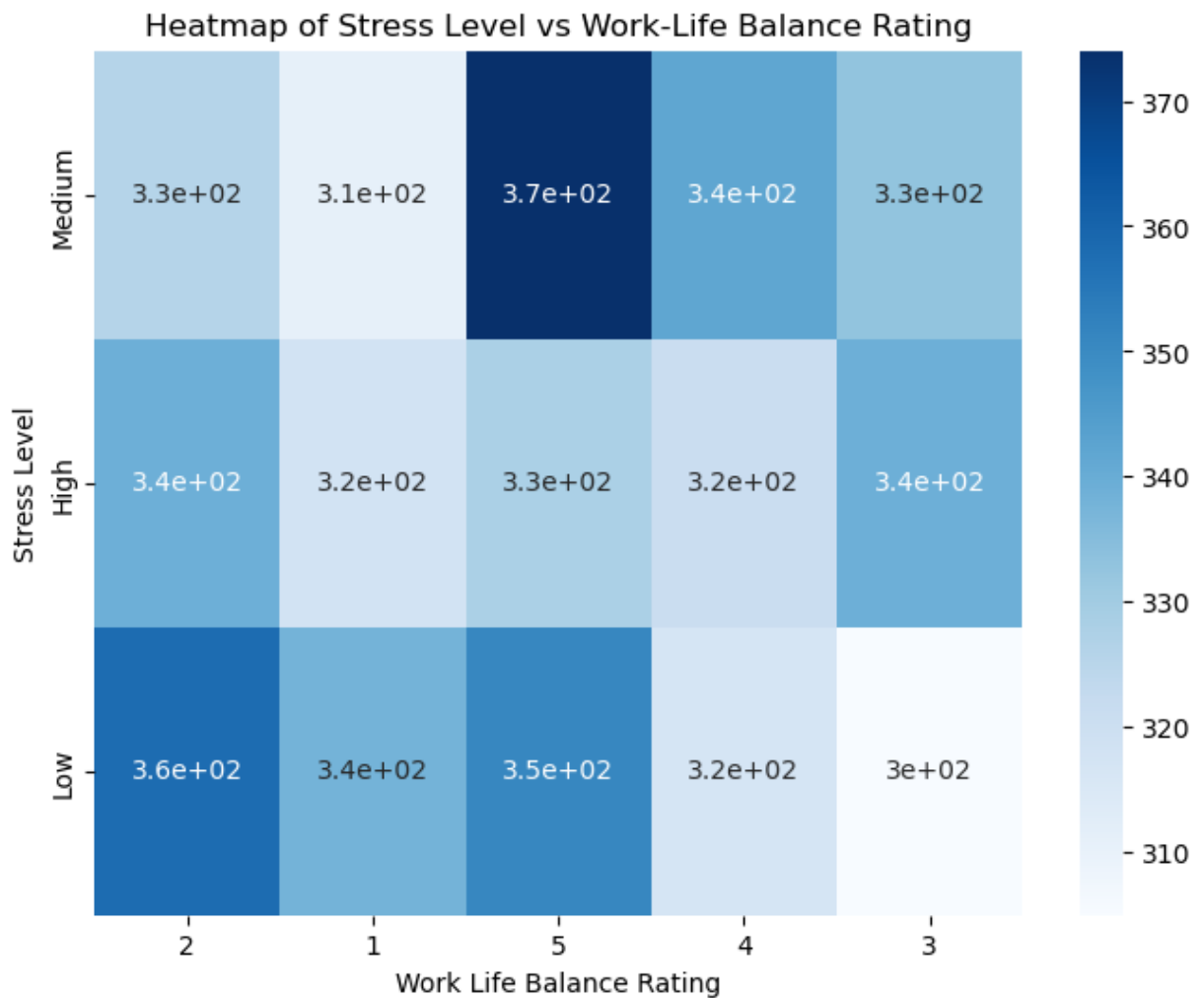
In []:

Heatmap – Stress Level vs. Work Life Balance Rating

```
In [20]: # Heatmap of stress level vs work-life balance rating
stress_work_life_corr = pd.crosstab(work['Stress_Level'], work['Work_Life
Balance_Rating'])

plt.figure(figsize=(8, 6))
sns.heatmap(stress_work_life_corr, annot=True, cmap='Blues', xticklabels=
work['Work_Life Balance_Rating'].unique(), yticklabels=work['Stress_Level'].
unique())

# Title and labels
plt.title('Heatmap of Stress Level vs Work-Life Balance Rating')
plt.xlabel('Work Life Balance Rating')
plt.ylabel('Stress Level')
plt.show()
```



This heatmap visualizes how stress levels relate to work-life balance ratings, indicating potential areas of concern where employees may feel more stress.

