In [53]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
import warnings
warnings.filterwarnings("ignore")

# Loading the dataset
df_thyroid = pd.read_csv('/Users/roshan/Downloads/thyroid_cancer_risk_data.c

# Print some data
print(df_thyroid.info())
print(df_thyroid.head())

# Check for missing values
print(df_thyroid.isnull().sum())

# Summary statistics
print(df_thyroid.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 212691 entries, 0 to 212690
Data columns (total 17 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   Patient_ID           212691 non-null  int64  
 1   Age                  212691 non-null  int64  
 2   Gender               212691 non-null  object 
 3   Country              212691 non-null  object 
 4   Ethnicity            212691 non-null  object 
 5   Family_History       212691 non-null  object 
 6   Radiation_Exposure   212691 non-null  object 
 7   Iodine_Deficiency    212691 non-null  object 
 8   Smoking              212691 non-null  object 
 9   Obesity              212691 non-null  object 
 10  Diabetes             212691 non-null  object 
 11  TSH_Level            212691 non-null  float64
 12  T3_Level             212691 non-null  float64
 13  T4_Level             212691 non-null  float64
 14  Nodule_Size          212691 non-null  float64
 15  Thyroid_Cancer_Risk  212691 non-null  object 
 16  Diagnosis            212691 non-null  object 
dtypes: float64(4), int64(2), object(11)
memory usage: 27.6+ MB
None
   Patient_ID  Age  Gender  Country  Ethnicity Family_History  \
0           1   66    Male   Russia  Caucasian             No   
1           2   29    Male  Germany   Hispanic             No   
2           3   86    Male  Nigeria  Caucasian             No   
3           4   75  Female    India      Asian             No   
4           5   35  Female  Germany    African            Yes   

  Radiation_Exposure Iodine_Deficiency Smoking Obesity Diabetes  TSH_Level
\
0                Yes                No      No      No       No       9.37
1                Yes                No      No      No       No       1.83
2                 No                No      No      No       No       6.26
3                 No                No      No      No       No       4.10
4                Yes                No      No      No       No       9.10

   T3_Level  T4_Level  Nodule_Size Thyroid_Cancer_Risk Diagnosis  
0      1.67      6.16         1.08                 Low    Benign  
1      1.73     10.54         4.05                 Low    Benign  
2      2.59     10.57         4.61                 Low    Benign  
3      2.62     11.04         2.46              Medium    Benign  
4      2.11     10.71         2.11                High    Benign  
Patient_ID             0
Age                    0
Gender                 0
Country                0
Ethnicity              0
Family_History         0
Radiation_Exposure     0
Iodine_Deficiency      0
Smoking                0
Obesity                0
```

```
Diabetes              0
TSH_Level             0
T3_Level              0
T4_Level              0
Nodule_Size           0
Thyroid_Cancer_Risk   0
Diagnosis             0
dtype: int64
         Patient_ID            Age      TSH_Level       T3_Level  \
count  212691.00000  212691.000000  212691.000000  212691.000000
mean   106346.00000      51.918497       5.045102       2.001727
std     61398.74739      21.632815       2.860264       0.866248
min         1.00000      15.000000       0.100000       0.500000
25%     53173.50000      33.000000       2.570000       1.250000
50%    106346.00000      52.000000       5.040000       2.000000
75%    159518.50000      71.000000       7.520000       2.750000
max    212691.00000      89.000000      10.000000       3.500000

            T4_Level    Nodule_Size
count  212691.000000  212691.000000
mean        8.246204       2.503403
std         2.164188       1.444631
min         4.500000       0.000000
25%         6.370000       1.250000
50%         8.240000       2.510000
75%        10.120000       3.760000
max        12.000000       5.000000
```

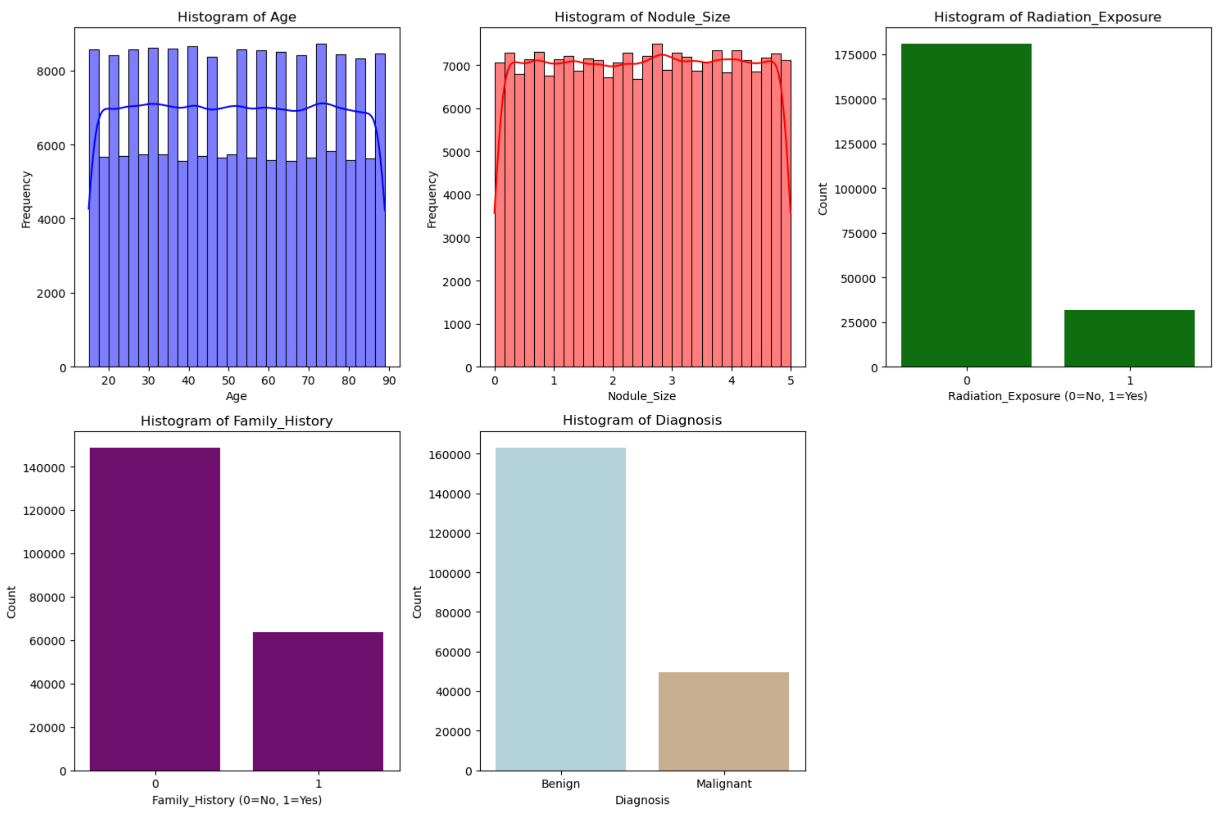
In [13]:
```
# Select relevant variables
selected_vars = ["Age", "Nodule_Size", "Radiation_Exposure", "Family_History

# Encode categorical variables
df_thyroid_encoded = df_thyroid.copy()
df_thyroid_encoded["Radiation_Exposure"] = df_thyroid_encoded["Radiation_Exp
df_thyroid_encoded["Family_History"] = df_thyroid_encoded["Family_History"].
df_thyroid_encoded["Diagnosis"] = df_thyroid_encoded["Diagnosis"].map({"Beni

# Compute descriptive statistics
desc_stats = df_thyroid_encoded[selected_vars].describe()

# Compute mode for each variable
mode_values = df_thyroid_encoded[selected_vars].mode().iloc[0]

# Append mode row to descriptive statistics
desc_stats.loc["Mode"] = mode_values
# Print descriptive statistics
print("### Descriptive Statistics: Mean, Mode, Spread, and Tails ###")
print(desc_stats)
```

```
### Descriptive Statistics: Mean, Mode, Spread, and Tails ###
                 Age    Nodule_Size  Radiation_Exposure  Family_History  \
count  212691.000000  212691.000000       212691.000000   212691.000000   
mean       51.918497       2.503403            0.149795        0.300083   
std        21.632815       1.444631            0.356871        0.458295   
min        15.000000       0.000000            0.000000        0.000000   
25%        33.000000       1.250000            0.000000        0.000000   
50%        52.000000       2.510000            0.000000        0.000000   
75%        71.000000       3.760000            0.000000        1.000000   
max        89.000000       5.000000            1.000000        1.000000   
Mode       72.000000       0.690000            0.000000        0.000000   

           Diagnosis  
count  212691.000000  
mean        0.232708  
std         0.422559  
min         0.000000  
25%         0.000000  
50%         0.000000  
75%         0.000000  
max         1.000000  
Mode        0.000000  
```

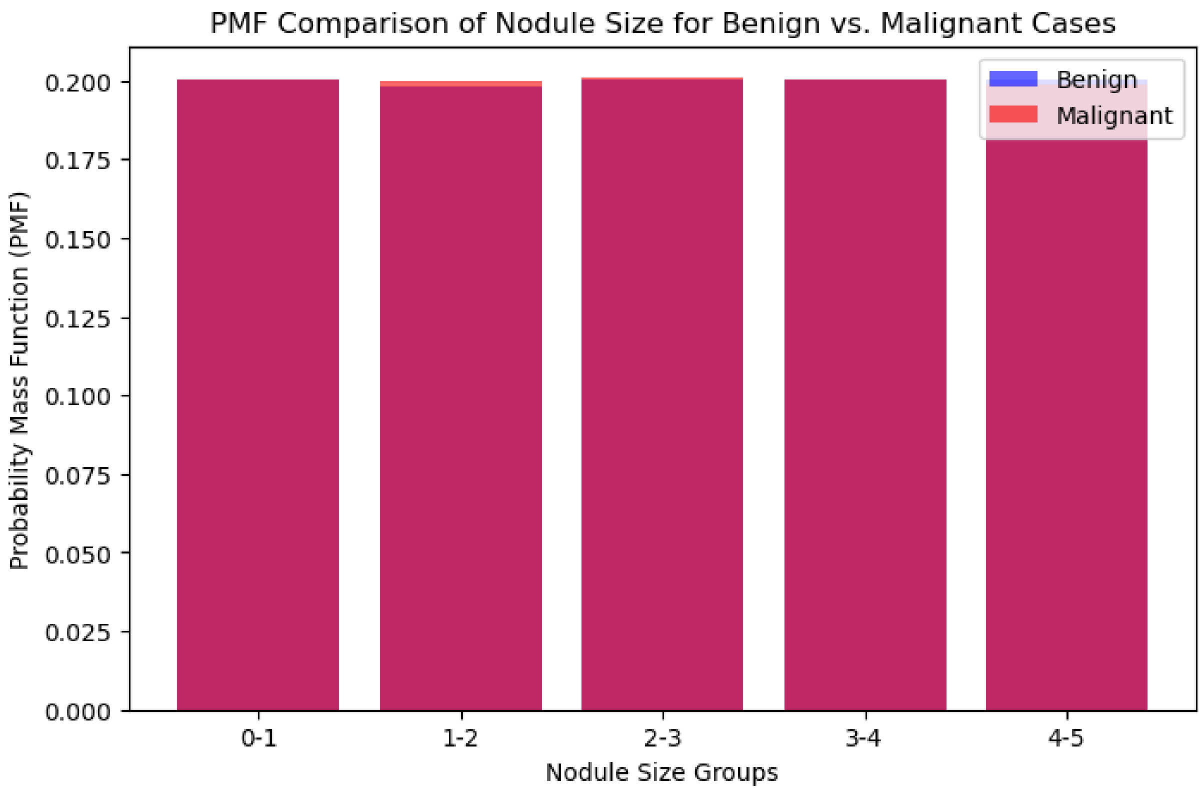
In [95]:
```
# Define selected variables for outlier detection (excluding Radiation_Expos
selected_vars_for_outliers = ["Age", "Nodule_Size", "Family_History"]

# Define function for IQR-based outlier detection
def detect_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1  # Compute IQR
    lower_bound = Q1 - 1.5 * IQR  # Lower bound for outliers
    upper_bound = Q3 + 1.5 * IQR  # Upper bound for outliers
    outliers = data[(data[column] < lower_bound) | (data[column] > upper_bou
    return Q1, Q3, IQR, lower_bound, upper_bound, len(outliers)

# Compute IQR and outliers for selected variables
iqr_results = {}
for col in selected_vars_for_outliers:
    Q1, Q3, IQR, lower_bound, upper_bound, num_outliers = detect_outliers_ic
    iqr_results[col] = {
        "Q1": Q1, "Q3": Q3, "IQR": IQR,
        "Lower Bound": lower_bound, "Upper Bound": upper_bound,
        "Num Outliers": num_outliers
    }

# Convert results to a DataFrame for better visualization
iqr_outlier_df = pd.DataFrame(iqr_results).T

# Print IQR-based Outlier Analysis
print("\n### IQR-Based Outlier Analysis (Excluding Radiation Exposure and Di
print(iqr_outlier_df)
```

```
### IQR-Based Outlier Analysis (Excluding Radiation Exposure and Diagnosis)
###
                  Q1     Q3    IQR  Lower Bound  Upper Bound  Num Outliers
Age            33.00  71.00  38.00      -24.000      128.000           0.0
Nodule_Size     1.25   3.76   2.51       -2.515        7.525           0.0
Family_History  0.00   1.00   1.00       -1.500        2.500           0.0
```

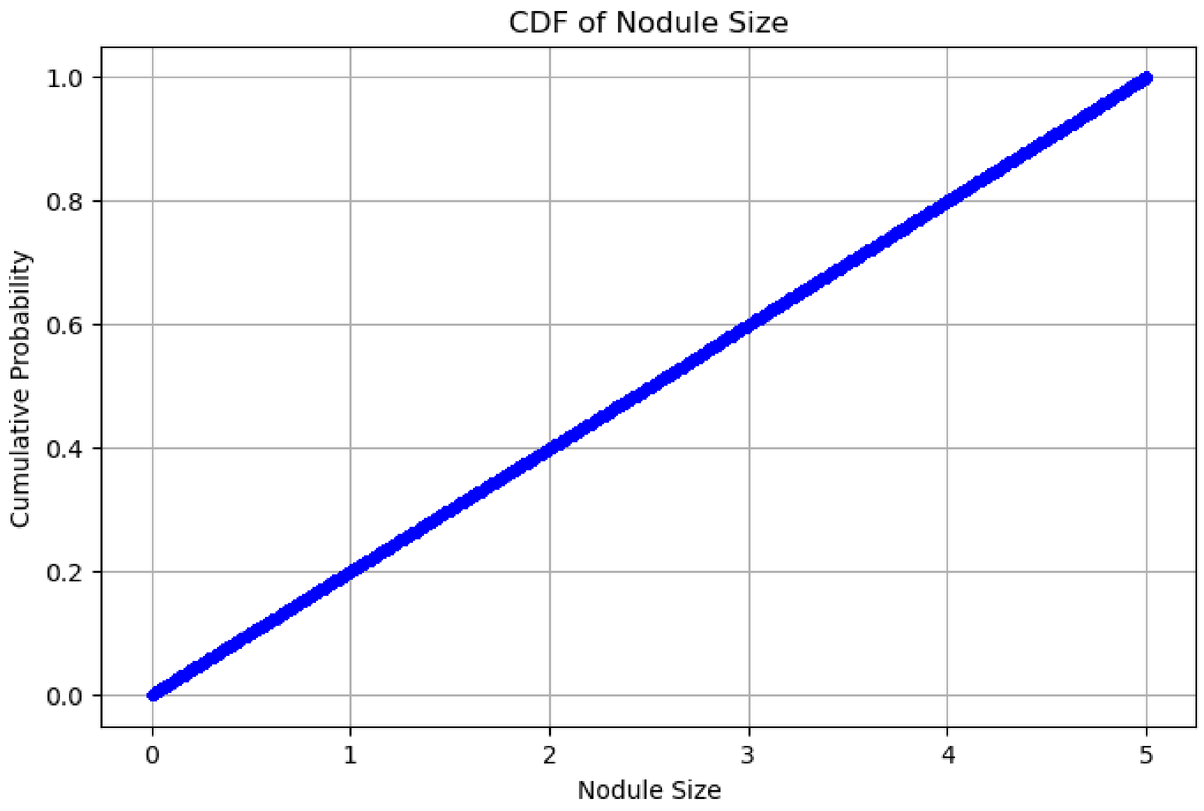
In [97]:
```
# Generate histograms

plt.figure(figsize=(15, 10))
colors = ["blue", "red", "green", "purple"]

for i, var in enumerate(selected_vars):
    plt.subplot(2, 3, i + 1)

    if var in ["Radiation_Exposure", "Family_History"]:
        sns.countplot(x=df_thyroid_encoded[var], palette=[colors[i % len(col
        plt.xlabel(f"{var} (0=No, 1=Yes)")
        plt.ylabel("Count")

    elif var == "Diagnosis":
        sns.countplot(x=df_thyroid_encoded[var], palette=["lightblue", "tan"
        plt.xlabel("Diagnosis")
        plt.ylabel("Count")
        plt.xticks(ticks=[0, 1], labels=["Benign", "Malignant"])

    else:  # Continuous variables (Age, Nodule Size)
        sns.histplot(df_thyroid_encoded[var], bins=30, kde=True, color=color
        plt.xlabel(var)
        plt.ylabel("Frequency")

    plt.title(f"Histogram of {var}")

plt.tight_layout()
plt.show()
```



```
In [99]: # PMF Analysis for Nodule Size
df_thyroid_encoded["Nodule_Size_Binned"] = pd.cut(df_thyroid_encoded["Nodule
pmf_benign = df_thyroid_encoded[df_thyroid_encoded["Diagnosis"] == 0]["Nodul
pmf_malignant = df_thyroid_encoded[df_thyroid_encoded["Diagnosis"] == 1]["No
plt.figure(figsize=(8, 5))
plt.bar(pmf_benign.index, pmf_benign, alpha=0.6, label="Benign", color="blue
plt.bar(pmf_malignant.index, pmf_malignant, alpha=0.6, label="Malignant", co
plt.xlabel("Nodule Size Groups")
plt.ylabel("Probability Mass Function (PMF)")
plt.title("PMF Comparison of Nodule Size for Benign vs. Malignant Cases")
plt.legend()
plt.show()
```

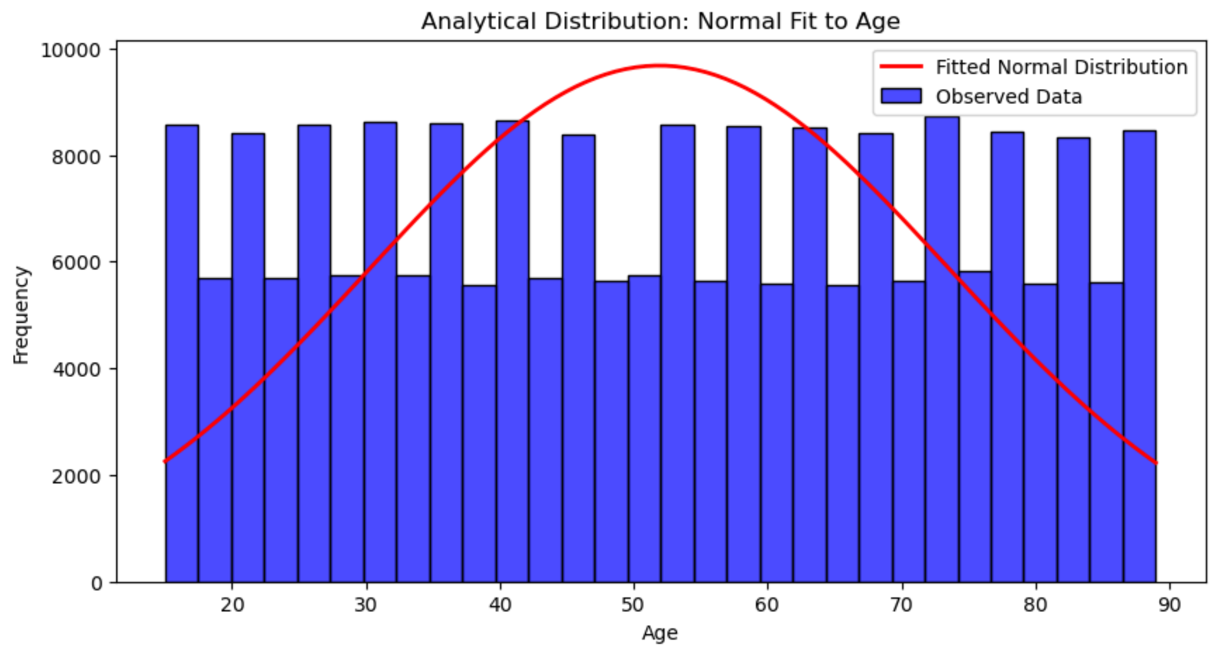


In [101…
```
# CDF Analysis
sorted_nodule_size = np.sort(df_thyroid_encoded["Nodule_Size"])
cdf_nodule_size = np.arange(1, len(sorted_nodule_size) + 1) / len(sorted_noc
plt.figure(figsize=(8, 5))
plt.plot(sorted_nodule_size, cdf_nodule_size, marker=".", linestyle="none",
plt.xlabel("Nodule Size")
plt.ylabel("Cumulative Probability")
plt.title("CDF of Nodule Size")
plt.grid(True)
plt.show()
```



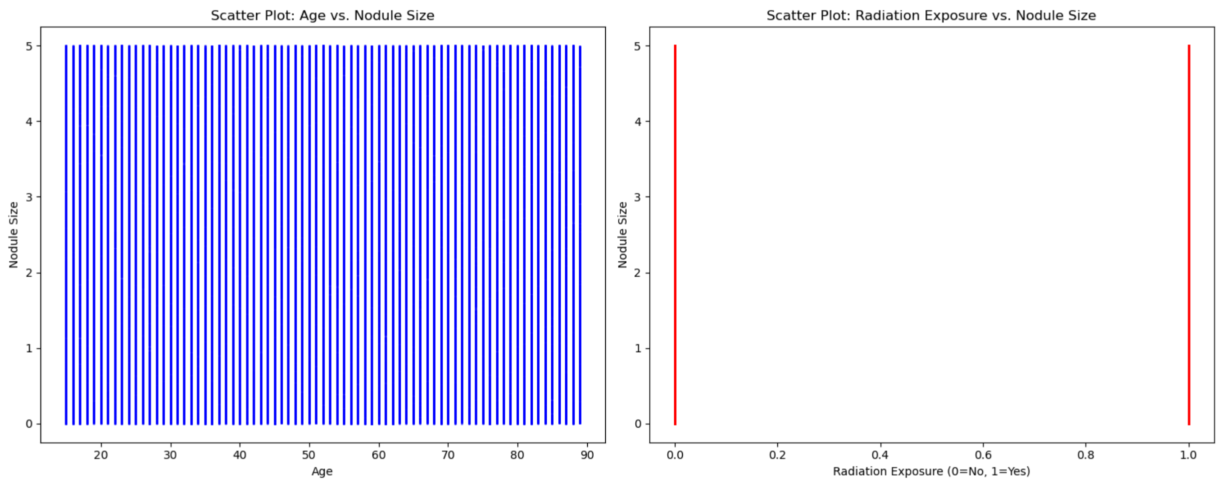
In [103…

```
# Analytical Distribution: Normal Fit to Age

# Fit a normal distribution to Age
mu, sigma = np.mean(df_thyroid_encoded["Age"]), np.std(df_thyroid_encoded["A

# Generate normal distribution curve
x = np.linspace(min(df_thyroid_encoded["Age"]), max(df_thyroid_encoded["Age"
pdf = stats.norm.pdf(x, mu, sigma) * len(df_thyroid_encoded["Age"]) * np.dif

# Plot histogram of observed Age data
plt.figure(figsize=(10, 5))
sns.histplot(df_thyroid_encoded["Age"], bins=30, kde=False, color="blue", al

# Overlay the fitted normal distribution curve
plt.plot(x, pdf, color="red", linewidth=2, label="Fitted Normal Distributior

# Titles and labels
plt.title("Analytical Distribution: Normal Fit to Age")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.legend()

# Show plot
plt.show()
```



In [105…

```
# scatter plots
# Define figure size
fig, axes = plt.subplots(1, 2, figsize=(15, 6))

# Scatter plot for Age vs. Nodule Size
axes[0].scatter(df_thyroid_encoded["Age"], df_thyroid_encoded["Nodule_Size"]
axes[0].set_xlabel("Age")
axes[0].set_ylabel("Nodule Size")
axes[0].set_title("Scatter Plot: Age vs. Nodule Size")

# Scatter plot for Radiation Exposure vs. Nodule Size
axes[1].scatter(df_thyroid_encoded["Radiation_Exposure"], df_thyroid_encodec
axes[1].set_xlabel("Radiation Exposure (0=No, 1=Yes)")
axes[1].set_ylabel("Nodule Size")
axes[1].set_title("Scatter Plot: Radiation Exposure vs. Nodule Size")

# Adjust layout and display
plt.tight_layout()
plt.show()
```



In [107…
```
# Compute Pearson Correlation Coefficient
selected_vars = ["Age", "Nodule_Size", "Radiation_Exposure", "Family_History
pearson_corr_all = df_thyroid_encoded[selected_vars].corr(method="pearson")

# Print Pearson Correlation Coefficient Matrix
print("\n### Pearson Correlation Coefficient Matrix (All Selected Variables)
print(pearson_corr_all)
```

```
### Pearson Correlation Coefficient Matrix (All Selected Variables) ###
                         Age  Nodule_Size  Radiation_Exposure  Family_Histor
y  \
Age                 1.000000    -0.001489            0.004007        0.00333
7
Nodule_Size        -0.001489     1.000000            0.000118       -0.00287
7
Radiation_Exposure  0.004007     0.000118            1.000000       -0.00079
5
Family_History      0.003337    -0.002877           -0.000795        1.00000
0
Diagnosis           0.000115    -0.002658            0.089043        0.14092
1

                    Diagnosis
Age                  0.000115
Nodule_Size         -0.002658
Radiation_Exposure   0.089043
Family_History       0.140921
Diagnosis            1.000000
```

In [109…
```
# Hypothesis Analysis Chi-Square Test for Radiation Exposure vs. Diagnosis
contingency_table = pd.crosstab(df_thyroid_encoded["Radiation_Exposure"], df
chi2, p, _, _ = stats.chi2_contingency(contingency_table)
print("\n --- Chi-Square Test Results --- ")
print(f"Chi-Square Statistic: {chi2}, p-value: {p}")
```

```
 --- Chi-Square Test Results --- 
Chi-Square Statistic: 1685.7536394973827, p-value: 0.0
```

In [111…
```
# Logistic Regression Analysis
X = sm.add_constant(df_thyroid_encoded[["Age", "Nodule_Size", "Radiation_Exp
y = df_thyroid_encoded["Diagnosis"]
logit_model = sm.Logit(y, X).fit()
print("\n ----Logistic Regression Summary ---")
print(logit_model.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.529167
         Iterations 5

 ----Logistic Regression Summary ---
                           Logit Regression Results                           
============================================================================
==
Dep. Variable:              Diagnosis   No. Observations:               2126
91
Model:                          Logit   Df Residuals:                   2126
86
Method:                           MLE   Df Model:                           
4
Date:                Sun, 02 Mar 2025   Pseudo R-squ.:                 0.024
63
Time:                        22:26:21   Log-Likelihood:          -1.1255e+
05
converged:                       True   LL-Null:                 -1.1539e+
05
Covariance Type:            nonrobust   LLR p-value:                     0.0
00
============================================================================
==========
                         coef    std err          z      P>|z|      [0.025      
0.975]
----------------------------------------------------------------------------
----------
const                 -1.5072      0.017    -88.944      0.000      -1.540      
-1.474
Age                -8.448e-05      0.000     -0.351      0.726      -0.001      
0.000
Nodule_Size           -0.0038      0.004     -1.042      0.297      -0.011      
0.003
Radiation_Exposure     0.5535      0.013     41.194      0.000       0.527      
0.580
Family_History         0.6953      0.011     64.643      0.000       0.674      
0.716
============================================================================
==========
```

In [ ]: