Remote Work Health Impact Survey Analysis June 2025

M2M - Capstone Project 1 - Data Analysis and Visualization

Senthilvadivu Makileeshwaran
08-July-2025

Project OverView

Exploring remote work's mental & physical health impact post-pandemic

- To analyze how mental health status is impacted by different work arrangements (remote, onsite, and hybrid)
- To evaluate how physical health is affected under various work models

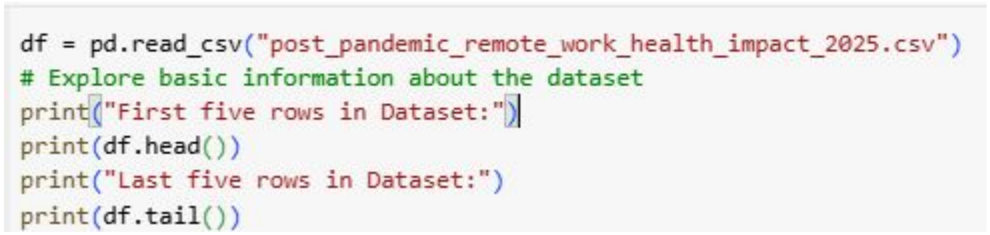
Tools and Technologies:

- Google Colab notebook
- Python - Pandas for read data from CSV and data manipulation
- Visualization Library: Bokeh
- Dashboard: PowerBI

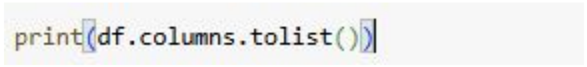
Dataset Overview:

- post_pandemic_remote_work_health_impact_2025.csv : https://www.kaggle.com/datasets/pratyushpuri/remote-work-health-impact-survey-2025
- The "Post-Pandemic Remote Work Health Impact 2025" - remote, hybrid, and onsite work arrangements are influencing the mental and physical health of employees in the post-pandemic era.
- Collected in June 2025, this dataset aggregates responses from different continents, industries, age groups, and job roles.
- Columns: Survey_Date, Age, Gender, Region, Industry, Job_Role, Work_Arrangement, Hours_Per_Week, Mental_Health_Status, Work_Life_Balance_Score, Physical_Health_Issues, Salary_Range, etc
- 3157 Rows

Basic Dataset Overview:

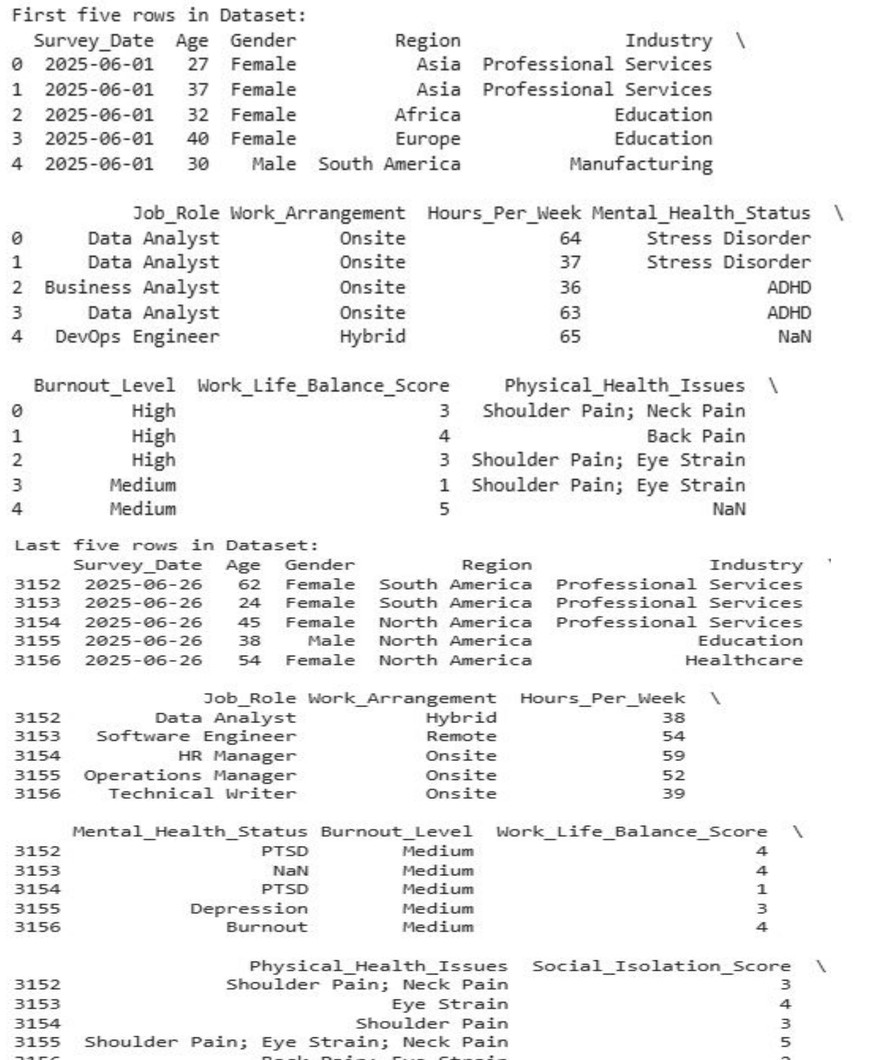


```
df = pd.read_csv("post_pandemic_remote_work_health_impact_2025.csv")
# Explore basic information about the dataset
print("First five rows in Dataset:")
print(df.head())
print("Last five rows in Dataset:")
print(df.tail())
```



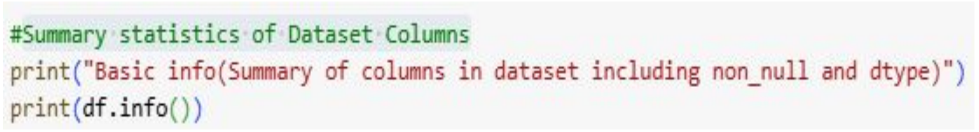
```
print(df.columns.tolist())
```

```
['Survey_Date', 'Age', 'Gender', 'Region',
'Industry', 'Job_Role', 'Work_Arrangement',
'Hours_Per_Week', 'Mental_Health_Status',
'Burnout_Level', 'Work_Life_Balance_Score',
'Physical_Health_Issues', 'Social_Isolation_Score',
'Salary_Range']
```