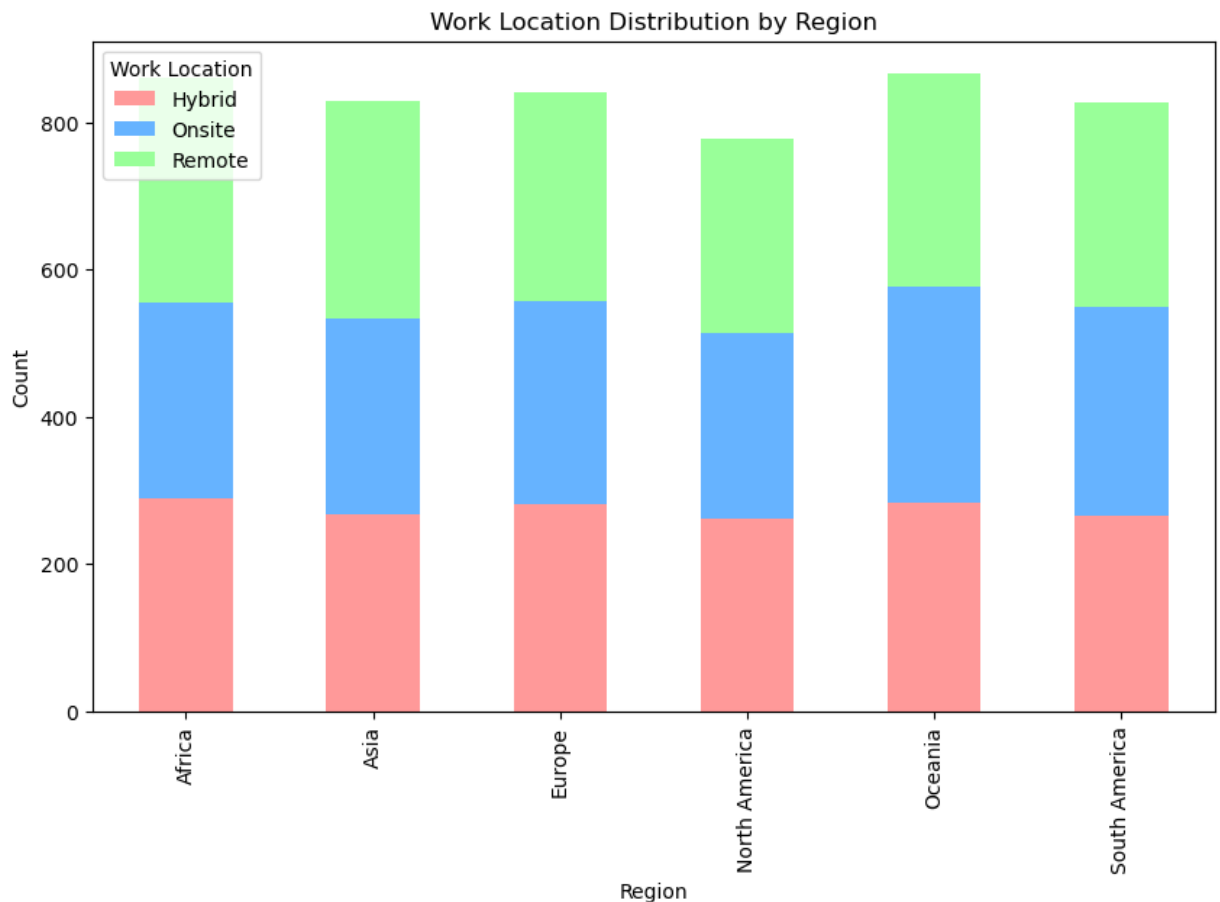
In []:

Stacked Bar Chart – Work Location by Region

```
In [21]: # Stacked bar chart of work location by region
work_location_region = pd.crosstab(work['Region'], work['Work_Location'])

work_location_region.plot(kind='bar', stacked=True, figsize=(10, 6), color=

# Title and labels
plt.title('Work Location Distribution by Region')
plt.xlabel('Region')
plt.ylabel('Count')
plt.legend(title='Work Location', loc='upper left')
plt.show()
```



This stacked bar chart helps visualize how different regions adopt various work setups (Remote, Hybrid, On-site), showing the distribution within each region.

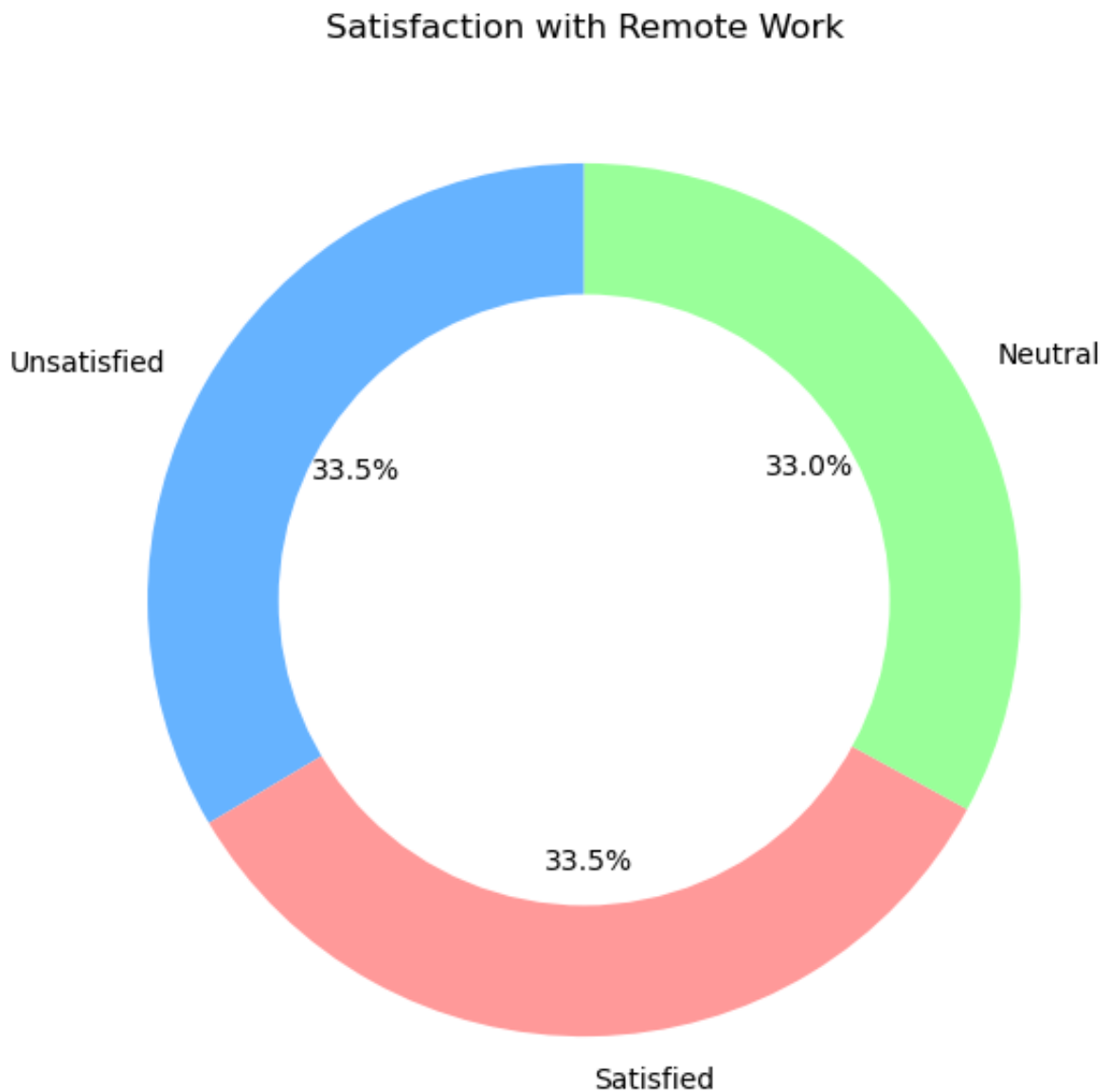
In []:

Donut Chart – Satisfaction with Remote Work

```
In [22]: # Donut chart of satisfaction with remote work
satisfaction_counts = work['Satisfaction_with_Remote_Work'].value_counts()

plt.figure(figsize=(7, 7))
plt.pie(satisfaction_counts, labels=satisfaction_counts.index, autopct='%
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
plt.gca().add_artist(centre_circle)

# Title
plt.title('Satisfaction with Remote Work')
plt.show()
```



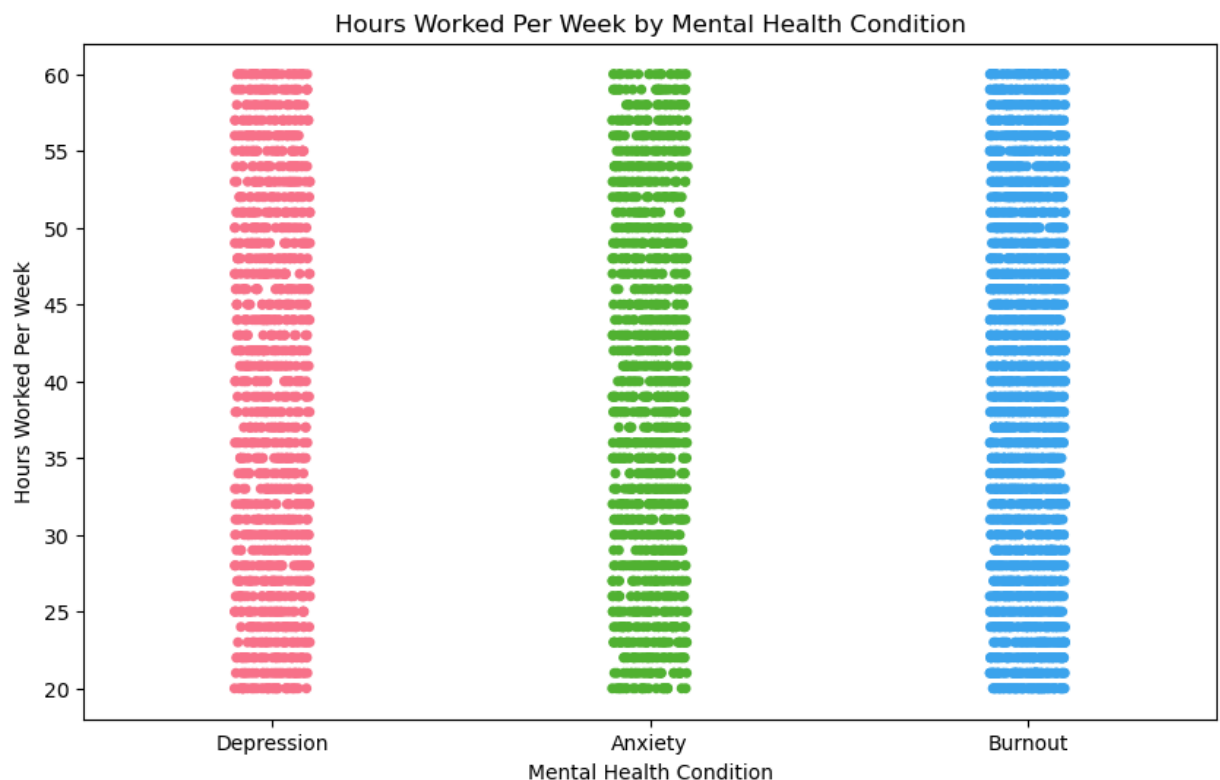
The donut chart shows the percentage of employees who are satisfied, unsatisfied, or neutral about remote work, giving a quick understanding of employee sentiment.

In []:

Strip Plot – Hours Worked per Week by Mental Health Condition

```
In [23]: # Strip plot of hours worked per week by mental health condition
plt.figure(figsize=(10, 6))
sns.stripplot(x='Mental_Health_Condition', y='Hours_Worked_Per_Week', data=

# Title and labels
plt.title('Hours Worked Per Week by Mental Health Condition')
plt.xlabel('Mental Health Condition')
plt.ylabel('Hours Worked Per Week')
plt.show()
```



The strip plot provides an overview of how many hours employees work each week, with respect to different mental health conditions. The jitter effect spreads out the data points to avoid overlap.

```
In [ ]:
```

Swarm Plot – Social Isolation Rating by Work Location

```
In [24]: pip install wordcloud
```


Requirement already satisfied: wordcloud in /opt/homebrew/anaconda3/lib/python3.11/site-packages (1.9.3)

Requirement already satisfied: numpy>=1.6.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from wordcloud) (1.23.5)

Requirement already satisfied: pillow in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from wordcloud) (9.4.0)

Requirement already satisfied: matplotlib in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from wordcloud) (3.7.2)

Requirement already satisfied: contourpy>=1.0.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (1.0.5)

Requirement already satisfied: cycler>=0.10 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (0.11.0)

Requirement already satisfied: fonttools>=4.22.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (4.25.0)

Requirement already satisfied: kiwisolver>=1.0.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (1.4.4)

Requirement already satisfied: packaging>=20.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (24.1)

Requirement already satisfied: pyparsing<3.1,>=2.3.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (3.0.9)

Requirement already satisfied: python-dateutil>=2.7 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from matplotlib->wordcloud) (2.8.2)

Requirement already satisfied: six>=1.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)

Note: you may need to restart the kernel to use updated packages.

```
In [25]: from wordcloud import WordCloud

# Generate word cloud for job roles
job_roles_text = ' '.join(work['Job_Role'].astype(str))
wordcloud = WordCloud(width=800, height=400, background_color='white').ge

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Job Roles')
plt.show()
```



The word cloud visualizes the frequency of different job roles in the dataset, where larger words indicate more frequent job roles. It gives a quick overview of the most common positions in your dataset.

In []:

In []:

Basic Dashboard

In [26]:

```
pip install dash jupyter-dash
```

```
Requirement already satisfied: dash in /opt/homebrew/anaconda3/lib/python3.11/site-packages (2.18.1)
Requirement already satisfied: jupyter-dash in /opt/homebrew/anaconda3/lib/python3.11/site-packages (0.4.2)
Requirement already satisfied: Flask<3.1,>=1.0.4 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.2.2)
Requirement already satisfied: Werkzeug<3.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.2.3)
Requirement already satisfied: plotly>=5.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (5.9.0)
Requirement already satisfied: dash-html-components==2.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.0.0)
Requirement already satisfied: dash-core-components==2.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.0.0)
Requirement already satisfied: dash-table==5.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (5.0.0)
Requirement already satisfied: importlib-metadata in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (6.0.0)
Requirement already satisfied: typing-extensions>=4.1.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (4.12.2)
```

Requirement already satisfied: requests in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.31.0)

Requirement already satisfied: retrying in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (1.3.4)

Requirement already satisfied: nest-asyncio in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (1.6.0)

Requirement already satisfied: setuptools in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (68.0.0)

Requirement already satisfied: ipython in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-dash) (8.15.0)

Requirement already satisfied: ipykernel in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-dash) (6.25.0)

Requirement already satisfied: ansi2html in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-dash) (1.9.2)

Requirement already satisfied: Jinja2>=3.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Flask<3.1,>=1.0.4->dash) (3.1.2)

Requirement already satisfied: itsdangerous>=2.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Flask<3.1,>=1.0.4->dash) (2.0.1)

Requirement already satisfied: click>=8.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Flask<3.1,>=1.0.4->dash) (8.0.4)

Requirement already satisfied: tenacity>=6.2.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from plotly>=5.0.0->dash) (8.5.0)

Requirement already satisfied: MarkupSafe>=2.1.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Werkzeug<3.1->dash) (2.1.1)

Requirement already satisfied: zipp>=0.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from importlib-metadata->dash) (3.11.0)

Requirement already satisfied: appnope in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (0.1.2)

Requirement already satisfied: comm>=0.1.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (0.1.2)

Requirement already satisfied: debugpy>=1.6.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (1.6.7)

Requirement already satisfied: jupyter-client>=6.1.12 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (7.4.9)

Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (5.3.0)

Requirement already satisfied: matplotlib-inline>=0.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (0.1.6)

Requirement already satisfied: packaging in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (24.1)

Requirement already satisfied: psutil in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (5.9.0)

Requirement already satisfied: pyzmq>=20 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (23.2.0)

Requirement already satisfied: tornado>=6.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (6.3.2)

Requirement already satisfied: traitlets>=5.4.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (5.7.1)

Requirement already satisfied: backcall in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.2.0)

Requirement already satisfied: decorator in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (5.1.1)

Requirement already satisfied: jedi>=0.16 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.18.1)

Requirement already satisfied: pickleshare in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.7.5)

Requirement already satisfied: prompt-toolkit!=3.0.37,<3.1.0,>=3.0.30 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (3.0.36)

Requirement already satisfied: pygments>=2.4.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (2.15.1)

Requirement already satisfied: stack-data in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.2.0)

Requirement already satisfied: pexpect>4.3 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (4.8.0)

Requirement already satisfied: charset-normalizer<4,>=2 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (2.2.2)

Requirement already satisfied: certifi>=2017.4.17 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (2023.7.22)

Requirement already satisfied: six>=1.7.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from retrying->dash) (1.16.0)

Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jedi>=0.16->ipython->jupyter-dash) (0.8.3)

Requirement already satisfied: entrypoints in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-client>=6.1.12->ipykernel->jupyter-dash) (0.4)

Requirement already satisfied: python-dateutil>=2.8.2 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-client>=6.1.12->ipykernel->jupyter-dash) (2.8.2)

Requirement already satisfied: platformdirs>=2.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-core!=5.0.*,>=4.12->ipykernel->jupyter-dash) (3.10.0)

Requirement already satisfied: ptyprocess>=0.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from pexpect>4.3->ipython->jupyter-dash) (0.7.0)

Requirement already satisfied: wcwidth in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from prompt-toolkit!=3.0.37,<3.1.0,>=3.0.30->ipython->jupyter-dash) (0.2.5)

Requirement already satisfied: executing in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from stack-data->ipython->jupyter-dash) (0.8.3)

Requirement already satisfied: asttokens in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from stack-data->ipython->jupyter-dash) (2.0.5)

Requirement already satisfied: pure-eval in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from stack-data->ipython->jupyter-dash) (0.2.2)

Note: you may need to restart the kernel to use updated packages.

```

In [27]: # Import necessary libraries
import pandas as pd
from jupyter_dash import JupyterDash
from dash import dcc, html
import plotly.express as px
from dash.dependencies import Input, Output

# Load the dataset
work = pd.read_csv("Impact_of_Remote_Work_on_Mental_Health.csv")

# Initialize the app
app = JupyterDash(__name__)

# Create a simple Plotly chart for the dashboard
fig = px.histogram(work, x="Stress_Level", color="Work_Location", title="")

# Define the layout of the app
app.layout = html.Div([
    html.H1("Remote Work Dashboard"),
    dcc.Graph(id='stress-levels-graph', figure=fig),

    # Dropdown for interactivity
    dcc.Dropdown(
        id='work-location-dropdown',
        options=[{'label': loc, 'value': loc} for loc in work['Work_Location'].unique()],
        value='Remote',
        style={'width': '50%'}
    ),

    # Placeholder for output graph
    dcc.Graph(id='productivity-change-graph')
])

# Create a callback for interactivity
@app.callback(
    Output('productivity-change-graph', 'figure'),
    [Input('work-location-dropdown', 'value')]
)
def update_graph(selected_location):
    filtered_data = work[work['Work_Location'] == selected_location]
    fig = px.histogram(filtered_data, x="Productivity_Change", color="Gender")
    return fig

# Run the app inside the notebook
app.run_server(mode='inline')

```

Loading...

Loading...

In []:

Using Panel for Dashboard

```
In [28]: # Import necessary libraries
import panel as pn
import pandas as pd
import plotly.express as px

# Load the dataset
work = pd.read_csv("Impact_of_Remote_Work_on_Mental_Health.csv")

# Enable Panel in Jupyter Notebook
pn.extension('plotly')

# Create interactive widgets
work_location = pn.widgets.Select(name="Work Location", options=list(work

# Function to update the plot based on selection
@pn.depends(work_location)
def update_plot(location):
    filtered_data = work[work['Work_Location'] == location]
    fig = px.bar(filtered_data, x="Job_Role", y="Stress_Level", color="Ge
    return fig

# Create dashboard layout
dashboard = pn.Column(
    pn.Row(pn.pane.Markdown("## Remote Work Dashboard")),
    pn.Row(work_location),
    pn.Row(pn.bind(update_plot, work_location))
)

# Display the dashboard in Jupyter Notebook
dashboard.show()
```

Launching server at http://localhost:49849

Out[28]: <panel.io.server.Server at 0x309d8bc50>

In []:

Complex Dashboard with Multiple Filters and Graph Types (Using Panel)

```
In [29]: pip install dash jupyter-dash
```

```
Requirement already satisfied: dash in /opt/homebrew/anaconda3/lib/python
3.11/site-packages (2.18.1)
Requirement already satisfied: jupyter-dash in /opt/homebrew/anaconda3/li
b/python3.11/site-packages (0.4.2)
Requirement already satisfied: Flask<3.1,>=1.0.4 in /opt/homebrew/anacond
a3/lib/python3.11/site-packages (from dash) (2.2.2)
Requirement already satisfied: Werkzeug<3.1 in /opt/homebrew/anaconda3/li
b/python3.11/site-packages (from dash) (2.2.3)
```


Requirement already satisfied: plotly>=5.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (5.9.0)

Requirement already satisfied: dash-html-components==2.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.0.0)

Requirement already satisfied: dash-core-components==2.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.0.0)

Requirement already satisfied: dash-table==5.0.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (5.0.0)

Requirement already satisfied: importlib-metadata in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (6.0.0)

Requirement already satisfied: typing-extensions>=4.1.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (4.12.2)

Requirement already satisfied: requests in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (2.31.0)

Requirement already satisfied: retrying in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (1.3.4)

Requirement already satisfied: nest-asyncio in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (1.6.0)

Requirement already satisfied: setuptools in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from dash) (68.0.0)

Requirement already satisfied: ipython in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-dash) (8.15.0)

Requirement already satisfied: ipykernel in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-dash) (6.25.0)

Requirement already satisfied: ansi2html in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-dash) (1.9.2)

Requirement already satisfied: Jinja2>=3.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Flask<3.1,>=1.0.4->dash) (3.1.2)

Requirement already satisfied: itsdangerous>=2.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Flask<3.1,>=1.0.4->dash) (2.0.1)

Requirement already satisfied: click>=8.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Flask<3.1,>=1.0.4->dash) (8.0.4)

Requirement already satisfied: tenacity>=6.2.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from plotly>=5.0.0->dash) (8.5.0)

Requirement already satisfied: MarkupSafe>=2.1.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from Werkzeug<3.1->dash) (2.1.1)

Requirement already satisfied: zipp>=0.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from importlib-metadata->dash) (3.11.0)

Requirement already satisfied: appnope in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (0.1.2)

Requirement already satisfied: comm>=0.1.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (0.1.2)

Requirement already satisfied: debugpy>=1.6.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (1.6.7)

Requirement already satisfied: jupyter-client>=6.1.12 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (7.4.9)

Requirement already satisfied: jupyter-core!=5.0.*,>=4.1.2 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (5.3.0)

Requirement already satisfied: matplotlib-inline>=0.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (0.1.6)

Requirement already satisfied: packaging in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (24.1)

Requirement already satisfied: psutil in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (5.9.0)

Requirement already satisfied: pyzmq>=20 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (23.2.0)

Requirement already satisfied: tornado>=6.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (6.3.2)

Requirement already satisfied: traitlets>=5.4.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipykernel->jupyter-dash) (5.7.1)

Requirement already satisfied: backcall in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.2.0)

Requirement already satisfied: decorator in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (5.1.1)

Requirement already satisfied: jedi>=0.16 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.18.1)

Requirement already satisfied: pickleshare in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.7.5)

Requirement already satisfied: prompt-toolkit!=3.0.37,<3.1.0,>=3.0.30 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (3.0.36)

Requirement already satisfied: pygments>=2.4.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (2.15.1)

Requirement already satisfied: stack-data in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (0.2.0)

Requirement already satisfied: pexpect>4.3 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from ipython->jupyter-dash) (4.8.0)

Requirement already satisfied: charset-normalizer<4,>=2 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (2.2.2)

Requirement already satisfied: certifi>=2017.4.17 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from requests->dash) (2023.7.22)

Requirement already satisfied: six>=1.7.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from retrying->dash) (1.16.0)

Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jedi>=0.16->ipython->jupyter-dash) (0.8.3)

Requirement already satisfied: entrypoints in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-client>=6.1.12->ipykernel->jupyter-dash) (0.4)

Requirement already satisfied: python-dateutil>=2.8.2 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-client>=6.1.12->ipykernel->jupyter-dash) (2.8.2)

Requirement already satisfied: platformdirs>=2.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from jupyter-core!=5.0.*,>=4.12->ipykernel->jupyter-dash) (3.10.0)

Requirement already satisfied: ptyprocess>=0.5 in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from pexpect>4.3->ipython->jupyter-dash) (0.7.0)

Requirement already satisfied: wcwidth in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from prompt-toolkit!=3.0.37,<3.1.0,>=3.0.30->ipython->jupyter-dash) (0.2.5)

Requirement already satisfied: executing in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from stack-data->ipython->jupyter-dash) (0.8.3)

Requirement already satisfied: asttokens in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from stack-data->ipython->jupyter-dash) (2.0.5)

Requirement already satisfied: pure-eval in /opt/homebrew/anaconda3/lib/python3.11/site-packages (from stack-data->ipython->jupyter-dash) (0.2.2)

Note: you may need to restart the kernel to use updated packages.

In []:

```
In [30]: # Import necessary libraries
import panel as pn
import pandas as pd
import plotly.express as px

# Load the dataset
work = pd.read_csv("Impact_of_Remote_Work_on_Mental_Health.csv")

# Enable Panel in Jupyter Notebook
pn.extension('plotly')

# Create widgets for filtering
gender_select = pn.widgets.Select(name="Gender", options=list(work['Gender']))
region_select = pn.widgets.Select(name="Region", options=list(work['Region']))

# Function to update graphs based on filters
@pn.depends(gender_select, region_select)
def update_charts(gender, region):
    filtered_data = work[(work['Gender'] == gender) & (work['Region'] == region)]

    # Create multiple charts
    stress_fig = px.histogram(filtered_data, x="Stress_Level", title=f'Stress Level by {region}')
    productivity_fig = px.bar(filtered_data, x="Job_Role", y="Productivity_Level", title=f'Productivity by {region}')
    satisfaction_fig = px.pie(filtered_data, names="Satisfaction_with_Remote_Work", title=f'Satisfaction with Remote Work by {region}')

    # Return the plots as a Panel Row layout
    return pn.Row(stress_fig, productivity_fig, satisfaction_fig)

# Create dashboard layout
dashboard = pn.Column(
    pn.Row(pn.pane.Markdown("## Complex Remote Work Dashboard")),
    pn.Row(gender_select, region_select),
    pn.Row(update_charts)
)

# Display the dashboard in Jupyter Notebook
dashboard.show()
```

Launching server at http://localhost:49851

Out[30]: <panel.io.server.Server at 0x30c346910>

In []:

In []:

In []: