ANA500 Module 2

February 16, 2025

[71]:
```
# Step 1: Install & Import Required Libraries
# Install Plotly and Dash
!pip install plotly

# Import libraries for visualization
import numpy as np
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns
```

```
Requirement already satisfied: plotly in c:\users\rdarn\anaconda3\lib\site-
packages (5.24.1)
Requirement already satisfied: tenacity>=6.2.0 in
c:\users\rdarn\anaconda3\lib\site-packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in c:\users\rdarn\anaconda3\lib\site-
packages (from plotly) (24.1)
```

[19]:
```
# Step 2: Load the dataset and display the first five rows
file_path = "C:\\MISC\\alzheimers_prediction_dataset.csv"
df = pd.read_csv(file_path)

# Display the first few rows
df.head()
```

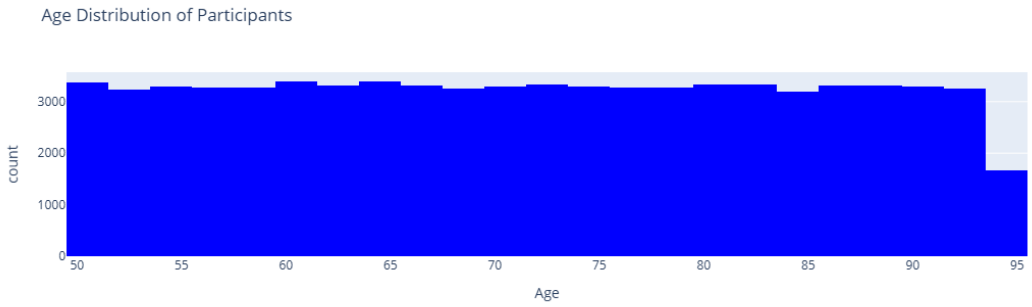
[19]:

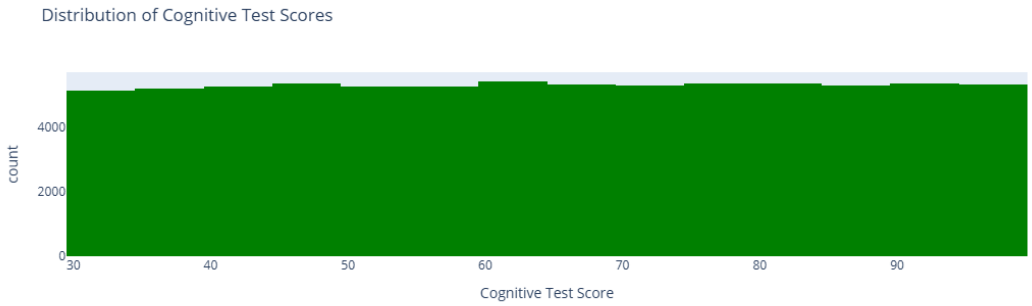
	Country	Age	Gender	Education Level	BMI	Physical Activity Level	\
0	Spain	90	Male	1	33.0	Medium	
1	Argentina	72	Male	7	29.9	Medium	
2	South Africa	86	Female	19	22.9	High	
3	China	53	Male	17	31.2	Low	
4	Sweden	58	Female	3	30.0	High	

	Smoking Status	Alcohol Consumption	Diabetes	Hypertension	...	\
0	Never	Occasionally	No	No	...	
1	Former	Never	No	No	...	
2	Current	Occasionally	No	Yes	...	
3	Never	Regularly	Yes	No	...	

```
4         Former                Never      Yes           No  ...

  Dietary Habits Air Pollution Exposure  Employment Status Marital Status  \
0        Healthy                   High            Retired         Single
1        Healthy                 Medium         Unemployed        Widowed
2        Average                 Medium           Employed         Single
3        Healthy                 Medium            Retired         Single
4      Unhealthy                   High           Employed        Married

  Genetic Risk Factor (APOE-ε4 allele) Social Engagement Level Income Level  \
0                                   No                     Low       Medium
1                                   No                    High          Low
2                                   No                     Low       Medium
3                                   No                    High       Medium
4                                   No                     Low       Medium

  Stress Levels Urban vs Rural Living Alzheimer's Diagnosis
0          High                  Urban                    No
1          High                  Urban                    No
2          High                  Rural                    No
3           Low                  Rural                    No
4          High                  Rural                    No

[5 rows x 25 columns]
```

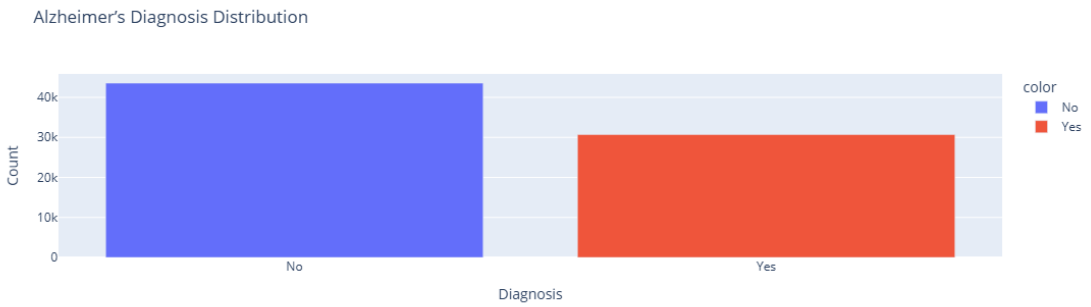
[33]:
```python
# Step 3: Visualize Numerical Data Distributions

# Histogram of Age Distribution
fig = px.histogram(df, x="Age",
                   nbins=30,
                   title="Age Distribution of Participants",
                   color_discrete_sequence=["blue"])
fig.show()

# Histogram of Cognitive Test Scores
fig = px.histogram(df, x="Cognitive Test Score",
                   nbins=20,
                   title="Distribution of Cognitive Test Scores",
                   color_discrete_sequence=["green"])
fig.show()
```

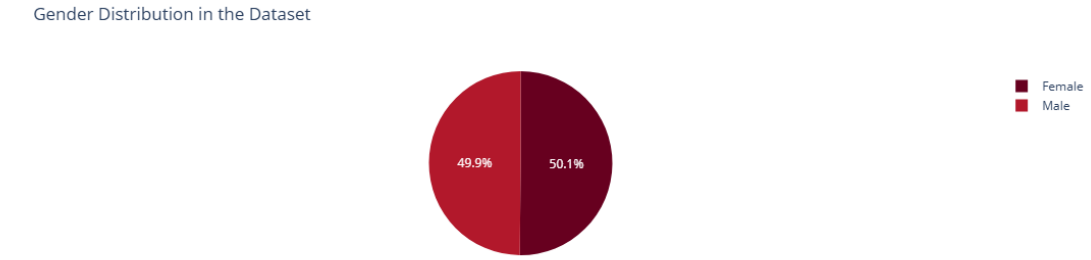


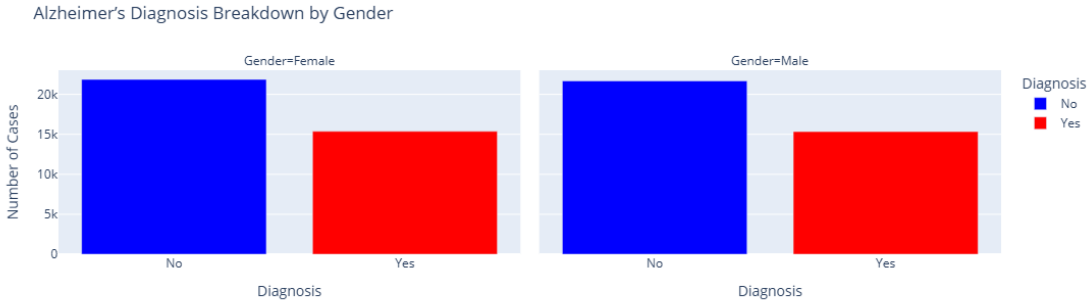


[60]:
```
# Step 4: Visualize Categorical Data

# Bar Chart of Alzheimer's Diagnosis
fig = px.bar(df["Alzheimer's Diagnosis"].value_counts(),
             x=df["Alzheimer's Diagnosis"].value_counts().index,
             y=df["Alzheimer's Diagnosis"].value_counts().values,
             title="Alzheimer's Diagnosis Distribution",
             color=df["Alzheimer's Diagnosis"].value_counts().index,
             labels={'x': 'Diagnosis', 'y': 'Count'})
fig.show()

# Gender Distribution
fig = px.pie(df, names="Gender",
             title="Gender Distribution in the Dataset",
             color_discrete_sequence=px.colors.sequential.RdBu)
fig.show()



# Count occurrences of "Yes" and "No" per gender
```

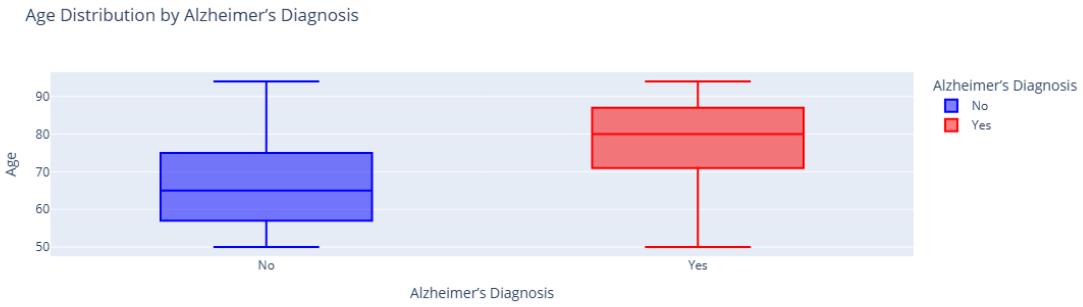
```
df_gender_diagnosis = df.groupby(["Gender", "Alzheimer's Diagnosis"]).size().
 ↪reset_index(name="Count")

# Create bar charts for Male and Female separately
fig = px.bar(df_gender_diagnosis,
             x="Alzheimer's Diagnosis",
             y="Count",
             facet_col="Gender",  # Create separate graphs for Male and Female
             title="Alzheimer's Diagnosis Breakdown by Gender",
             labels={"Count": "Number of Cases", "Alzheimer's Diagnosis":␣
 ↪"Diagnosis"},
             color="Alzheimer's Diagnosis",
             color_discrete_map={"Yes": "red", "No": "blue"})  # Assign colors

fig.show()
```

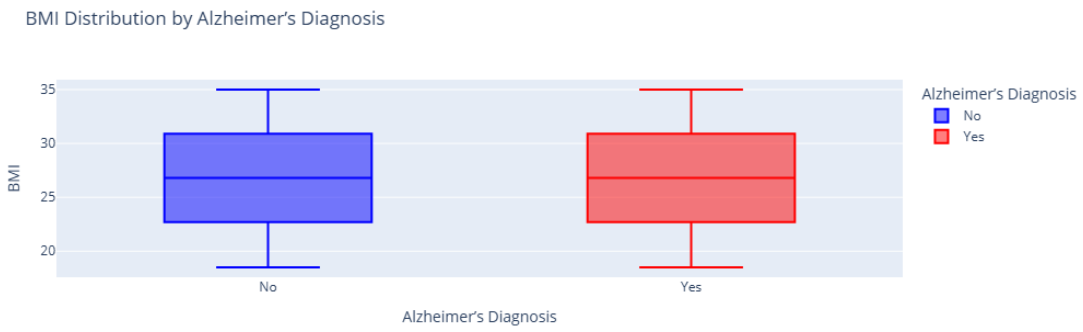


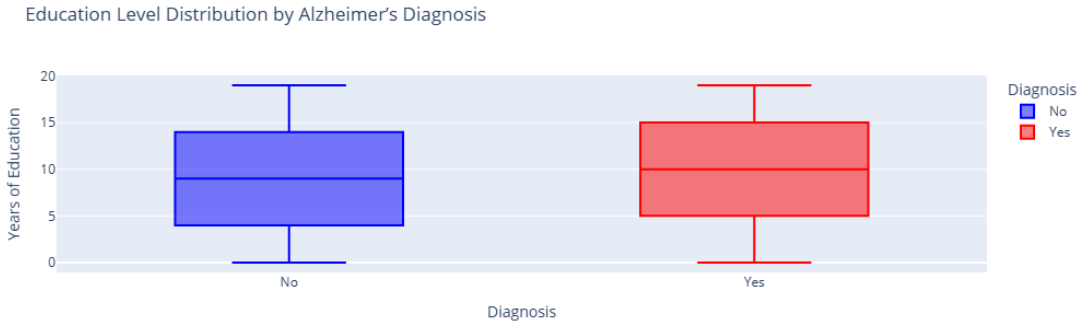




[88]:
```
# Step 5: Relationship Between Risk Factors & Alzheimer's

# Box Plot: Age vs. Alzheimer's Diagnosis
fig = px.box(df,
             x="Alzheimer's Diagnosis",
             y="Age",
             title="Age Distribution by Alzheimer's Diagnosis",
             color="Alzheimer's Diagnosis",
             color_discrete_map={"Yes": "red", "No": "blue"})  # Ensures correct␣
 ↪colors
fig.show()

# Box Plot: BMI vs. Alzheimer's Diagnosis
fig = px.box(df,
             x="Alzheimer's Diagnosis",
             y="BMI",
             title="BMI Distribution by Alzheimer's Diagnosis",
             color="Alzheimer's Diagnosis",
             color_discrete_map={"Yes": "red", "No": "blue"})  # Ensures correct␣
 ↪colors
fig.show()

# Box Plot: Education Level vs. Alzheimer's Diagnosis
fig = px.box(df,
             x="Alzheimer's Diagnosis",
             y="Education Level",
             title="Education Level Distribution by Alzheimer's Diagnosis",
             labels={"Education Level": "Years of Education", "Alzheimer's␣
 ↪Diagnosis": "Diagnosis"},
             color="Alzheimer's Diagnosis",
             color_discrete_map={"Yes": "red", "No": "blue"})  # Ensures correct␣
 ↪colors
fig.show()
```

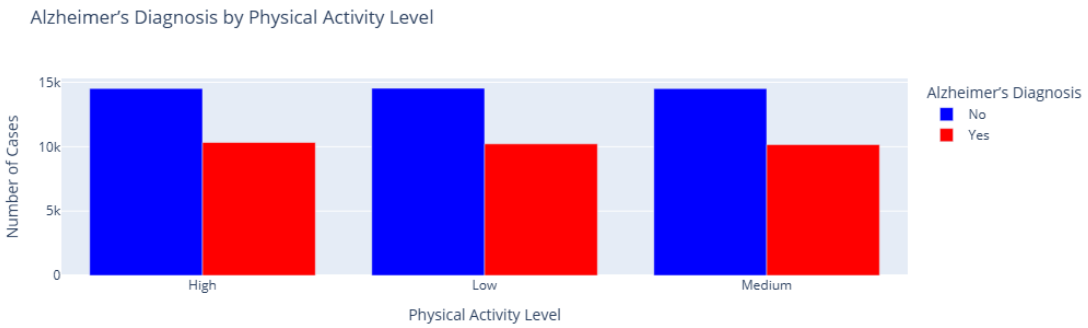


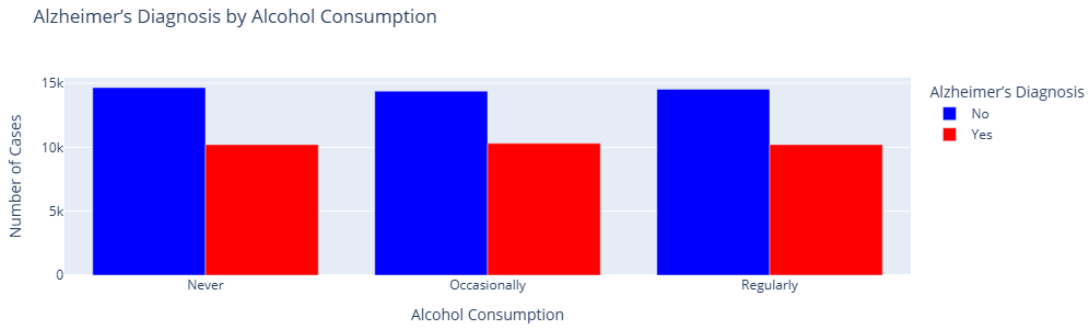




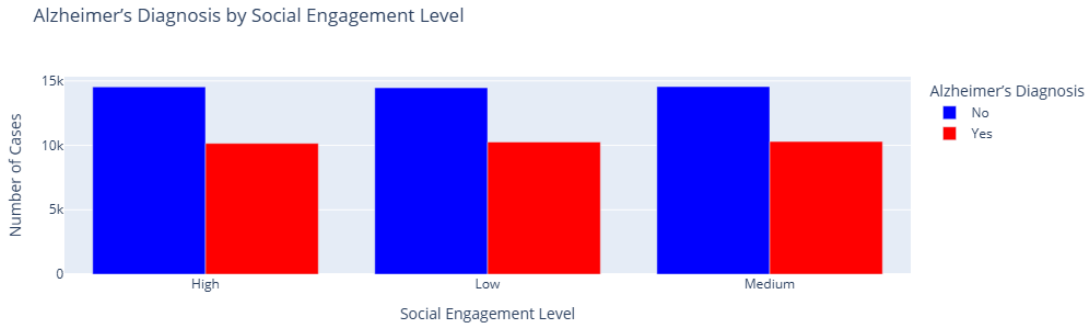
[107]:
```
# Bar Charts of Lifestyle Variables

# List of lifestyle variables
lifestyle_variables = ["Physical Activity Level", "Smoking Status", "Alcohol␣
↪Consumption",
```

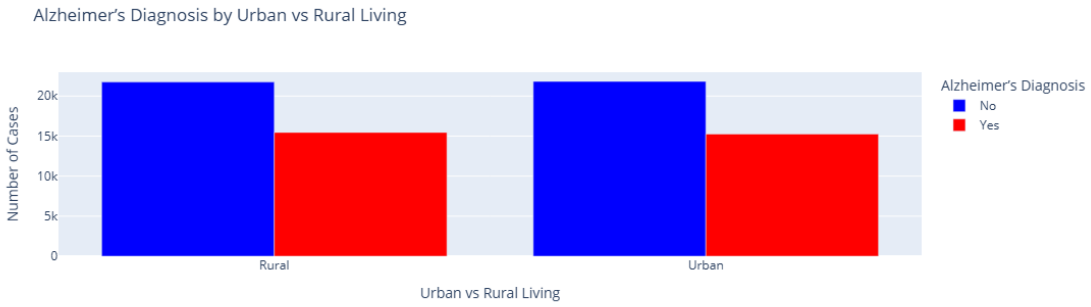
```
                    "Dietary Habits", "Social Engagement Level", "Urban vs␣
↪Rural Living"]

# Loop through each lifestyle factor and generate bar charts
for col in lifestyle_variables:
    df_counts = df.groupby([col, "Alzheimer's Diagnosis"]).size().
↪reset_index(name="Count")

    fig = px.bar(df_counts,
                 x=col,
                 y="Count",
                 color="Alzheimer's Diagnosis",
                 title=f"Alzheimer's Diagnosis by {col}",
                 labels={"Count": "Number of Cases", col: col},
                 barmode="group",  # Side-by-side bars for comparison
                 color_discrete_map={"Yes": "red", "No": "blue"})  # Assign␣
↪colors
    fig.show()
```




Alzheimer's Diagnosis by Alcohol Consumption
15k
10k
5k
0
Number of Cases
Never
Occasionally
Regularly
Alcohol Consumption
Alzheimer's Diagnosis
No
Yes

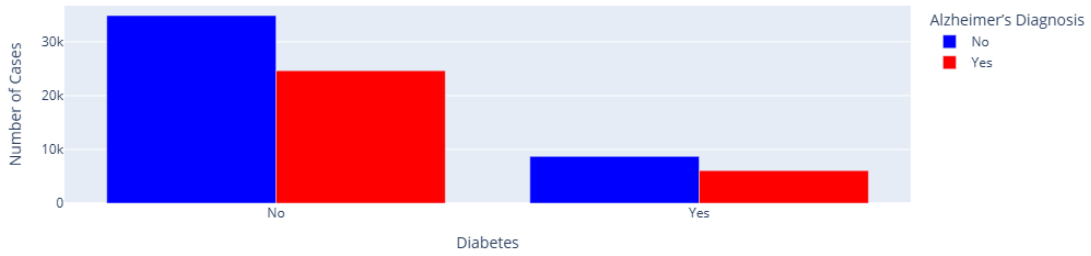

Alzheimer's Diagnosis by Dietary Habits
15k
10k
5k
0
Number of Cases
Average
Healthy
Unhealthy
Dietary Habits
Alzheimer's Diagnosis
No
Yes


Alzheimer's Diagnosis by Social Engagement Level
15k
10k
5k
0
Number of Cases
High
Low
Medium
Social Engagement Level
Alzheimer's Diagnosis
No
Yes

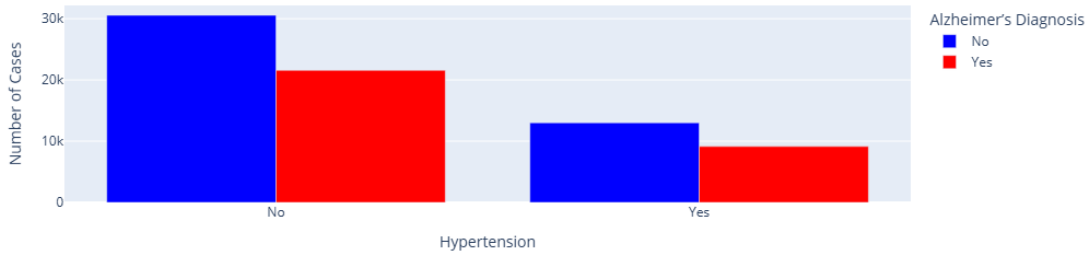


[109]:
```
# Bar Charts for Medical & Genetic Factors

# List of medical & genetic variables
medical_genetic_variables = ["Diabetes", "Hypertension", "Cholesterol Level",
                             "Genetic Risk Factor (APOE-ε4 allele)", "Family␣
↪History of Alzheimer's"]

# Loop through each medical & genetic factor and generate bar charts
for col in medical_genetic_variables:
    df_counts = df.groupby([col, "Alzheimer's Diagnosis"]).size().
↪reset_index(name="Count")

    fig = px.bar(df_counts,
                 x=col,
                 y="Count",
                 color="Alzheimer's Diagnosis",
                 title=f"Alzheimer's Diagnosis by {col}",
                 labels={"Count": "Number of Cases", col: col},
                 barmode="group",  # Side-by-side bars for comparison
                 color_discrete_map={"Yes": "red", "No": "blue"})  # Assign␣
↪colors

    fig.show()
```

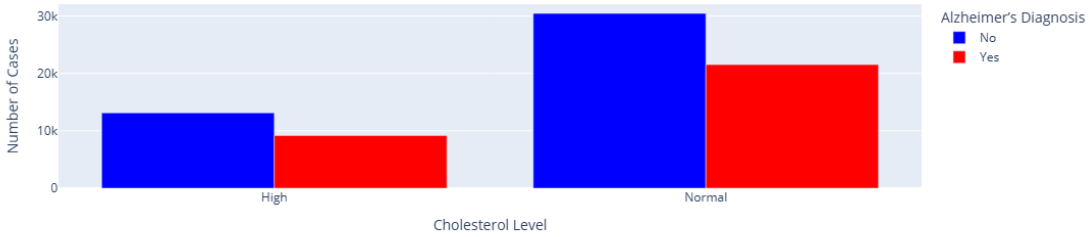
Alzheimer's Diagnosis by Diabetes

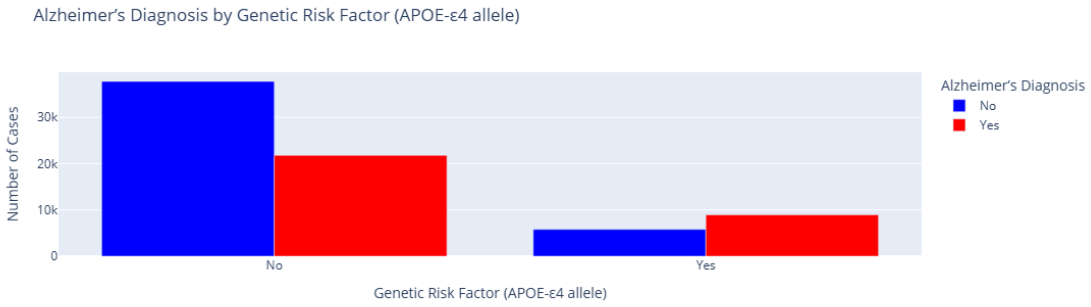


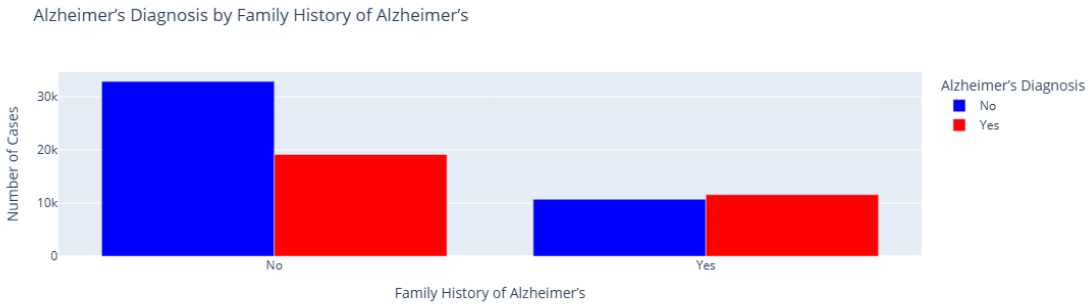
Alzheimer's Diagnosis by Hypertension



Alzheimer's Diagnosis by Cholesterol Level

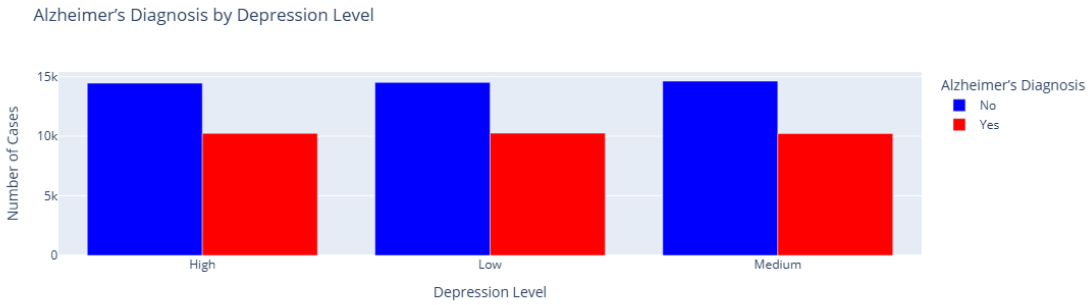


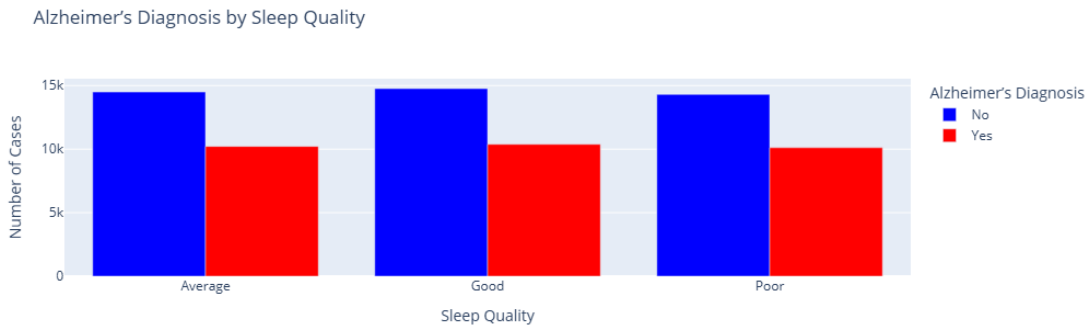


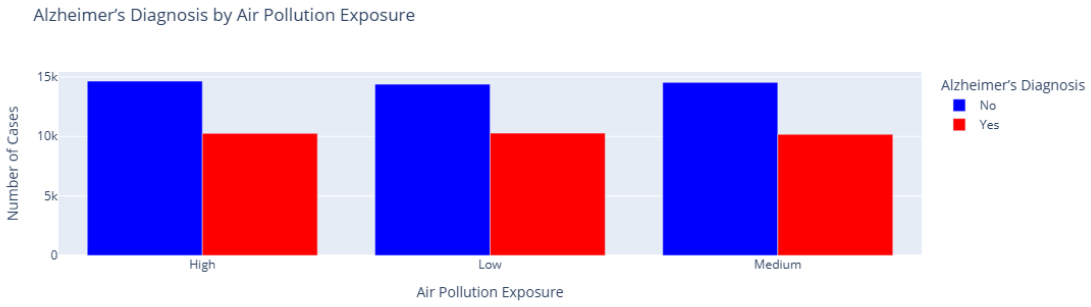


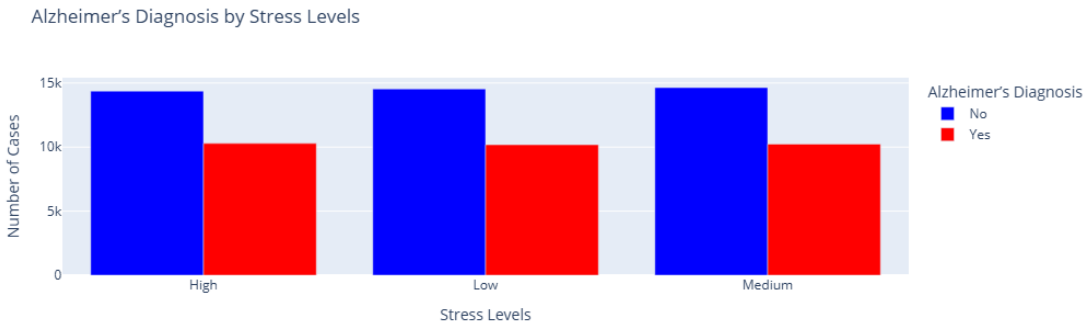
[111]:
```
# Bar Charts for Environmental & Psychological Factors

# List of environmental & psychological variables
env_psych_variables = ["Depression Level", "Sleep Quality", "Air Pollution␣
↪Exposure", "Stress Levels"]

# Loop through each factor and generate bar charts
for col in env_psych_variables:
    df_counts = df.groupby([col, "Alzheimer's Diagnosis"]).size().
↪reset_index(name="Count")

    fig = px.bar(df_counts,
                 x=col,
                 y="Count",
                 color="Alzheimer's Diagnosis",
                 title=f"Alzheimer's Diagnosis by {col}",
                 labels={"Count": "Number of Cases", col: col},
                 barmode="group",  # Side-by-side bars for comparison
                 color_discrete_map={"Yes": "red", "No": "blue"})  # Assign␣
↪colors
```

```
fig.show()
```


Alzheimer's Diagnosis by Depression Level
15k
10k
5k
0
Number of Cases
High
Low
Medium
Depression Level
Alzheimer's Diagnosis
No
Yes


Alzheimer's Diagnosis by Sleep Quality
15k
10k
5k
0
Number of Cases
Average
Good
Poor
Sleep Quality
Alzheimer's Diagnosis
No
Yes


Alzheimer's Diagnosis by Air Pollution Exposure
15k
10k
5k
0
Number of Cases
High
Low
Medium
Air Pollution Exposure
Alzheimer's Diagnosis
No
Yes


Alzheimer's Diagnosis by Stress Levels
15k
10k
5k
0
Number of Cases
High
Low
Medium
Stress Levels
Alzheimer's Diagnosis
No
Yes