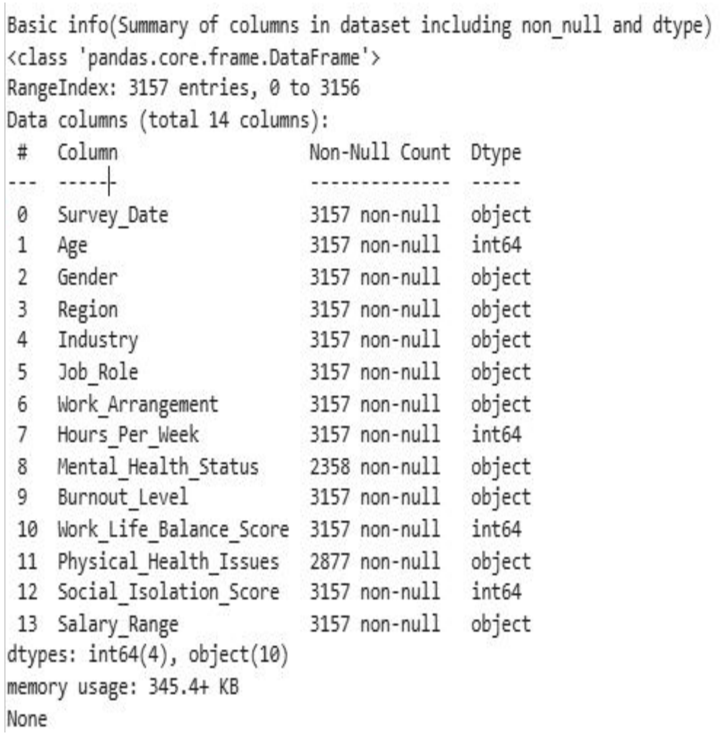


```
First five rows in Dataset:
  Survey_Date  Age  Gender         Region               Industry  \
0  2025-06-01   27  Female           Asia  Professional Services
1  2025-06-01   37  Female           Asia  Professional Services
2  2025-06-01   32  Female         Africa              Education
3  2025-06-01   40  Female         Europe              Education
4  2025-06-01   30    Male  South America          Manufacturing

           Job_Role Work_Arrangement  Hours_Per_Week Mental_Health_Status  \
0      Data Analyst           Onsite              64      Stress Disorder
1      Data Analyst           Onsite              37      Stress Disorder
2  Business Analyst           Onsite              36                 ADHD
3      Data Analyst           Onsite              63                 ADHD
4   DevOps Engineer           Hybrid              65                  NaN

  Burnout_Level  Work_Life_Balance_Score     Physical_Health_Issues  \
0          High                        3    Shoulder Pain; Neck Pain
1          High                        4                   Back Pain
2          High                        3   Shoulder Pain; Eye Strain
3        Medium                        1   Shoulder Pain; Eye Strain
4        Medium                        5                         NaN
Last five rows in Dataset:
     Survey_Date  Age  Gender         Region               Industry  \
3152  2025-06-26   62  Female  South America  Professional Services
3153  2025-06-26   24  Female  South America  Professional Services
3154  2025-06-26   45  Female  North America  Professional Services
3155  2025-06-26   38    Male  North America              Education
3156  2025-06-26   54  Female  North America             Healthcare

                Job_Role Work_Arrangement  Hours_Per_Week  \
3152        Data Analyst           Hybrid              38
3153   Software Engineer           Remote              54
3154          HR Manager           Onsite              59
3155  Operations Manager           Onsite              52
3156    Technical Writer           Onsite              39

     Mental_Health_Status Burnout_Level  Work_Life_Balance_Score  \
3152                 PTSD        Medium                        4
3153                  NaN        Medium                        4
3154                 PTSD        Medium                        1
3155           Depression        Medium                        3
3156              Burnout        Medium                        4

                    Physical_Health_Issues  Social_Isolation_Score  \
3152             Shoulder Pain; Neck Pain                        3
3153                           Eye Strain                        4
3154                        Shoulder Pain                        3
3155  Shoulder Pain; Eye Strain; Neck Pain                        5
```

Summary Statistics:



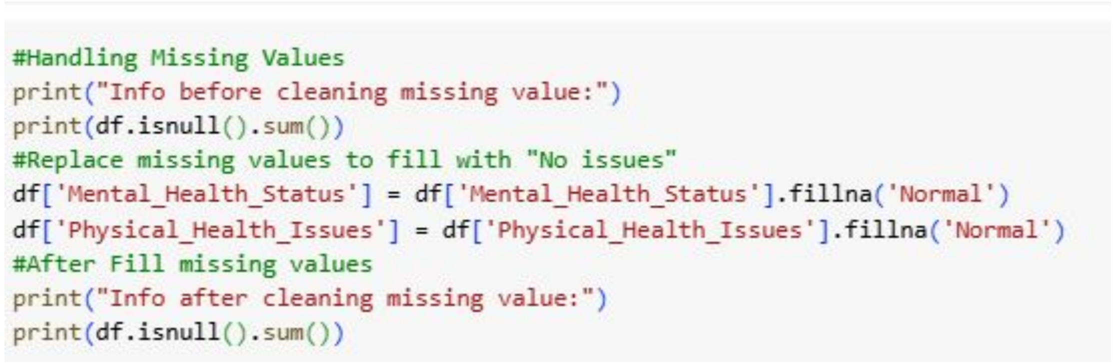
```
#Summary statistics of Dataset Columns
print("Basic info(Summary of columns in dataset including non_null and dtype)")
print(df.info())
```



```
Basic info(Summary of columns in dataset including non_null and dtype)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3157 entries, 0 to 3156
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Survey_Date             3157 non-null   object
 1   Age                     3157 non-null   int64
 2   Gender                  3157 non-null   object
 3   Region                  3157 non-null   object
 4   Industry                3157 non-null   object
 5   Job_Role                3157 non-null   object
 6   Work_Arrangement        3157 non-null   object
 7   Hours_Per_Week          3157 non-null   int64
 8   Mental_Health_Status    2358 non-null   object
 9   Burnout_Level           3157 non-null   object
 10  Work_Life_Balance_Score 3157 non-null   int64
 11  Physical_Health_Issues  2877 non-null   object
 12  Social_Isolation_Score  3157 non-null   int64
 13  Salary_Range            3157 non-null   object
dtypes: int64(4), object(10)
memory usage: 345.4+ KB
None
```

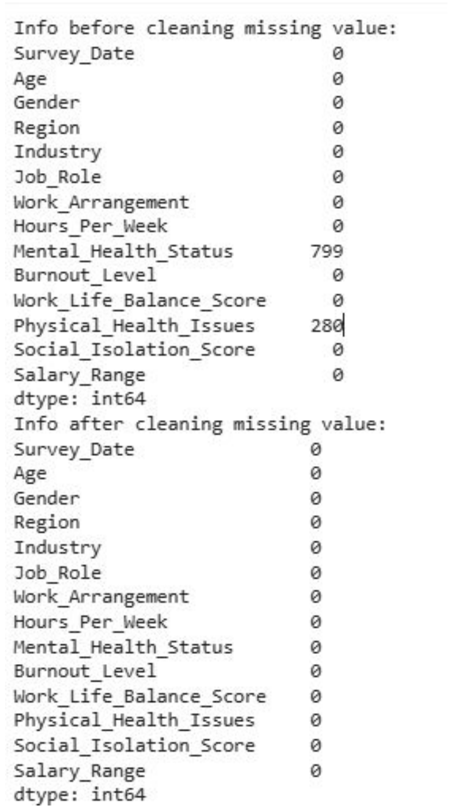
"Mental_Health_Issues"and "Physical_Health_Issues" contains null values.

Data Preparation and Cleaning:



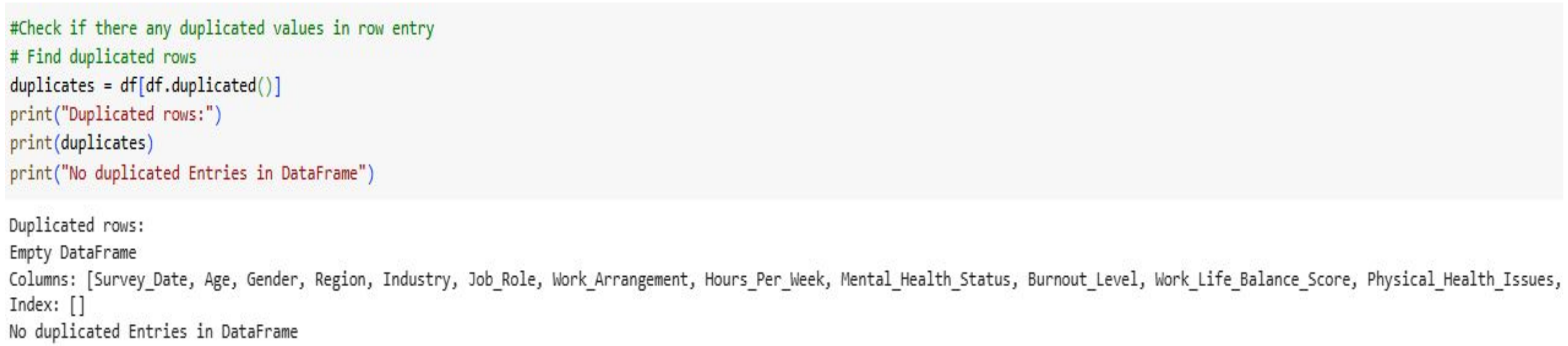
```python
#Handling Missing Values
print("Info before cleaning missing value:")
print(df.isnull().sum())
#Replace missing values to fill with "No issues"
df['Mental_Health_Status'] = df['Mental_Health_Status'].fillna('Normal')
df['Physical_Health_Issues'] = df['Physical_Health_Issues'].fillna('Normal')
#After Fill missing values
print("Info after cleaning missing value:")
print(df.isnull().sum())
```

Here I filled null values with Normal using 'fillna'



```
Info before cleaning missing value:
Survey_Date                 0
Age                         0
Gender                      0
Region                      0
Industry                    0
Job_Role                    0
Work_Arrangement            0
Hours_Per_Week              0
Mental_Health_Status      799
Burnout_Level               0
Work_Life_Balance_Score     0
Physical_Health_Issues    280
Social_Isolation_Score      0
Salary_Range                0
dtype: int64
Info after cleaning missing value:
Survey_Date                0
Age                        0
Gender                     0
Region                     0
Industry                   0
Job_Role                   0
Work_Arrangement           0
Hours_Per_Week             0
Mental_Health_Status       0
Burnout_Level              0
Work_Life_Balance_Score    0
Physical_Health_Issues     0
Social_Isolation_Score     0
Salary_Range               0
dtype: int64
```

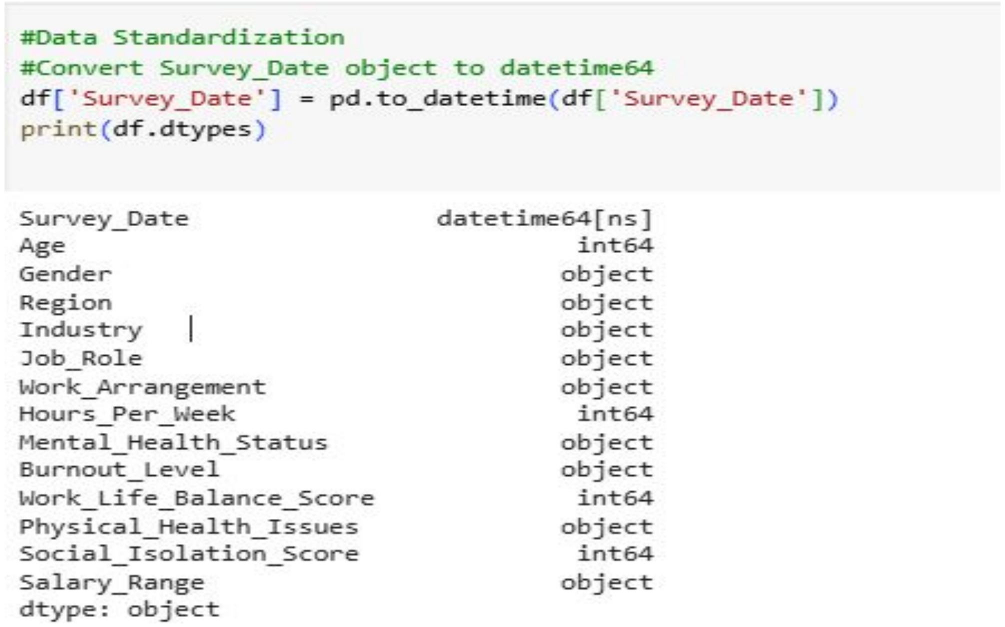
Check for Duplicate Entries



```
#Check if there any duplicated values in row entry
# Find duplicated rows
duplicates = df[df.duplicated()]
print("Duplicated rows:")
print(duplicates)
print("No duplicated Entries in DataFrame")
```

```
Duplicated rows:
Empty DataFrame
Columns: [Survey_Date, Age, Gender, Region, Industry, Job_Role, Work_Arrangement, Hours_Per_Week, Mental_Health_Status, Burnout_Level, Work_Life_Balance_Score, Physical_Health_Issues,
Index: []
No duplicated Entries in DataFrame
```

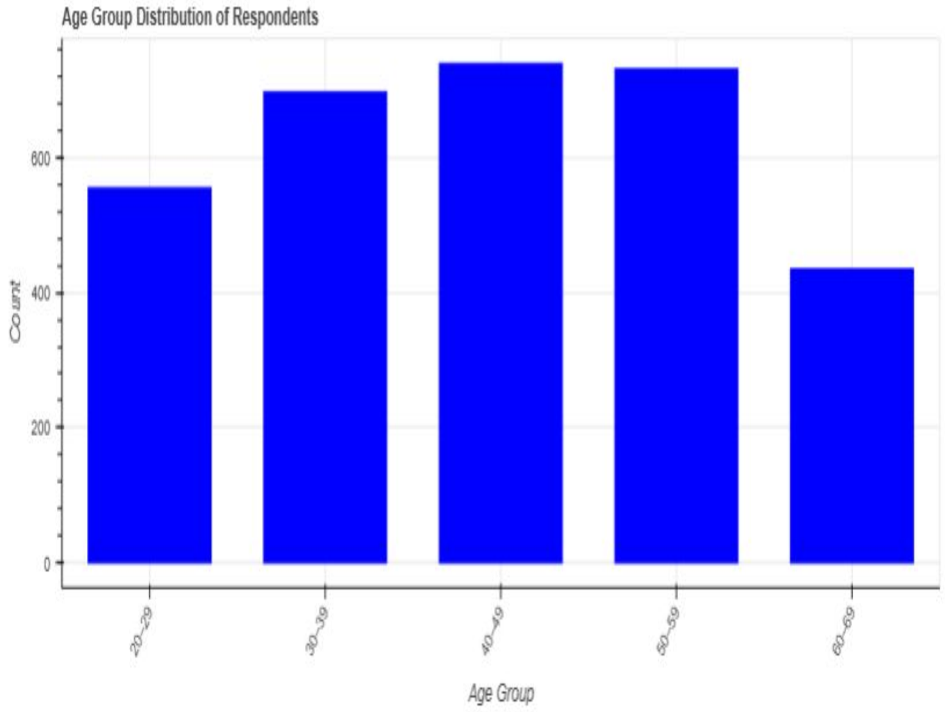
Data Standardization:

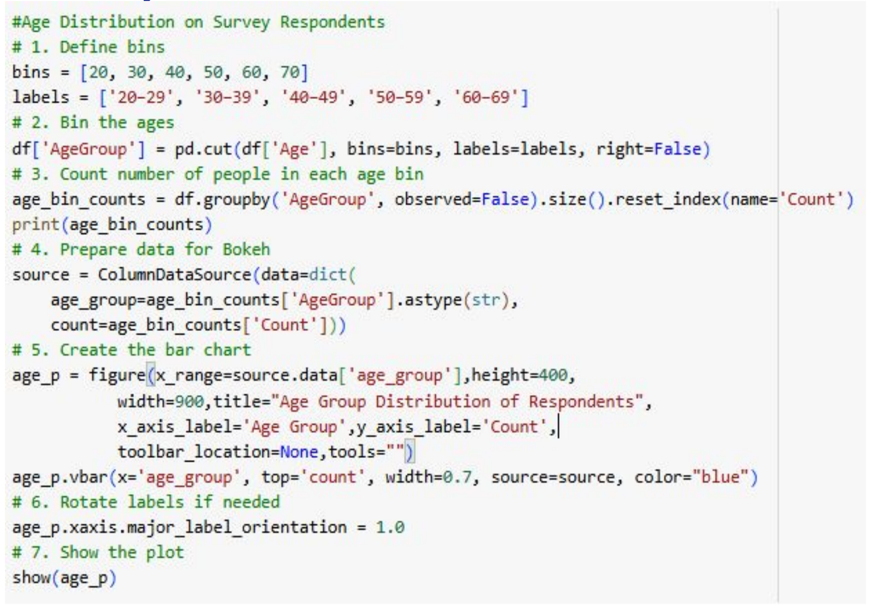


```
#Data Standardization
#Convert Survey_Date object to datetime64
df['Survey_Date'] = pd.to_datetime(df['Survey_Date'])
print(df.dtypes)
```

```
Survey_Date                  datetime64[ns]
Age                                   int64
Gender                               object
Region                               object
Industry                             object
Job_Role                             object
Work_Arrangement                     object
Hours_Per_Week                        int64
Mental_Health_Status                 object
Burnout_Level                        object
Work_Life_Balance_Score               int64
Physical_Health_Issues               object
Social_Isolation_Score                int64
Salary_Range                         object
dtype: object
```

Exploratory Data Analysis(EDA):

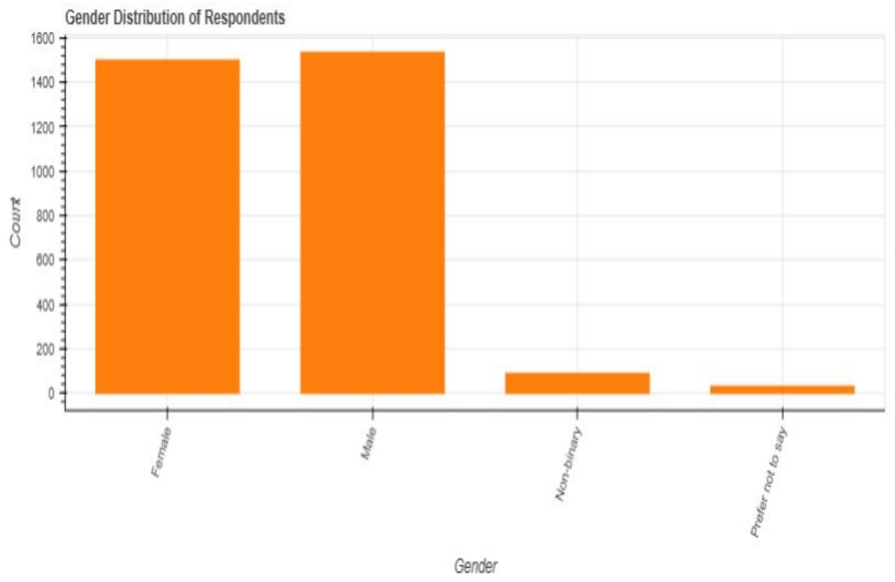


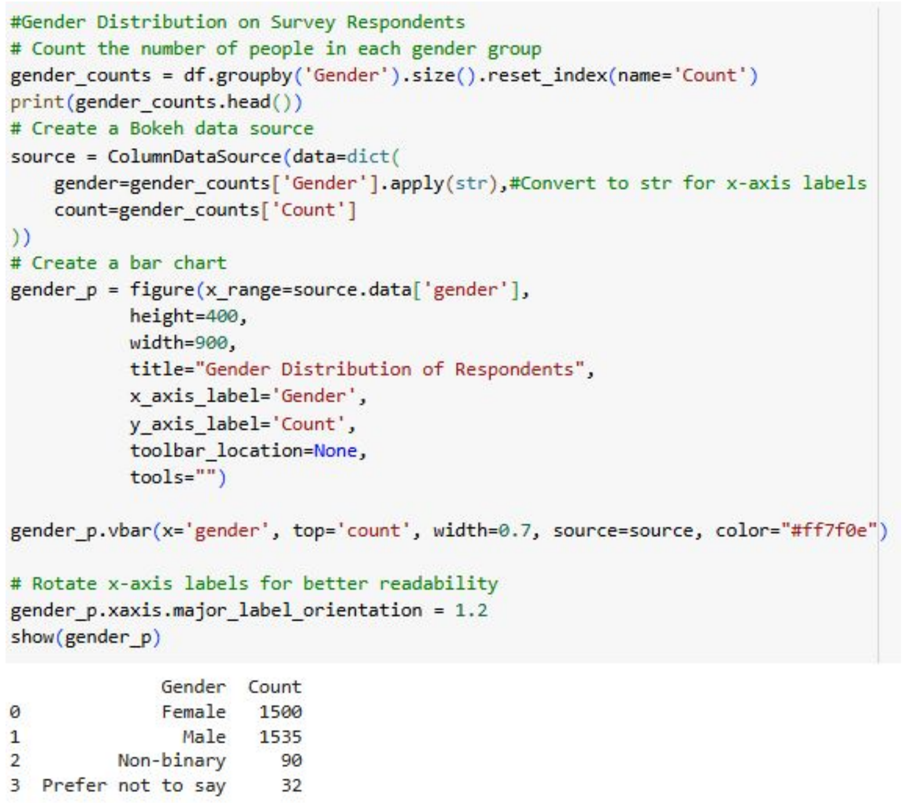


```
#Age Distribution on Survey Respondents
# 1. Define bins
bins = [20, 30, 40, 50, 60, 70]
labels = ['20-29', '30-39', '40-49', '50-59', '60-69']
# 2. Bin the ages
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)
# 3. Count number of people in each age bin
age_bin_counts = df.groupby('AgeGroup', observed=False).size().reset_index(name='Count')
print(age_bin_counts)
# 4. Prepare data for Bokeh
source = ColumnDataSource(data=dict(
    age_group=age_bin_counts['AgeGroup'].astype(str),
    count=age_bin_counts['Count']))
# 5. Create the bar chart
age_p = figure(x_range=source.data['age_group'],height=400,
           width=900,title="Age Group Distribution of Respondents",
           x_axis_label='Age Group',y_axis_label='Count',
           toolbar_location=None,tools="")
age_p.vbar(x='age_group', top='count', width=0.7, source=source, color="blue")
# 6. Rotate labels if needed
age_p.xaxis.major_label_orientation = 1.0
# 7. Show the plot
show(age_p)
```

```
  AgeGroup  Count
0    20-29    555
1    30-39    697
2    40-49    739
3    50-59    731
4    60-69    435
```

Exploratory Data Analysis(EDA):

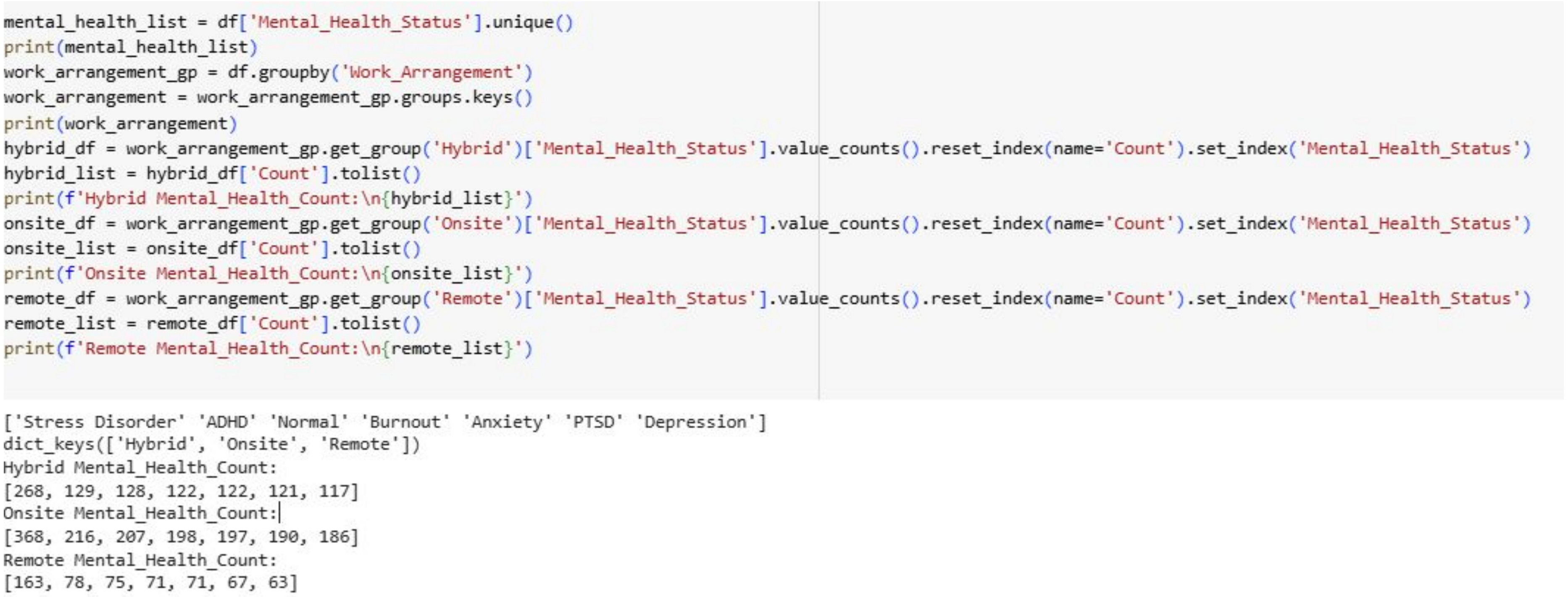




```
#Gender Distribution on Survey Respondents
# Count the number of people in each gender group
gender_counts = df.groupby('Gender').size().reset_index(name='Count')
print(gender_counts.head())
# Create a Bokeh data source
source = ColumnDataSource(data=dict(
    gender=gender_counts['Gender'].apply(str),#Convert to str for x-axis labels
    count=gender_counts['Count']
))
# Create a bar chart
gender_p = figure(x_range=source.data['gender'],
           height=400,
           width=900,
           title="Gender Distribution of Respondents",
           x_axis_label='Gender',
           y_axis_label='Count',
           toolbar_location=None,
           tools="")

gender_p.vbar(x='gender', top='count', width=0.7, source=source, color="#ff7f0e")

# Rotate x-axis labels for better readability
gender_p.xaxis.major_label_orientation = 1.2
show(gender_p)
```

```
              Gender  Count
0             Female   1500
1               Male   1535
2         Non-binary     90
3  Prefer not to say     32
```

Data Analysis: Mental Health Status by Work Arrangement

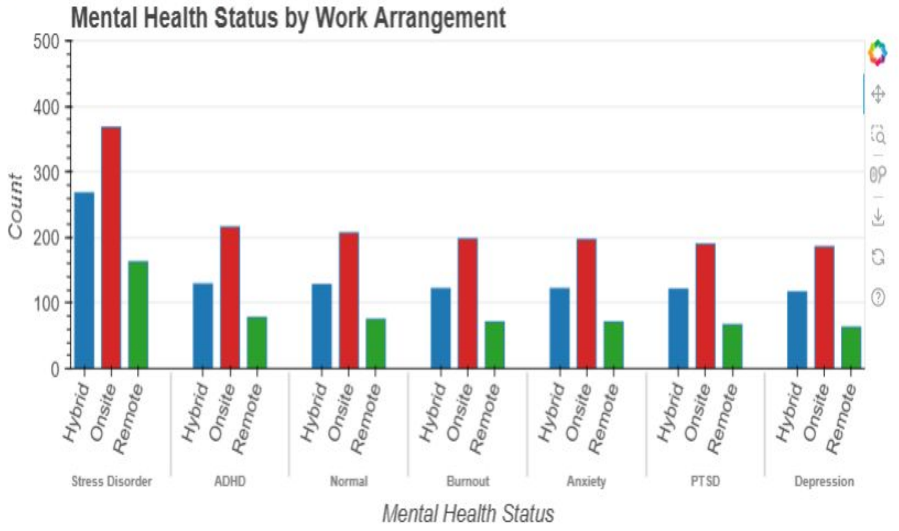


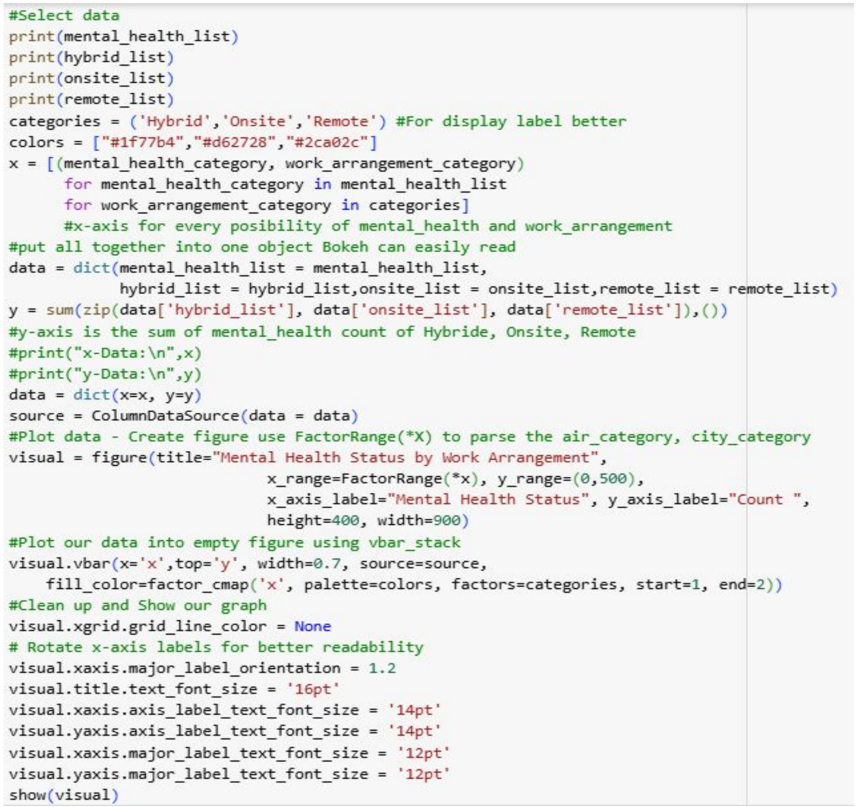
```python
mental_health_list = df['Mental_Health_Status'].unique()
print(mental_health_list)
work_arrangement_gp = df.groupby('Work_Arrangement')
work_arrangement = work_arrangement_gp.groups.keys()
print(work_arrangement)
hybrid_df = work_arrangement_gp.get_group('Hybrid')['Mental_Health_Status'].value_counts().reset_index(name='Count').set_index('Mental_Health_Status')
hybrid_list = hybrid_df['Count'].tolist()
print(f'Hybrid Mental_Health_Count:\n{hybrid_list}')
onsite_df = work_arrangement_gp.get_group('Onsite')['Mental_Health_Status'].value_counts().reset_index(name='Count').set_index('Mental_Health_Status')
onsite_list = onsite_df['Count'].tolist()
print(f'Onsite Mental_Health_Count:\n{onsite_list}')
remote_df = work_arrangement_gp.get_group('Remote')['Mental_Health_Status'].value_counts().reset_index(name='Count').set_index('Mental_Health_Status')
remote_list = remote_df['Count'].tolist()
print(f'Remote Mental_Health_Count:\n{remote_list}')
```

```
['Stress Disorder' 'ADHD' 'Normal' 'Burnout' 'Anxiety' 'PTSD' 'Depression']
dict_keys(['Hybrid', 'Onsite', 'Remote'])
Hybrid Mental_Health_Count:
[268, 129, 128, 122, 122, 121, 117]
Onsite Mental_Health_Count:
[368, 216, 207, 198, 197, 190, 186]
Remote Mental_Health_Count:
[163, 78, 75, 71, 71, 67, 63]
```

Data Analysis: Mental Health Status by Work Arrangement

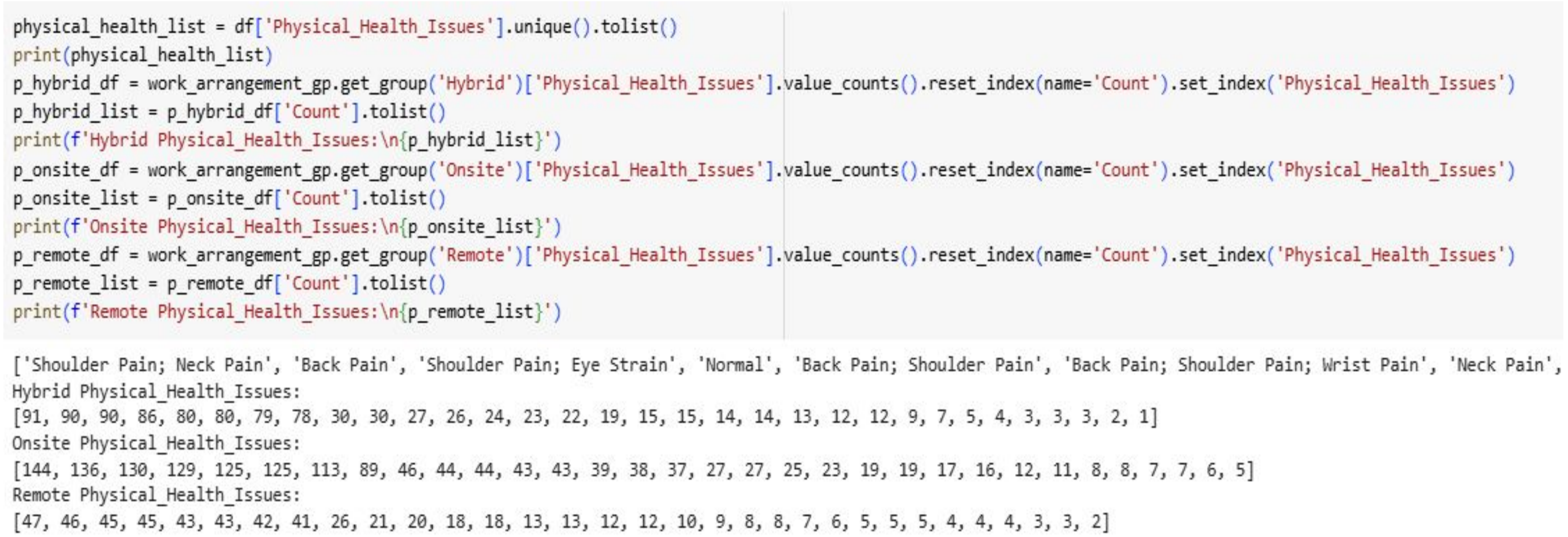
```
['Stress Disorder' 'ADHD' 'Normal' 'Burnout' 'Anxiety' 'PTSD' 'Depression']
[268, 129, 128, 122, 122, 121, 117]
[368, 216, 207, 198, 197, 190, 186]
[163, 78, 75, 71, 71, 67, 63]
```





```
#Select data
print(mental_health_list)
print(hybrid_list)
print(onsite_list)
print(remote_list)
categories = ('Hybrid','Onsite','Remote') #For display label better
colors = ["#1f77b4","#d62728","#2ca02c"]
x = [(mental_health_category, work_arrangement_category)
      for mental_health_category in mental_health_list
      for work_arrangement_category in categories]
      #x-axis for every posibility of mental_health and work_arrangement
#put all together into one object Bokeh can easily read
data = dict(mental_health_list = mental_health_list,
            hybrid_list = hybrid_list,onsite_list = onsite_list,remote_list = remote_list)
y = sum(zip(data['hybrid_list'], data['onsite_list'], data['remote_list']),())
#y-axis is the sum of mental_health count of Hybride, Onsite, Remote
#print("x-Data:\n",x)
#print("y-Data:\n",y)
data = dict(x=x, y=y)
source = ColumnDataSource(data = data)
#Plot data - Create figure use FactorRange(*X) to parse the air_category, city_category
visual = figure(title="Mental Health Status by Work Arrangement",
                            x_range=FactorRange(*x), y_range=(0,500),
                            x_axis_label="Mental Health Status", y_axis_label="Count ",
                            height=400, width=900)
#Plot our data into empty figure using vbar_stack
visual.vbar(x='x',top='y', width=0.7, source=source,
    fill_color=factor_cmap('x', palette=colors, factors=categories, start=1, end=2))
#Clean up and Show our graph
visual.xgrid.grid_line_color = None
# Rotate x-axis labels for better readability
visual.xaxis.major_label_orientation = 1.2
visual.title.text_font_size = '16pt'
visual.xaxis.axis_label_text_font_size = '14pt'
visual.yaxis.axis_label_text_font_size = '14pt'
visual.xaxis.major_label_text_font_size = '12pt'
visual.yaxis.major_label_text_font_size = '12pt'
show(visual)
```

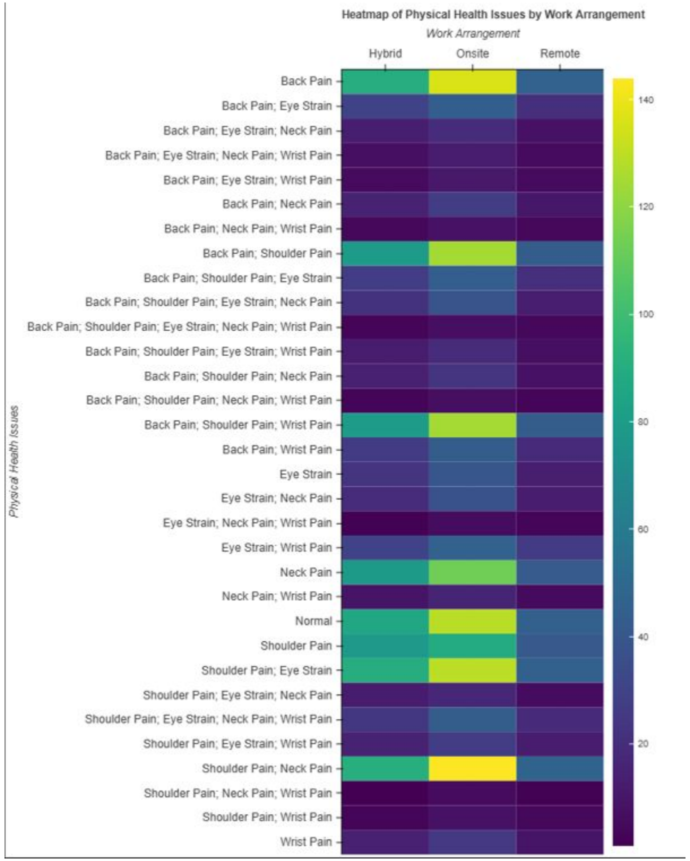
Data Analysis: Physical Health Issues by Work Arrangement

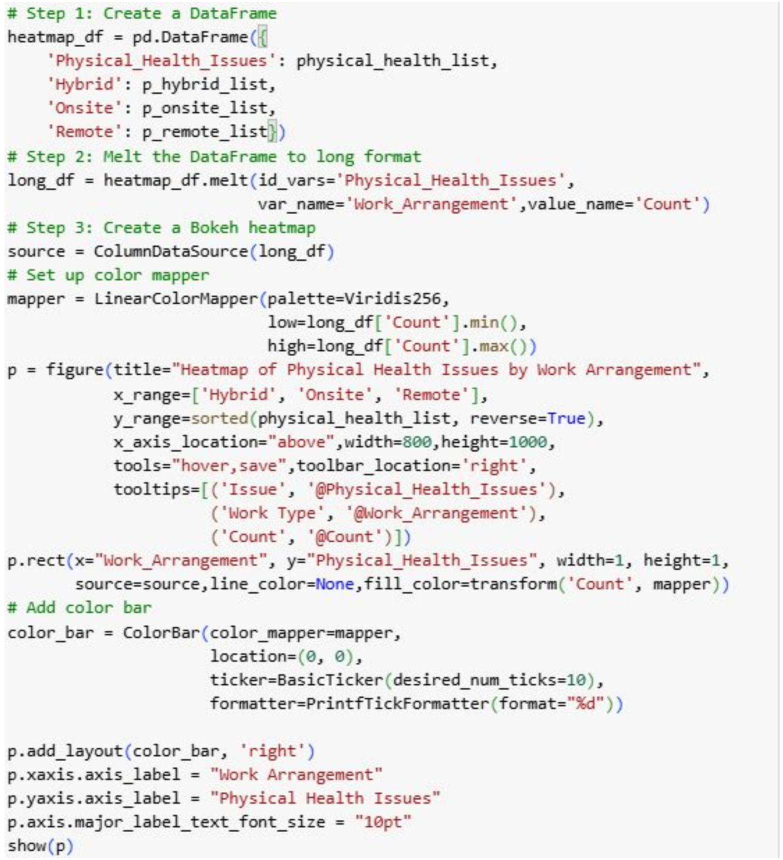


```
physical_health_list = df['Physical_Health_Issues'].unique().tolist()
print(physical_health_list)
p_hybrid_df = work_arrangement_gp.get_group('Hybrid')['Physical_Health_Issues'].value_counts().reset_index(name='Count').set_index('Physical_Health_Issues')
p_hybrid_list = p_hybrid_df['Count'].tolist()
print(f'Hybrid Physical_Health_Issues:\n{p_hybrid_list}')
p_onsite_df = work_arrangement_gp.get_group('Onsite')['Physical_Health_Issues'].value_counts().reset_index(name='Count').set_index('Physical_Health_Issues')
p_onsite_list = p_onsite_df['Count'].tolist()
print(f'Onsite Physical_Health_Issues:\n{p_onsite_list}')
p_remote_df = work_arrangement_gp.get_group('Remote')['Physical_Health_Issues'].value_counts().reset_index(name='Count').set_index('Physical_Health_Issues')
p_remote_list = p_remote_df['Count'].tolist()
print(f'Remote Physical_Health_Issues:\n{p_remote_list}')
```

```
['Shoulder Pain; Neck Pain', 'Back Pain', 'Shoulder Pain; Eye Strain', 'Normal', 'Back Pain; Shoulder Pain', 'Back Pain; Shoulder Pain; Wrist Pain', 'Neck Pain',
Hybrid Physical_Health_Issues:
[91, 90, 90, 86, 80, 80, 79, 78, 30, 30, 27, 26, 24, 23, 22, 19, 15, 15, 14, 14, 13, 12, 12, 9, 7, 5, 4, 3, 3, 3, 2, 1]
Onsite Physical_Health_Issues:
[144, 136, 130, 129, 125, 125, 113, 89, 46, 44, 44, 43, 43, 39, 38, 37, 27, 27, 25, 23, 19, 19, 17, 16, 12, 11, 8, 8, 7, 7, 6, 5]
Remote Physical_Health_Issues:
[47, 46, 45, 45, 43, 43, 42, 41, 26, 21, 20, 18, 18, 13, 13, 12, 12, 10, 9, 8, 8, 7, 6, 5, 5, 5, 4, 4, 4, 3, 3, 2]
```

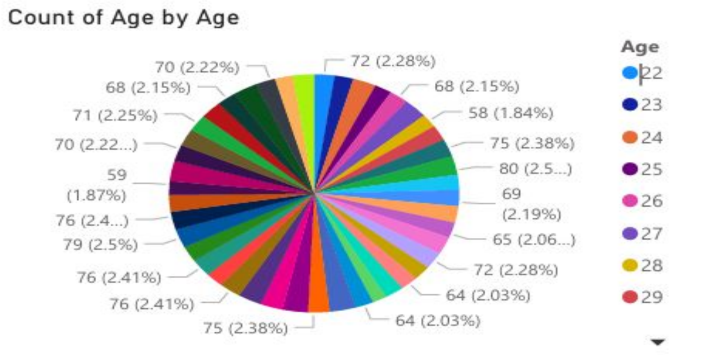
Data Analysis: Physical Health Issues by Work Arrangement

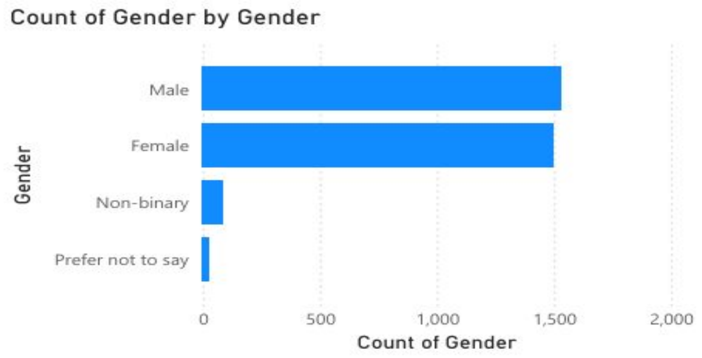


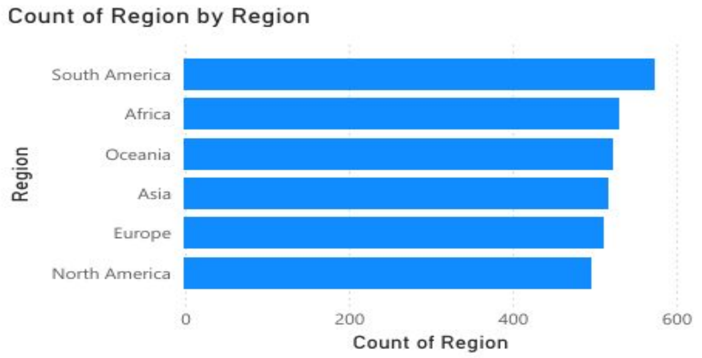


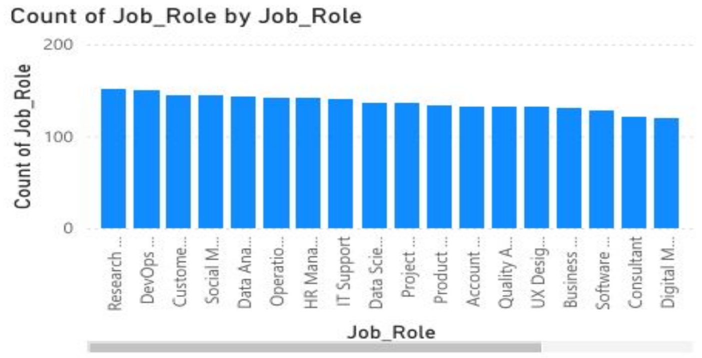
```python
# Step 1: Create a DataFrame
heatmap_df = pd.DataFrame({
    'Physical_Health_Issues': physical_health_list,
    'Hybrid': p_hybrid_list,
    'Onsite': p_onsite_list,
    'Remote': p_remote_list})
# Step 2: Melt the DataFrame to long format
long_df = heatmap_df.melt(id_vars='Physical_Health_Issues',
                          var_name='Work_Arrangement',value_name='Count')
# Step 3: Create a Bokeh heatmap
source = ColumnDataSource(long_df)
# Set up color mapper
mapper = LinearColorMapper(palette=Viridis256,
                           low=long_df['Count'].min(),
                           high=long_df['Count'].max())
p = figure(title="Heatmap of Physical Health Issues by Work Arrangement",
           x_range=['Hybrid', 'Onsite', 'Remote'],
           y_range=sorted(physical_health_list, reverse=True),
           x_axis_location="above",width=800,height=1000,
           tools="hover,save",toolbar_location='right',
           tooltips=[('Issue', '@Physical_Health_Issues'),
                     ('Work Type', '@Work_Arrangement'),
                     ('Count', '@Count')])
p.rect(x="Work_Arrangement", y="Physical_Health_Issues", width=1, height=1,
       source=source,line_color=None,fill_color=transform('Count', mapper))
# Add color bar
color_bar = ColorBar(color_mapper=mapper,
                     location=(0, 0),
                     ticker=BasicTicker(desired_num_ticks=10),
                     formatter=PrintfTickFormatter(format="%d"))

p.add_layout(color_bar, 'right')
p.xaxis.axis_label = "Work Arrangement"
p.yaxis.axis_label = "Physical Health Issues"
p.axis.major_label_text_font_size = "10pt"
show(p)
```

PowerBI - Dashboard



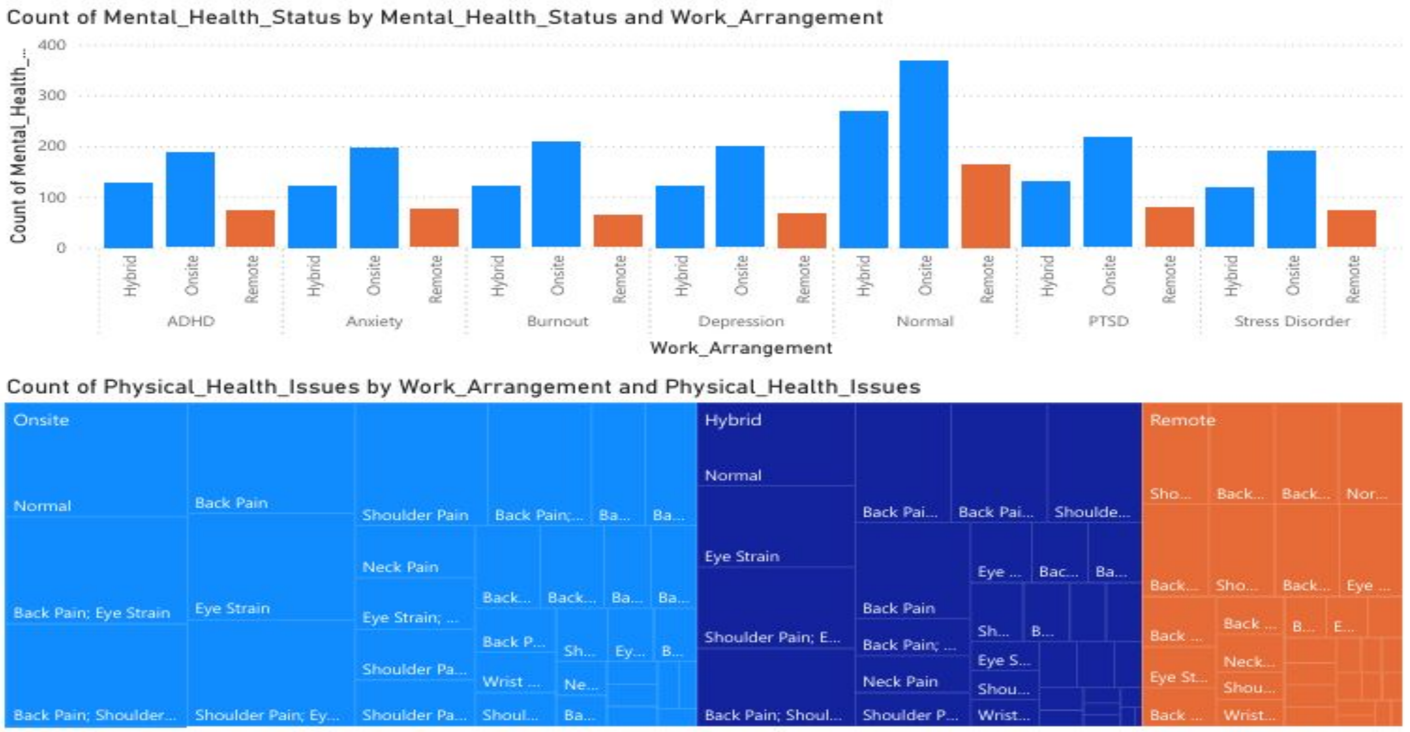






Distribution Of Respondents Count

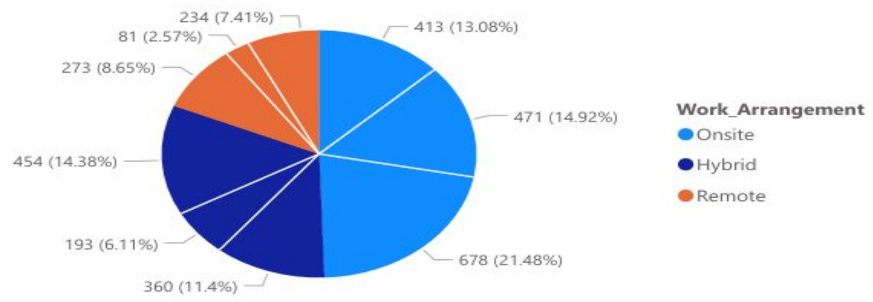
PowerBI - Dashboard



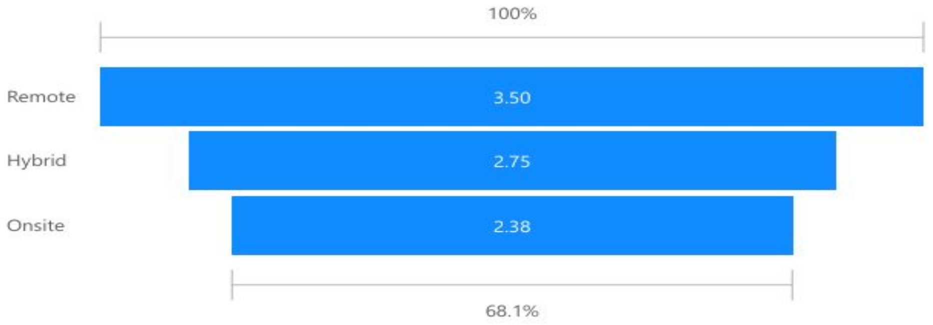
Visualization Of Work Arrangement Impact on Mental and Physical Issues

PowerBI - Dashboard

Count of Burnout_Level by Work_Arrangement and Burnout_Level



Average of Social_Isolation_Score by Work_Arrangement



Work Arrangement impact on Burnout Level And Social Isolation

Key Insights:

- People reported better mental health while working remotely compared to onsite or hybrid work.
- Remote work caused fewer physical health problems than onsite or hybrid work.
- Onsite work led to more mental and physical stress overall.
- Burnout levels were highest among people working onsite.
- Social isolation was reported higher in remote work settings, and lower in onsite work environments.

Thank you!