Assignment 1

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats
from matplotlib import pyplot as plt
```

1.

a. The average reaction time in the Stroop test of 12 participants in a neurocognitive study is {590, 748, 579, 544, 570, 598, 599, 673, 635, 714, 580, 565} in milliseconds. Is the reaction time of this cohort unusual compared to the standard reaction time of 570 milliseconds at a significance level of 0.05? What are the sample mean, standard deviation, t-statistic, degree of freedom, and p-value?

In this scenario, we want to perform a two-tailed one-sample t-test where the null hypothesis is

$$H_0 : \mu = \mu_0$$

and the alternative hypothesis is

$$H_A : \mu \neq \mu_0$$

In [2]:
```
samples = np.array([590, 748, 579, 544, 570, 598, 599, 673, 635, 714, 580, 565])

# number of samples
N = len(samples)
# alpha
alpha = 0.05
# standard reaction time
mu0 = 570
```

For one sample t-test, t is given by

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

where $\bar{x}$ is the sample mean, s is the sample standard deviation given by $s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$, μ_0 is the specified value of the population mean under the null hypothesis, and n is the sample size.

In [3]:
```
# sample mean
x = samples.mean()
# sample std
s = samples.std(ddof=1)
print('Sample mean: ', x, '\nSample STD: ', s)
```

```
Sample mean:  616.25
Sample STD:  63.59834903517543
```

In [4]:
```
# t-statistic
t = (x - mu0) / (s/np.sqrt(N))
print('T-statistic: ', t)
```

```
T-statistic:  2.519164445786296
```

In one sample t-test, the degrees of freedom are $n - 1$

In [5]:
```
# degree of freedom
df = N - 1
print('DOF: ', df)
```

```
DOF:  11
```

In [6]:
```
# one sample t test
stats.ttest_1samp(samples, mu0, nan_policy='raise', alternative='two-sided')
```

Out[6]:
```
TtestResult(statistic=2.5191644457862963, pvalue=0.02851786955873765, df=11)
```

The reaction time of the cohort is significant compared to the standard reaction time (p < 0.05).

- **sample mean: 616.25**
- **sample standard deviation: 63.589**
- **t-statistic: 2.519**
- **degree of freedom: 11**
- **p-value: 0.0285**

b. The Stroop test was then separately taken for another cohort of 8 participants {570, 535, 606, 572, 568, 600, 554, 575}. Are the two cohorts significantly different in the reaction time based on two-tailed two-sample t-test? What about the significance under one-tailed test?

Here, we want to perform a two-sample t-test where sample sizes are unequal and variances are assumed to be similar. This is given by

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where

$$s_p = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{n_1 + n_2 - 2}}$$

is the pooled standard deviation.

In [7]:
```
samples2 = np.array([570, 535, 606, 572, 568, 600, 554, 575])

# sample size of 2nd batch of samples
N2 = len(samples2)
```

In [8]:
```
# pooled standard deviation
sp = np.sqrt(((N-1)*samples.var(ddof=1) + (N2-1)*samples2.var(ddof=1))/(N + N2 - 2))
# t statistic
```

```
t = (samples.mean() - samples2.mean()) / (sp * np.sqrt((1/N) + (1/N2)))
print(t, sp)
```

```
1.8532185387550502 51.721610570437576
```

In [9]:
```
# two sample t test assuming similar variances (two tailed)
stats.ttest_ind(samples, samples2, axis=0, equal_var=True, nan_policy='raise', alternati
```

Out[9]:
```
Ttest_indResult(statistic=1.8532185387550504, pvalue=0.08031716997740455)
```

In [10]:
```
# compare means of the 2 samples
print(samples.mean(), samples2.mean())
```

```
616.25 572.5
```

The null hypothesis is $H_0 : \mu_1 = \mu_2$. Alternative hypothesis is $H_A : \mu_1 > \mu_2$.

In [11]:
```
# two sample t test (one tailed)
stats.ttest_ind(samples, samples2, axis=0, equal_var=True, nan_policy='raise', alternati
```

Out[11]:
```
Ttest_indResult(statistic=1.8532185387550504, pvalue=0.04015858498870228)
```

The two groups are not significantly different based on a two-tailed test but are significantly different based on a one-tailed test.

2. A recently proposed machine learning model was applied to an MRI dataset to classify the binary diagnosis label of 100 subjects of age 40 years, resulting in 85 subjects correctly classified and 15 subjects wrongly classified. For the following three questions, what are the chi-squared statistic and p-value?

In [12]:
```
# marginal frequency
N = 100
# contingency table
df = pd.DataFrame([[85, 15]], columns=['correct', 'incorrect'], index=['40 yo'])
df
```

Out[12]:

	correct	incorrect
40 yo	85	15

a. Is the proposed classifier significantly more accurate than random guessing?

Random guessing will result in uniform probability for each class

$$p_i = 1/K$$

where K is the number of classes. In our case $K = 2$.

We can get the expected value (frequencies) of random guessing:

$$m_i = p_i \times N$$

where m_i is the expected value of class $k = i$ and N is the marginal frequency.

In [13]:
```
# number of classes
K = 2
# probabilities for each class
probs = [1/K for p in ['correct', 'incorrect']]
# expected frequency when guessing randomly
df_expected = pd.DataFrame([[p*N for p in probs]], columns=['correct', 'incorrect'], ind
df_expected
```

Out[13]:

	correct	incorrect
40 yo	50.0	50.0

X^2 is given by

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - m_i)^2}{m_i}$$

In [14]:
```
# calculate chi squared
chisq = (np.square(df.loc['40 yo'] - df_expected.loc['40 yo'])/ df_expected.loc['40 yo']
print('Chi-squared: ', chisq)
```

```
Chi-squared:  49.0
```

In [15]:
```
# chi squared test with DOF = K-1 = 1
stats.chisquare(f_obs=df.loc['40 yo'], f_exp=df_expected.loc['40 yo'], ddof=0)
```

Out[15]:
```
Power_divergenceResult(statistic=49.0, pvalue=2.559625087771672e-12)
```

The proposed classifier is significantly more accurate than random guessing.

- **Chi-squared: 49**
- **p-value: 2.56e-12**

b. Is the proposed classifier significantly more accurate than the expected accuracy of 75% from a baseline classifier at a significance level of 0.05?

In [16]:
```
# probabilities for each class
probs = [0.75, 0.25]
# expected frequency
df_expected = pd.DataFrame([[p*N for p in probs]], columns=['correct', 'incorrect'], ind
df_expected
```

Out[16]:

	correct	incorrect
40 yo	75.0	25.0

In [17]:
```
# calculate chi squared
chisq = (np.square(df.loc['40 yo'] - df_expected.loc['40 yo'])/ df_expected.loc['40 yo']
print('Chi-squared: ', chisq)
```

```
Chi-squared:  5.333333333333333
```

In [18]:
```
# chi squared test with DOF = K-1 = 1
stats.chisquare(f_obs=df.loc['40 yo'], f_exp=df_expected.loc['40 yo'], ddof=0)
```

Out[18]:
```
Power_divergenceResult(statistic=5.333333333333333, pvalue=0.020921335337794035)
```

The proposed classifier is significantly more accurate than expected accuracy of 75%.

- **Chi-squared: 5.33**
- **p-value: 0.021**

c. Applying the proposed classifier to another 50 subjects of age 60 years resulted in 37 correct classifications and 13 wrong classifications. Does the classifier have significantly different predictive power between the two ages?

In [19]:
```
# contingency table
df = pd.DataFrame([[85, 15], [37, 13]], columns=['correct', 'incorrect'], index=['40 yo'
df
```

Out[19]:

	correct	incorrect
40 yo	85	15
60 yo	37	13

In [20]:
```
# expected frequency
df_expected = df.copy()
vals = df.values
tot = df.sum().sum()
for i in range(len(vals[0,:])):
    row_tot = vals[i,:].sum()
    for j in range(len(vals[:,0])):
        col_tot = vals[:,j].sum()
        df_expected.iloc[i,j] = row_tot * col_tot / tot
df_expected
```

Out[20]:

	correct	incorrect
40 yo	81.333333	18.666667
60 yo	40.666667	9.333333

In [21]:
```
# calculate chi squared
chisq = np.array(np.square(df - df_expected)/ df_expected).sum(axis=None)
print('Chi-squared: ', chisq)
```

```
Chi-squared:  2.656615925058548
```

In [22]:
```
# chi squared test with DOF = (R-1)(C-1) = 1
stats.chi2_contingency(df, correction=False, lambda_=None)
```

Out[22]:
```
Chi2ContingencyResult(statistic=2.656615925058548, pvalue=0.10311991569550351, dof=1, ex
pected_freq=array([[81.33333333, 18.66666667],
       [40.66666667,  9.33333333]]))
```

The proposed classifier does not have significantly different predictive power for the two ages.

- **Chi-squared: 2.66**
- **p-value: 0.1**

3. (Programming, data available at here) A new biomarker was extracted from 10 brain regions of 50 control participants and 50 patients. Given that the biomarker depends linearly on age and sex, please perform an analysis to test whether the disease group had significantly unusual values in the regional biomarkers compared to the control group and whether the disease effects differed between sexes (sex-disease interaction). Please also use appropriate plots to visualize the significant effects if there is any.

In [23]:
```
data = pd.read_csv('data_assignment_1.csv')
data
```

Out[23]:

	age	sex	diagnosis	Frontal_Sup	Frontal_Inf	Cingulum_Ant	Cingulum_Post	Parietal_Sup	Parietal_Inf
0	43.368493	0	0	0.541704	0.553985	0.577727	0.502631	0.539654	0.562739
1	36.654435	0	0	0.665915	0.477778	0.525422	0.473192	0.589977	0.656130
2	37.998386	0	0	0.385500	0.535842	0.651637	0.556026	0.451074	0.602841
3	36.640988	0	0	0.601210	0.500756	0.614601	0.581634	0.489078	0.615674
4	42.878145	0	0	0.529871	0.521045	0.576609	0.557342	0.494459	0.577652
...	...	...	...	...	...	...	...	...	...
95	45.966536	1	1	0.555299	0.597638	0.553309	0.523025	0.525431	0.667517
96	48.160286	1	1	0.524961	0.631617	0.382947	0.410978	0.505315	0.611400
97	32.339052	1	1	0.490657	0.652892	0.654747	0.656929	0.521088	0.702478
98	33.315738	1	1	0.584763	0.602108	0.465685	0.602878	0.693304	0.707188
99	32.630768	1	1	0.498238	0.661749	0.460143	0.661333	0.694017	0.742571

100 rows × 13 columns

In [24]:
```
import statsmodels.api as sm # Statistical models
from patsy import dmatrices
```

In [25]:
```
data
```

Out[25]:

	age	sex	diagnosis	Frontal_Sup	Frontal_Inf	Cingulum_Ant	Cingulum_Post	Parietal_Sup	Parietal_Inf
0	43.368493	0	0	0.541704	0.553985	0.577727	0.502631	0.539654	0.562739
1	36.654435	0	0	0.665915	0.477778	0.525422	0.473192	0.589977	0.656130
2	37.998386	0	0	0.385500	0.535842	0.651637	0.556026	0.451074	0.602841

3	36.640988	0	0	0.601210	0.500756	0.614601	0.581634	0.489078	0.615674
4	42.878145	0	0	0.529871	0.521045	0.576609	0.557342	0.494459	0.577652
...	...	...	...	...	...	...	...	...	...
95	45.966536	1	1	0.555299	0.597638	0.553309	0.523025	0.525431	0.667517
96	48.160286	1	1	0.524961	0.631617	0.382947	0.410978	0.505315	0.611400
97	32.339052	1	1	0.490657	0.652892	0.654747	0.656929	0.521088	0.702478
98	33.315738	1	1	0.584763	0.602108	0.465685	0.602878	0.693304	0.707188
99	32.630768	1	1	0.498238	0.661749	0.460143	0.661333	0.694017	0.742571

100 rows × 13 columns

GLM

$$Y = XB + \epsilon$$

where $Y \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{m \times x}$, $B \in \mathbb{R}^{x \times n}$, m is number of participants (i.e. $m = 100$), x is number of features, n is number of brain regions (i.e $n = 10$)

It is expected that there are following interactions: sex and diagnosis. Interaction terms are thus added.

In [26]:
```
B = pd.DataFrame([])
pvals = pd.DataFrame([])
for i in data.drop(columns=['age', 'sex', 'diagnosis']).columns:
    y, X = dmatrices(
        # interaction terms
        f'{i} ~ age + sex + diagnosis + sex:diagnosis',
        data=data,
        return_type='dataframe'
    )
    # fit GLM
    md = sm.GLM(y, X)
    md = md.fit()
    # get weights
    B[i] = md.params
    # get pvals
    pvals[i] = md.pvalues
    print(md.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:            Frontal_Sup   No. Observations:                  100
Model:                            GLM   Df Residuals:                       95
Model Family:                Gaussian   Df Model:                            4
Link Function:               Identity   Scale:                       0.0058220
Method:                          IRLS   Log-Likelihood:                117.98
Date:                Tue, 24 Oct 2023   Deviance:                      0.55309
Time:                        02:41:52   Pearson chi2:                    0.553
No. Iterations:                     3   Pseudo R-squ. (CS):             0.1764
Covariance Type:            nonrobust
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept         0.7449      0.060     12.422      0.000       0.627       0.862
age              -0.0045      0.001     -3.044      0.002      -0.007      -0.002
sex               0.0082      0.022      0.376      0.707      -0.035       0.051
diagnosis        -0.0470      0.022     -2.167      0.030      -0.089      -0.004
sex:diagnosis     0.0436      0.031      1.403      0.161      -0.017       0.105
================================================================================
```

Generalized Linear Model Regression Results

Dep. Variable:	Frontal_Inf	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0043983
Method:	IRLS	Log-Likelihood:	132.00
Date:	Tue, 24 Oct 2023	Deviance:	0.41784
Time:	02:41:52	Pearson chi2:	0.418
No. Iterations:	3	Pseudo R-squ. (CS):	0.3717
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.6912	0.052	13.260	0.000	0.589	0.793
age	-0.0042	0.001	-3.298	0.001	-0.007	-0.002
sex	0.0422	0.019	2.227	0.026	0.005	0.079
diagnosis	-0.0336	0.019	-1.781	0.075	-0.071	0.003
sex:diagnosis	0.0564	0.027	2.087	0.037	0.003	0.109

Generalized Linear Model Regression Results

Dep. Variable:	Cingulum_Ant	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0047251
Method:	IRLS	Log-Likelihood:	128.41
Date:	Tue, 24 Oct 2023	Deviance:	0.44888
Time:	02:41:52	Pearson chi2:	0.449
No. Iterations:	3	Pseudo R-squ. (CS):	0.4341
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.7624	0.054	14.112	0.000	0.656	0.868
age	-0.0046	0.001	-3.525	0.000	-0.007	-0.002
sex	0.0657	0.020	3.348	0.001	0.027	0.104
diagnosis	-0.0664	0.020	-3.399	0.001	-0.105	-0.028
sex:diagnosis	-0.0074	0.028	-0.264	0.792	-0.062	0.047

Generalized Linear Model Regression Results

Dep. Variable:	Cingulum_Post	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0039557
Method:	IRLS	Log-Likelihood:	137.30
Date:	Tue, 24 Oct 2023	Deviance:	0.37579
Time:	02:41:52	Pearson chi2:	0.376
No. Iterations:	3	Pseudo R-squ. (CS):	0.7351
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.8367	0.049	16.927	0.000	0.740	0.934
age	-0.0080	0.001	-6.659	0.000	-0.010	-0.006
sex	0.0530	0.018	2.949	0.003	0.018	0.088
diagnosis	-0.0852	0.018	-4.767	0.000	-0.120	-0.050
sex:diagnosis	0.0861	0.026	3.361	0.001	0.036	0.136

Generalized Linear Model Regression Results

Dep. Variable:	Parietal_Sup	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0034592

Method:	IRLS	Log-Likelihood:	144.01
Date:	Tue, 24 Oct 2023	Deviance:	0.32862
Time:	02:41:52	Pearson chi2:	0.329
No. Iterations:	3	Pseudo R-squ. (CS):	0.3254
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.7221	0.046	15.622	0.000	0.632	0.813
age	-0.0047	0.001	-4.145	0.000	-0.007	-0.002
sex	0.0533	0.017	3.170	0.002	0.020	0.086
diagnosis	0.0159	0.017	0.952	0.341	-0.017	0.049
sex:diagnosis	-0.0021	0.024	-0.087	0.931	-0.049	0.045

Generalized Linear Model Regression Results

Dep. Variable:	Parietal_Inf	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0040872
Method:	IRLS	Log-Likelihood:	135.67
Date:	Tue, 24 Oct 2023	Deviance:	0.38828
Time:	02:41:52	Pearson chi2:	0.388
No. Iterations:	3	Pseudo R-squ. (CS):	0.2707
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.7129	0.050	14.188	0.000	0.614	0.811
age	-0.0033	0.001	-2.722	0.006	-0.006	-0.001
sex	0.0765	0.018	4.191	0.000	0.041	0.112
diagnosis	0.0009	0.018	0.047	0.962	-0.035	0.036
sex:diagnosis	-0.0310	0.026	-1.189	0.235	-0.082	0.020

Generalized Linear Model Regression Results

Dep. Variable:	Occipital_Sup	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0038365
Method:	IRLS	Log-Likelihood:	138.83
Date:	Tue, 24 Oct 2023	Deviance:	0.36447
Time:	02:41:52	Pearson chi2:	0.364
No. Iterations:	3	Pseudo R-squ. (CS):	0.3184
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.7347	0.049	15.093	0.000	0.639	0.830
age	-0.0035	0.001	-2.924	0.003	-0.006	-0.001
sex	0.0271	0.018	1.530	0.126	-0.008	0.062
diagnosis	-0.0585	0.018	-3.322	0.001	-0.093	-0.024
sex:diagnosis	0.0471	0.025	1.868	0.062	-0.002	0.097

Generalized Linear Model Regression Results

Dep. Variable:	Occipital_Inf	No. Observations:	100
Model:	GLM	Df Residuals:	95
Model Family:	Gaussian	Df Model:	4
Link Function:	Identity	Scale:	0.0045174
Method:	IRLS	Log-Likelihood:	130.66
Date:	Tue, 24 Oct 2023	Deviance:	0.42916
Time:	02:41:52	Pearson chi2:	0.429
No. Iterations:	3	Pseudo R-squ. (CS):	0.4349
Covariance Type:	nonrobust		

```
                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept         0.6833      0.053     12.935      0.000       0.580       0.787
age              -0.0035      0.001     -2.717      0.007      -0.006      -0.001
sex               0.0648      0.019      3.375      0.001       0.027       0.102
diagnosis        -0.0380      0.019     -1.989      0.047      -0.075      -0.001
sex:diagnosis     0.0482      0.027      1.762      0.078      -0.005       0.102
==============================================================================
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:           Temporal_Sup   No. Observations:                  100
Model:                            GLM   Df Residuals:                       95
Model Family:                Gaussian   Df Model:                            4
Link Function:               Identity   Scale:                       0.0042819
Method:                          IRLS   Log-Likelihood:                133.34
Date:                Tue, 24 Oct 2023   Deviance:                     0.40678
Time:                        02:41:52   Pearson chi2:                   0.407
No. Iterations:                     3   Pseudo R-squ. (CS):            0.3700
Covariance Type:            nonrobust
==============================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept         0.7538      0.051     14.657      0.000       0.653       0.855
age              -0.0065      0.001     -5.198      0.000      -0.009      -0.004
sex               0.0555      0.019      2.969      0.003       0.019       0.092
diagnosis         0.0107      0.019      0.575      0.565      -0.026       0.047
sex:diagnosis    -0.0014      0.027     -0.054      0.957      -0.054       0.051
==============================================================================
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:           Temporal_Inf   No. Observations:                  100
Model:                            GLM   Df Residuals:                       95
Model Family:                Gaussian   Df Model:                            4
Link Function:               Identity   Scale:                       0.0047164
Method:                          IRLS   Log-Likelihood:                128.51
Date:                Tue, 24 Oct 2023   Deviance:                     0.44806
Time:                        02:41:52   Pearson chi2:                   0.448
No. Iterations:                     3   Pseudo R-squ. (CS):            0.4216
Covariance Type:            nonrobust
==============================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept         0.7681      0.054     14.230      0.000       0.662       0.874
age              -0.0064      0.001     -4.891      0.000      -0.009      -0.004
sex               0.0794      0.020      4.047      0.000       0.041       0.118
diagnosis         0.0058      0.020      0.297      0.767      -0.032       0.044
sex:diagnosis    -0.0070      0.028     -0.249      0.804      -0.062       0.048
==============================================================================
```

In [27]:
```
B
```

Out[27]:

	Frontal_Sup	Frontal_Inf	Cingulum_Ant	Cingulum_Post	Parietal_Sup	Parietal_Inf	Occipital_Sup	Oc
Intercept	0.744909	0.691166	0.762385	0.836728	0.722108	0.712912	0.734713	
age	-0.004452	-0.004193	-0.004645	-0.008029	-0.004674	-0.003336	-0.003473	
sex	0.008196	0.042190	0.065746	0.052982	0.053261	0.076543	0.027081	
diagnosis	-0.046995	-0.033565	-0.066402	-0.085204	0.015909	0.000862	-0.058479	
sex:diagnosis	0.043603	0.056382	-0.007382	0.086083	-0.002074	-0.030950	0.047120	

In [28]:
```
pvals
```

Out[28]:

	Frontal_Sup	Frontal_Inf	Cingulum_Ant	Cingulum_Post	Parietal_Sup	Parietal_Inf	Occipital_Sup	Oc
Intercept	1.992517e-35	3.934482e-40	3.211762e-45	2.836827e-64	5.176871e-55	1.079899e-45	1.813665e-51	2
age	2.337985e-03	9.751071e-04	4.238395e-04	2.759998e-11	3.390942e-05	6.492969e-03	3.450384e-03	6
sex	7.069391e-01	2.596235e-02	8.143245e-04	3.192142e-03	1.525511e-03	2.779913e-05	1.259159e-01	7
diagnosis	3.022509e-02	7.495385e-02	6.764950e-04	1.873344e-06	3.412243e-01	9.621529e-01	8.938022e-04	4
sex:diagnosis	1.605763e-01	3.684532e-02	7.920126e-01	7.775451e-04	9.309968e-01	2.345575e-01	6.177499e-02	7

Before making conclusions for whether diagnosis results in significant difference in biomarkers, we need to correct for multiple testing using the Benjamini-Hochberg procedure. Here, we set the false discovery rate to 0.05.

In [29]:
```
FDR = 0.05
```

In [30]:
```
df_bh = pvals.loc['diagnosis'].rename('pvals').to_frame()
df_bh
```

Out[30]:

	pvals
Frontal_Sup	0.030225
Frontal_Inf	0.074954
Cingulum_Ant	0.000676
Cingulum_Post	0.000002
Parietal_Sup	0.341224
Parietal_Inf	0.962153
Occipital_Sup	0.000894
Occipital_Inf	0.046726
Temporal_Sup	0.565483
Temporal_Inf	0.766704

In [31]:
```
# sort by pval
df_bh = df_bh.sort_values('pvals')
# rank
df_bh['rank'] = df_bh.rank()
# get BH critical value
df_bh['crit'] = df_bh.apply(lambda x: (x['rank']/len(df_bh))*FDR, axis=1)

df_bh
```

Out[31]:

	pvals	rank	crit
Cingulum_Post	0.000002	1.0	0.005
Cingulum_Ant	0.000676	2.0	0.010
Occipital_Sup	0.000894	3.0	0.015
Frontal_Sup	0.030225	4.0	0.020

Occipital_Inf	0.046726	5.0	0.025
Frontal_Inf	0.074954	6.0	0.030
Parietal_Sup	0.341224	7.0	0.035
Temporal_Sup	0.565483	8.0	0.040
Temporal_Inf	0.766704	9.0	0.045
Parietal_Inf	0.962153	10.0	0.050

In [32]:
```
# get highest rank where pval < crit
rank_max = df_bh[df_bh['pvals'] < df_bh['crit']]['rank'].max()
# all significant pvals
df_bh[df_bh['rank'] <= rank_max]
```

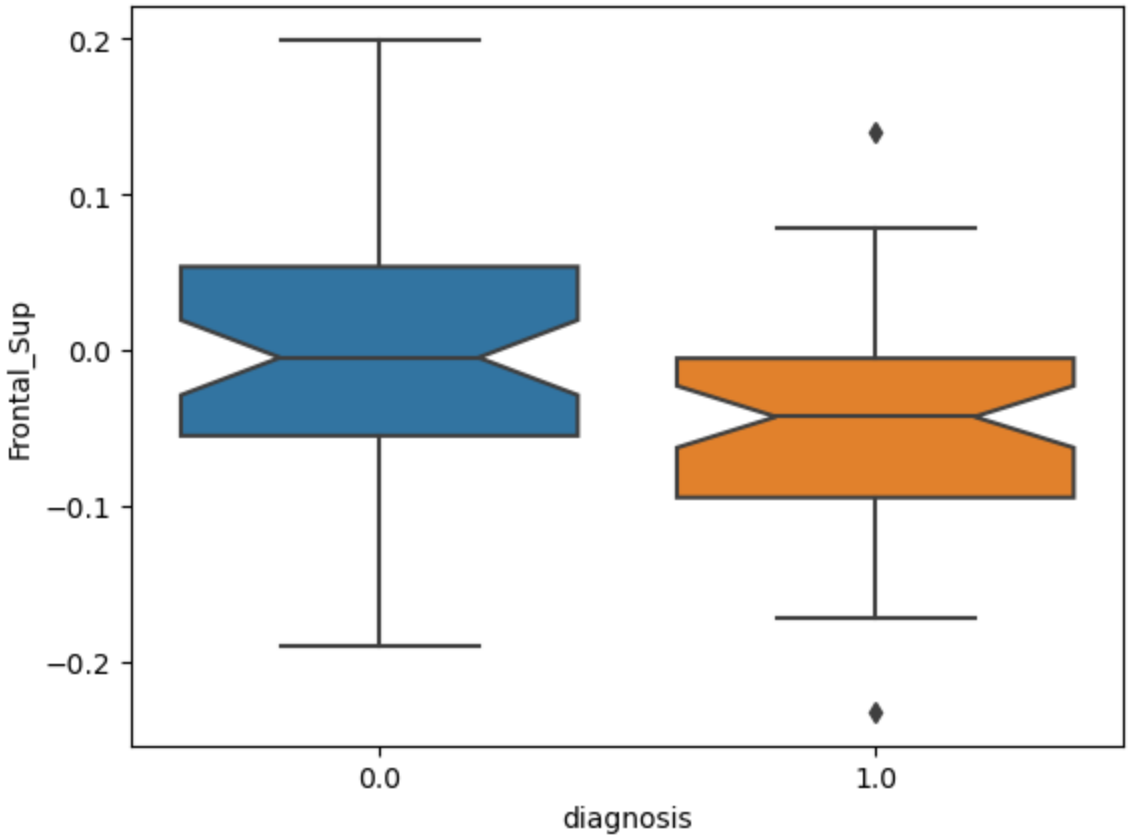
Out[32]:

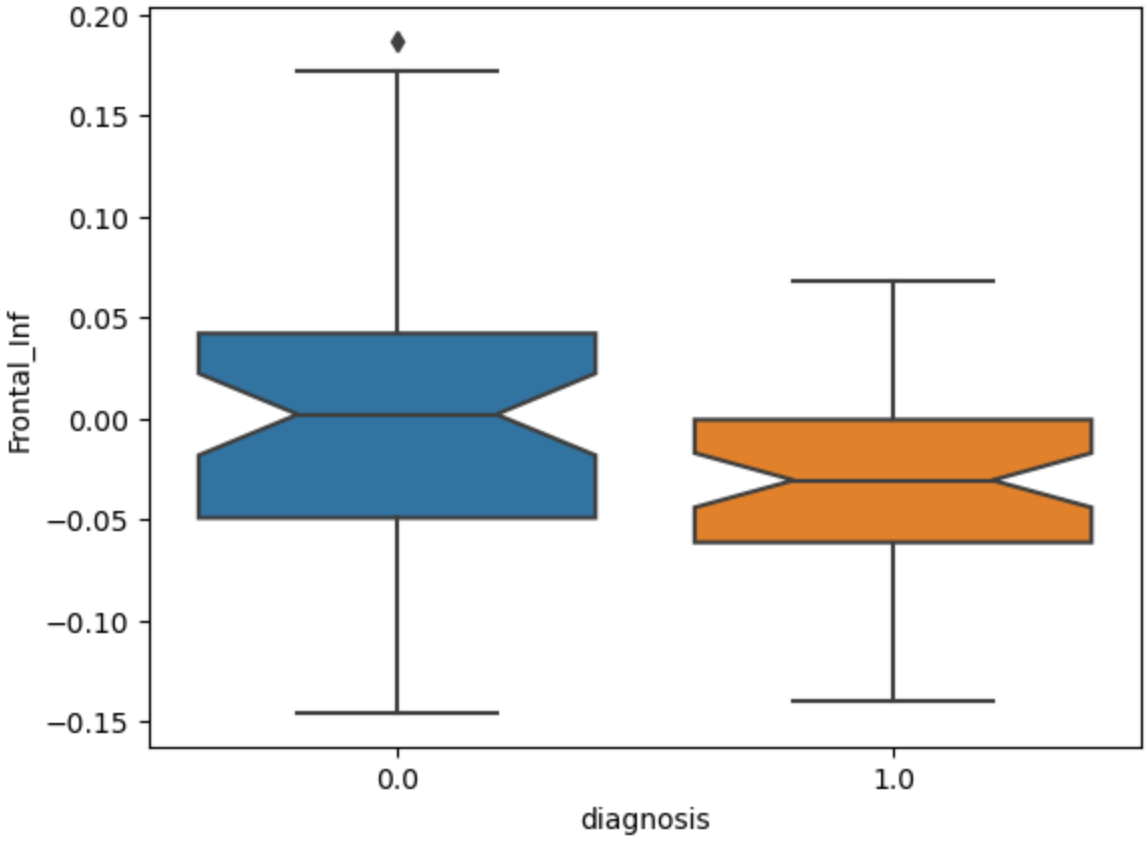
	pvals	rank	crit
Cingulum_Post	0.000002	1.0	0.005
Cingulum_Ant	0.000676	2.0	0.010
Occipital_Sup	0.000894	3.0	0.015

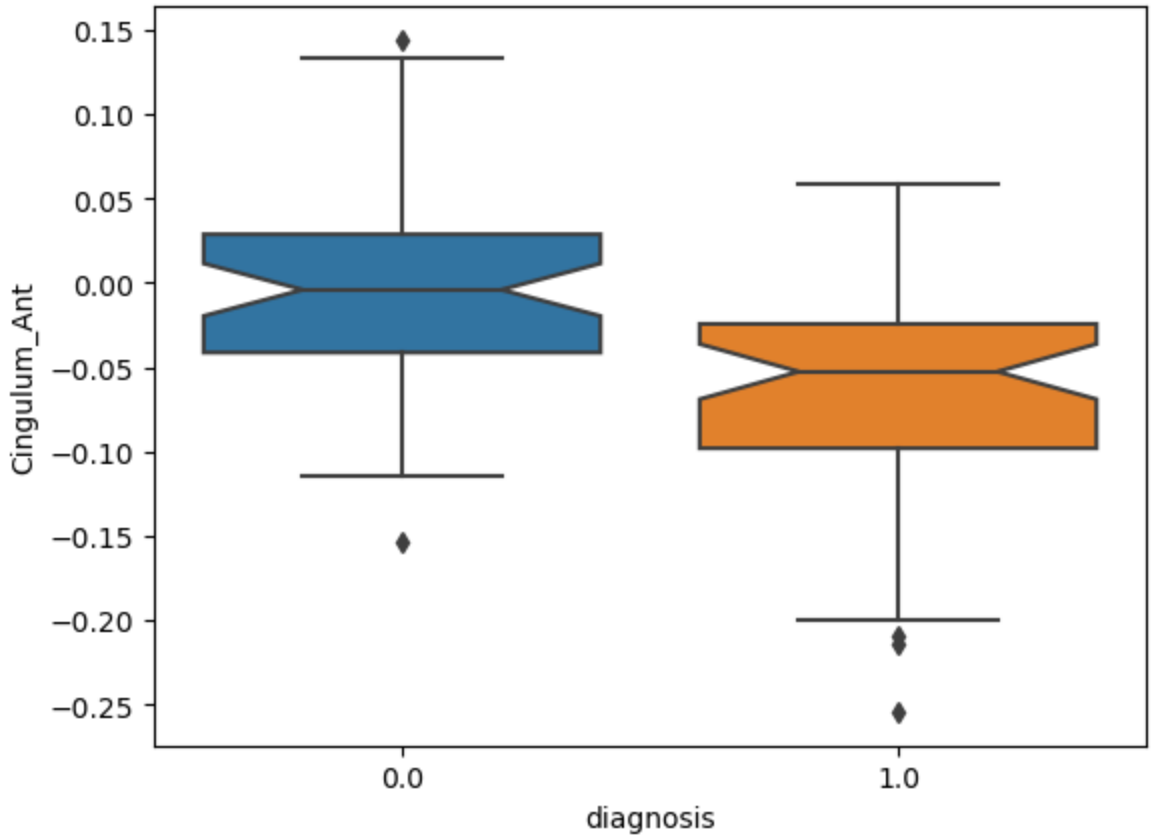
We can see that the following regions are significantly affected by the disease: Cingulum_Post, Cingulum_Ant, Occipital_Sup

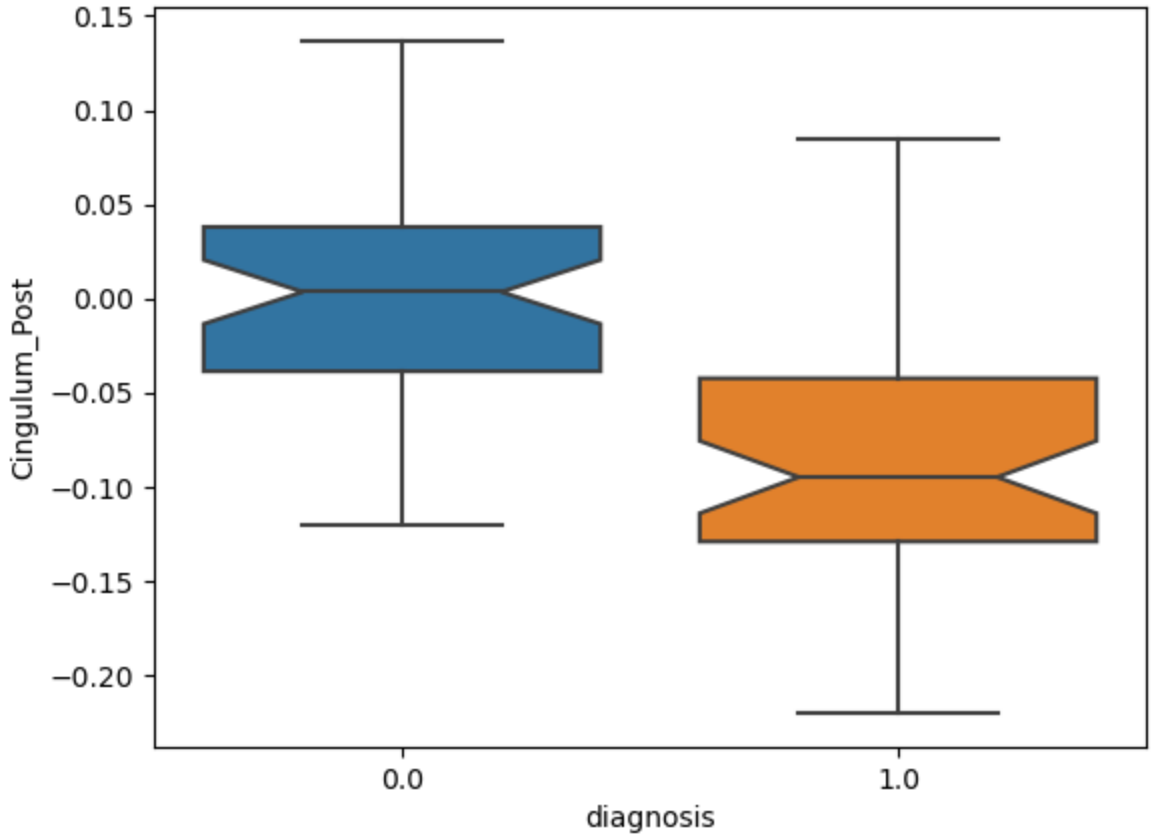
We can visualize the differences.

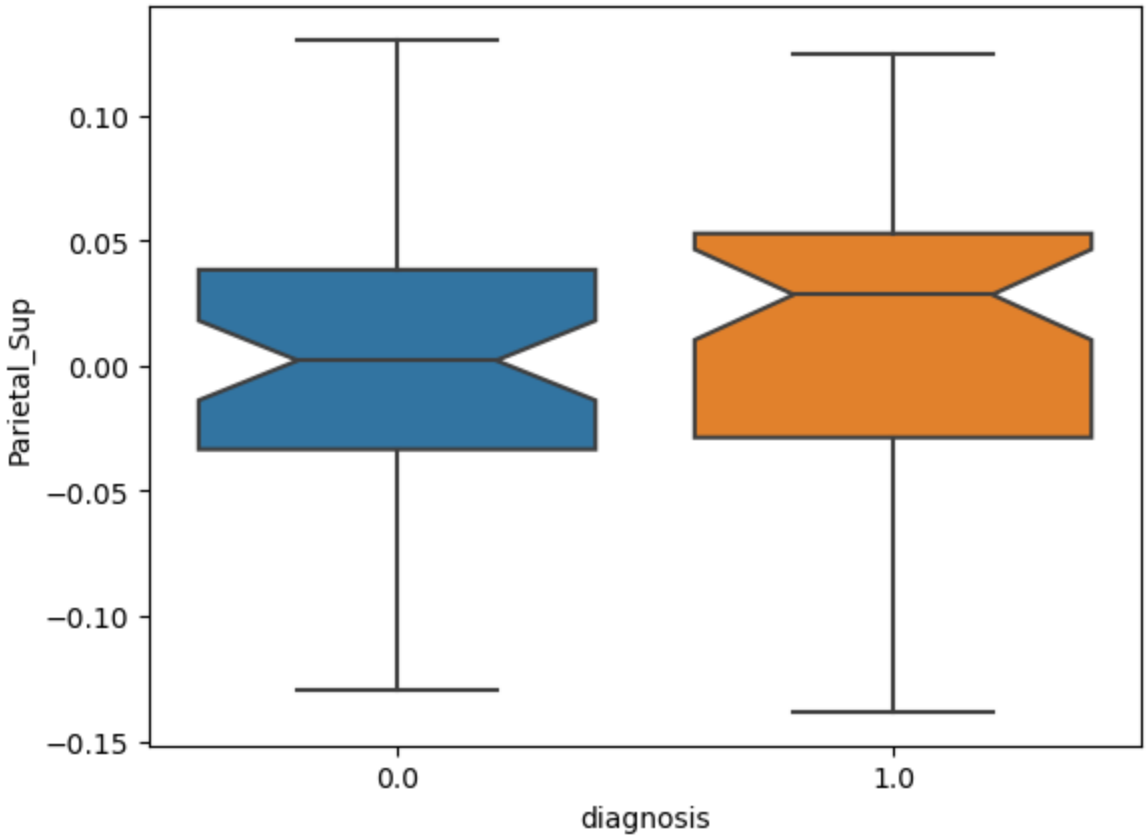
In [33]:
```
for i in data.drop(columns=['age', 'sex', 'diagnosis']).columns:
    y, X = dmatrices(
        # interaction terms
        f'{i} ~ age + sex + diagnosis + sex:diagnosis',
        data=data,
        return_type='dataframe'
    )
    # remove other effects
    y_hat = y[i] - np.dot(X[X.columns[X.columns != 'diagnosis']], B.loc[B.index != 'diag
    df = pd.concat([X['diagnosis'], y_hat], axis=1)
    # visualize
    sns.boxplot(data=df, x='diagnosis', y=i, notch=True)
    plt.show()
```

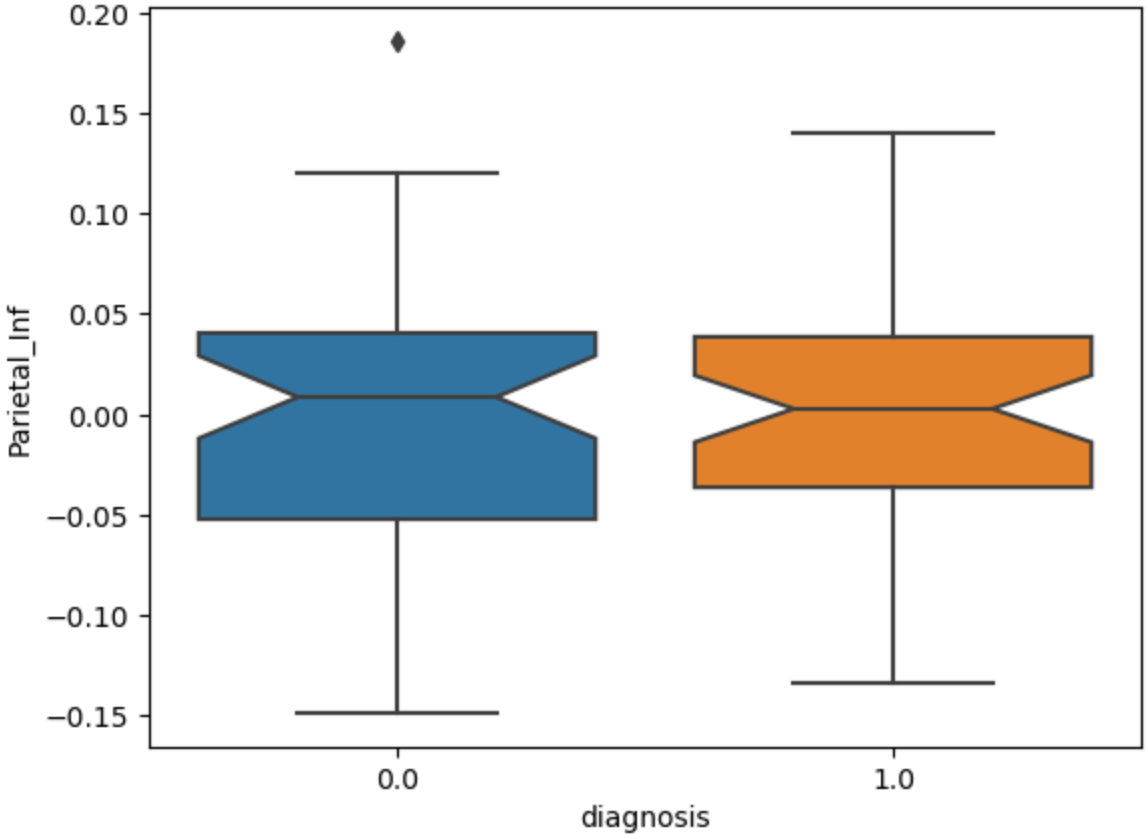

0.2
0.1
0.0
−0.1
−0.2
Frontal_Sup
0.0
1.0
diagnosis

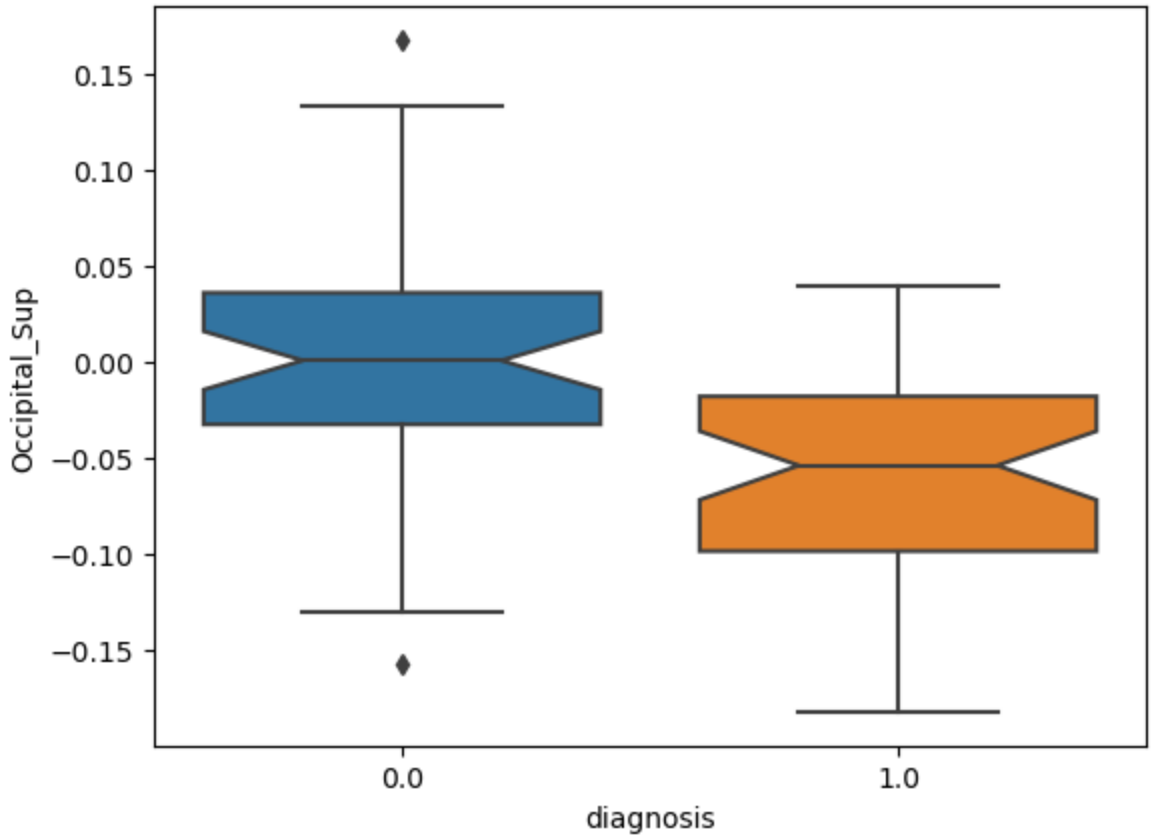

0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Frontal_Inf
0.0
1.0
diagnosis

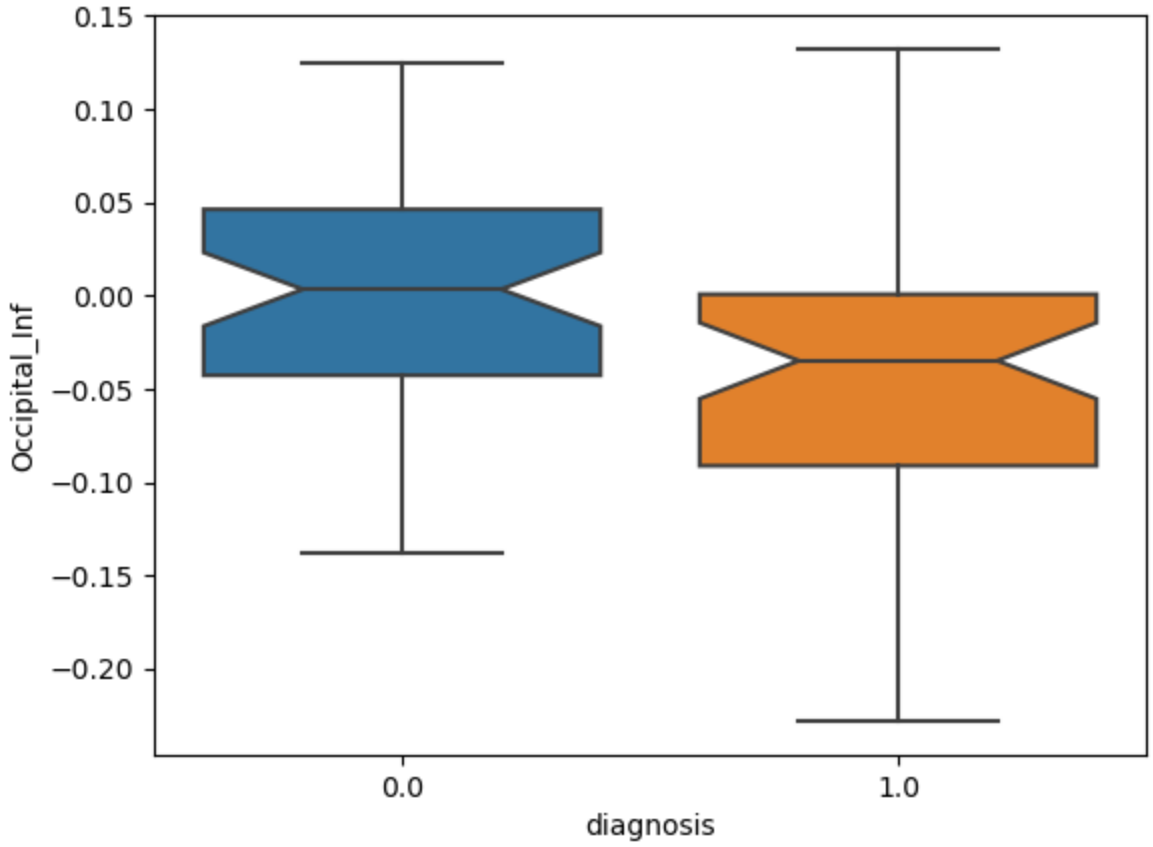

0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
Cingulum_Ant
0.0
1.0
diagnosis

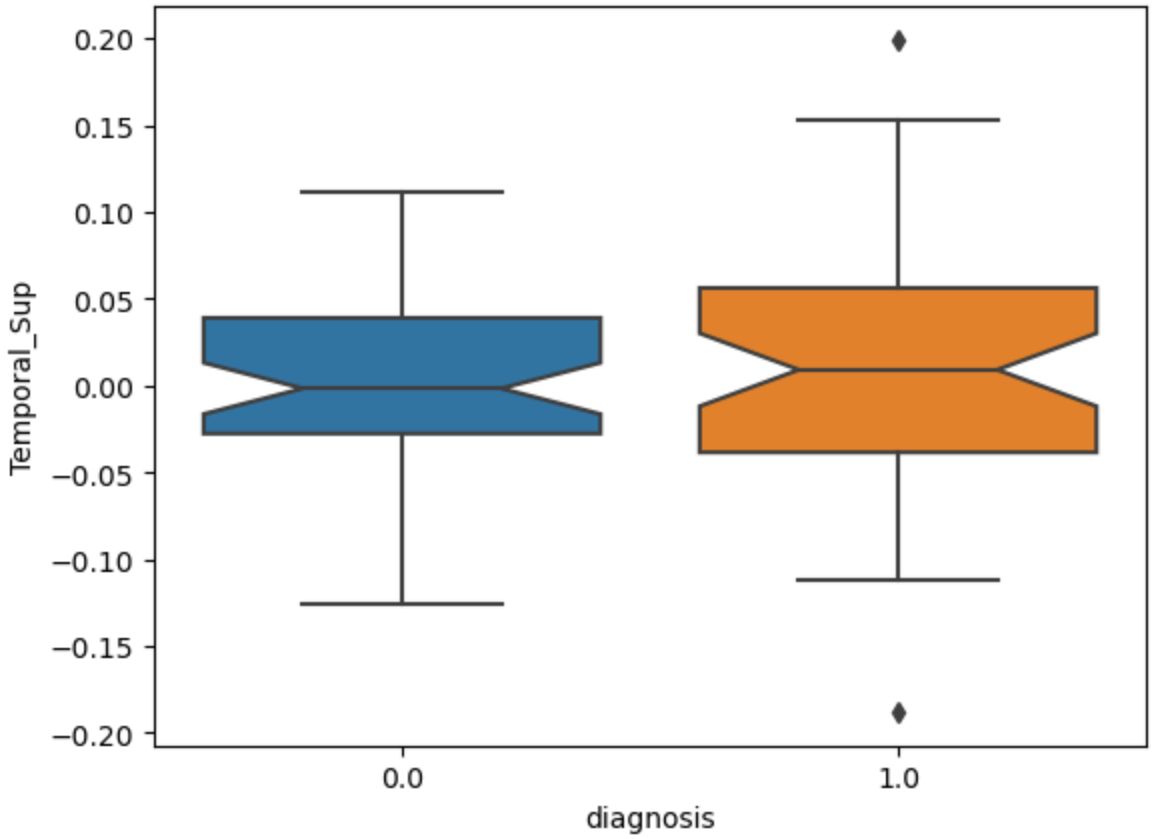

0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
Cingulum_Post
0.0
1.0
diagnosis

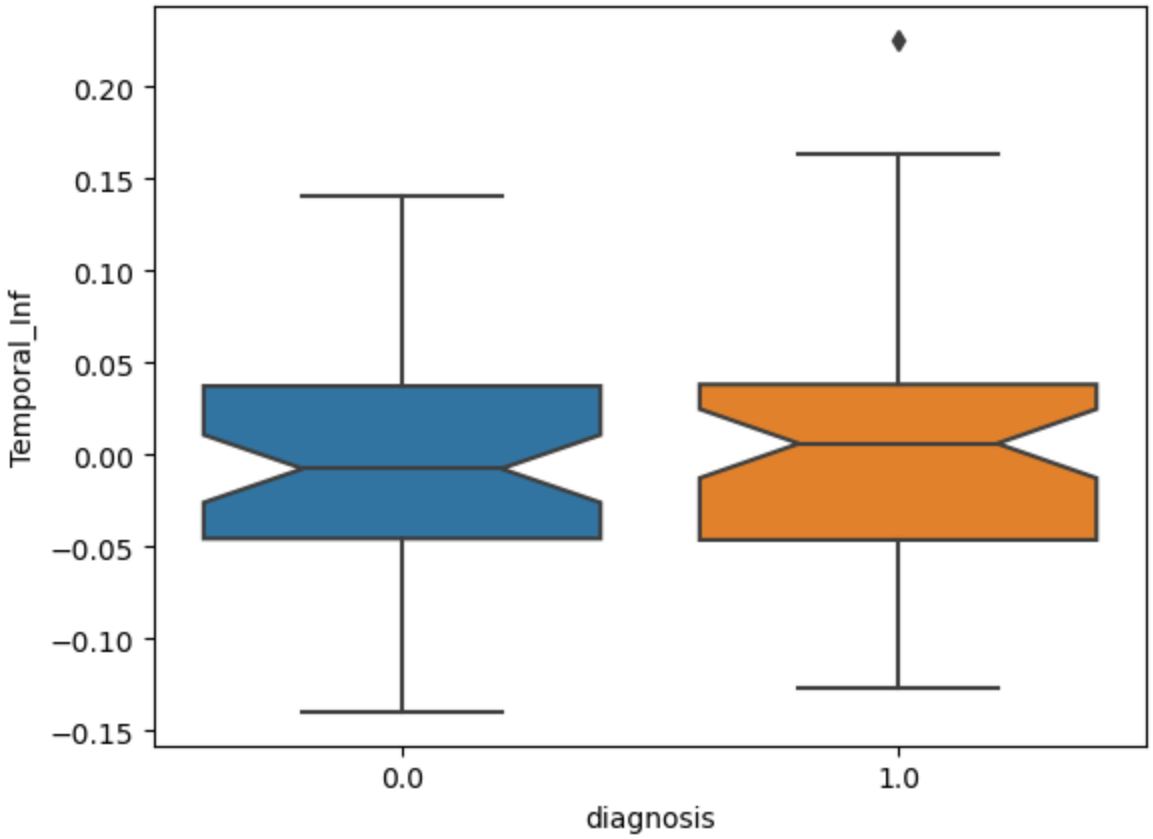

0.10
0.05
0.00
−0.05
−0.10
−0.15
Parietal_Sup
0.0
1.0
diagnosis


0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Parietal_Inf
0.0
1.0
diagnosis


0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Occipital_Sup
0.0
1.0
diagnosis


0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
Occipital_Inf
0.0
1.0
diagnosis





We can also explore the interaction effect between sex and diagnosis.

In [34]:
```
df_bh = pvals.loc['sex:diagnosis'].rename('pvals').to_frame()
df_bh
```

Out[34]:

	pvals
Frontal_Sup	0.160576
Frontal_Inf	0.036845

Cingulum_Ant	0.792013
Cingulum_Post	0.000778
Parietal_Sup	0.930997
Parietal_Inf	0.234558
Occipital_Sup	0.061775
Occipital_Inf	0.078049
Temporal_Sup	0.956729
Temporal_Inf	0.803743

In [35]:
```
# sort by pval
df_bh = df_bh.sort_values('pvals')
# rank
df_bh['rank'] = df_bh.rank()
# get BH critical value
df_bh['crit'] = df_bh.apply(lambda x: (x['rank']/len(df_bh))*FDR, axis=1)

df_bh
```

Out[35]:

	pvals	rank	crit
Cingulum_Post	0.000778	1.0	0.005
Frontal_Inf	0.036845	2.0	0.010
Occipital_Sup	0.061775	3.0	0.015
Occipital_Inf	0.078049	4.0	0.020
Frontal_Sup	0.160576	5.0	0.025
Parietal_Inf	0.234558	6.0	0.030
Cingulum_Ant	0.792013	7.0	0.035
Temporal_Inf	0.803743	8.0	0.040
Parietal_Sup	0.930997	9.0	0.045
Temporal_Sup	0.956729	10.0	0.050

In [36]:
```
# get highest rank where pval < crit
rank_max = df_bh[df_bh['pvals'] < df_bh['crit']]['rank'].max()
# all significant pvals
df_bh[df_bh['rank'] <= rank_max]
```

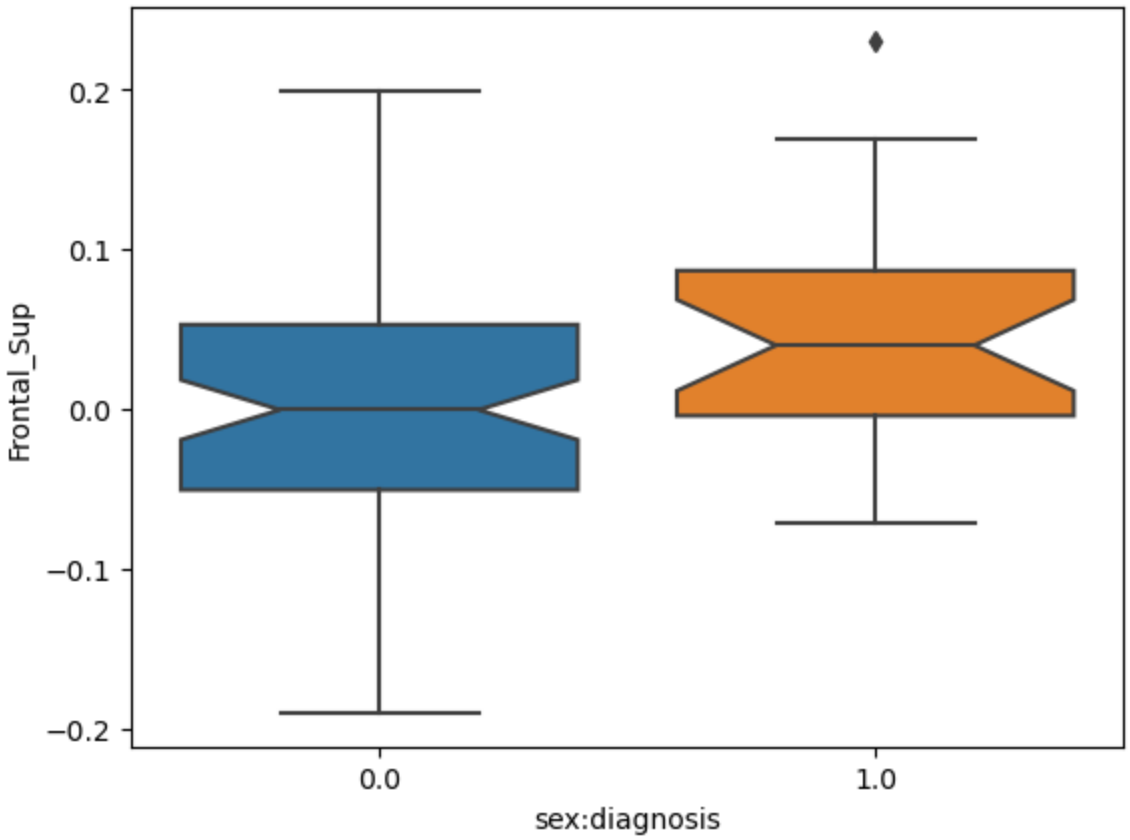
Out[36]:

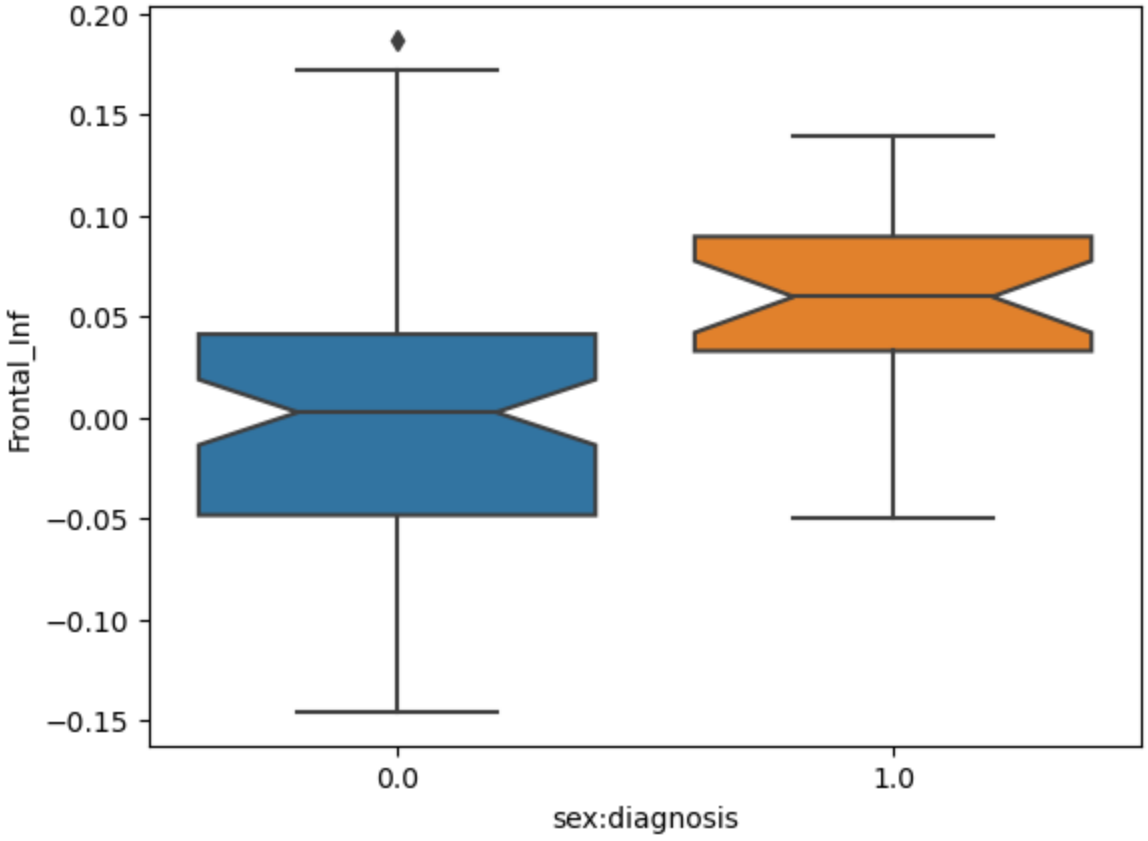
	pvals	rank	crit
Cingulum_Post	0.000778	1.0	0.005

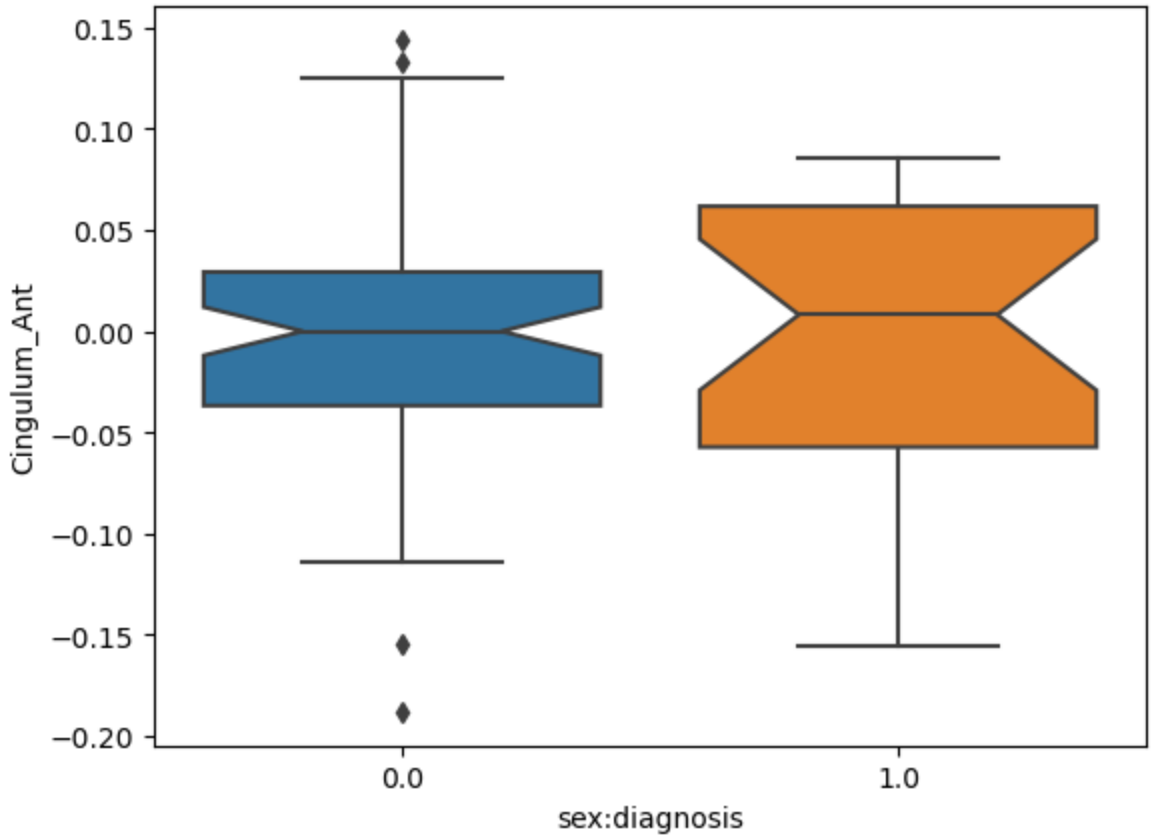
We can see that the following regions are significantly affected by the interaction between sex and diagnosis: Cingulum_Post

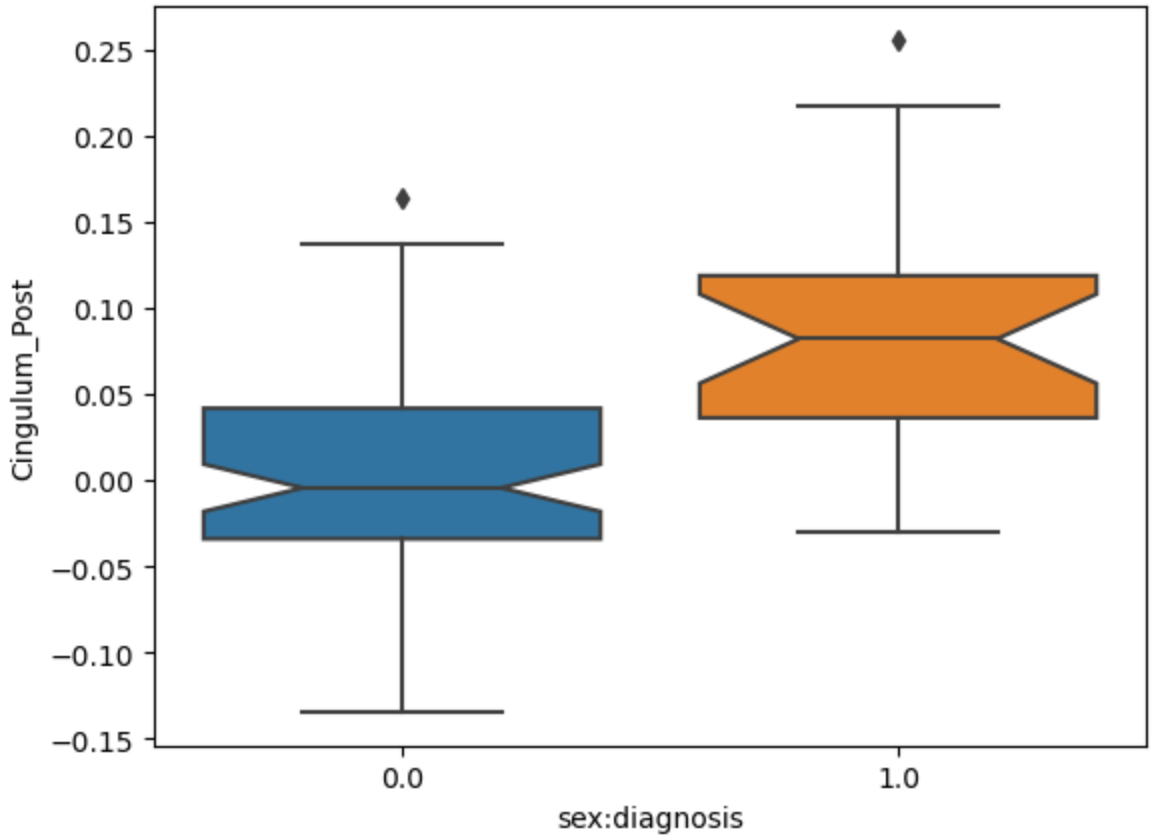
In [44]:
```
for i in data.drop(columns=['age', 'sex', 'diagnosis']).columns:
    y, X = dmatrices(
        # interaction terms
        f'{i} ~ age + sex + diagnosis + sex:diagnosis',
        data=data,
        return_type='dataframe'
```

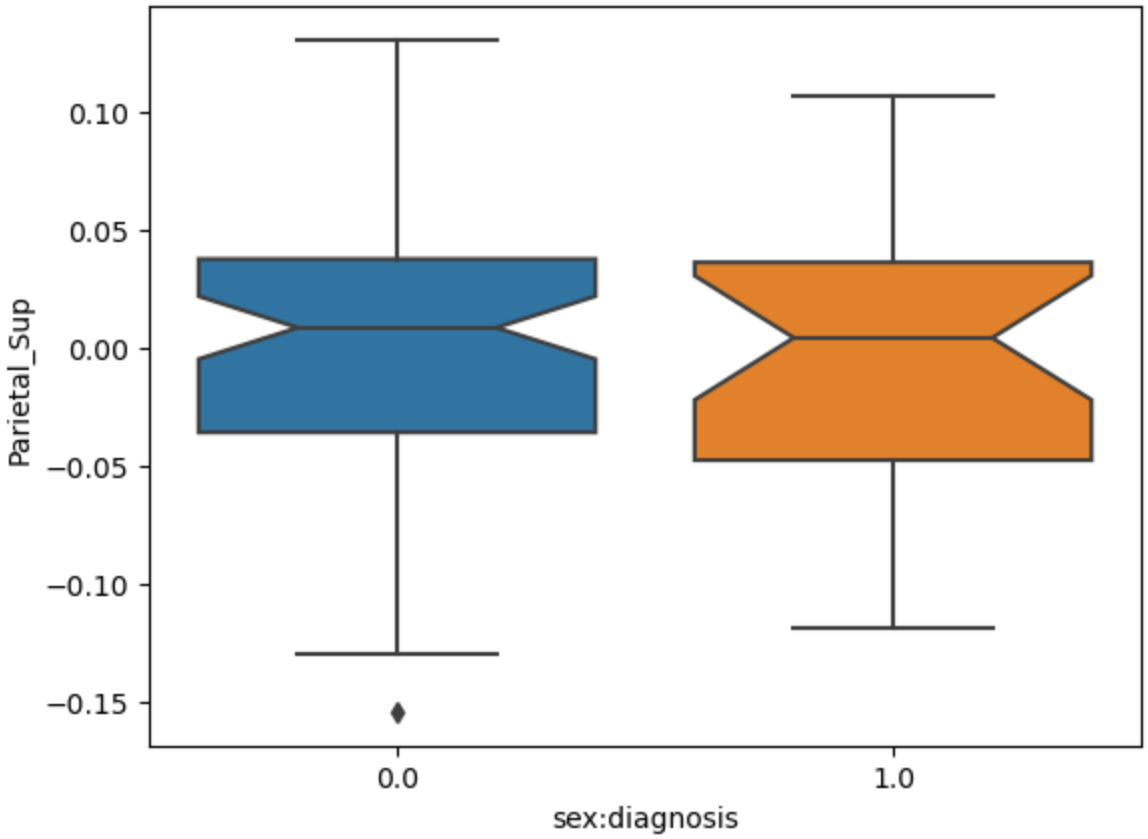
```
    )
    # remove other effects
    y_hat = y[i] - np.dot(X[X.columns[X.columns != 'sex:diagnosis']], B.loc[B.index != '
    df = pd.concat([X['sex:diagnosis'], y_hat], axis=1)
    # visualize
    sns.boxplot(data=df, x='sex:diagnosis', y=i, notch=True)
    plt.show()
```

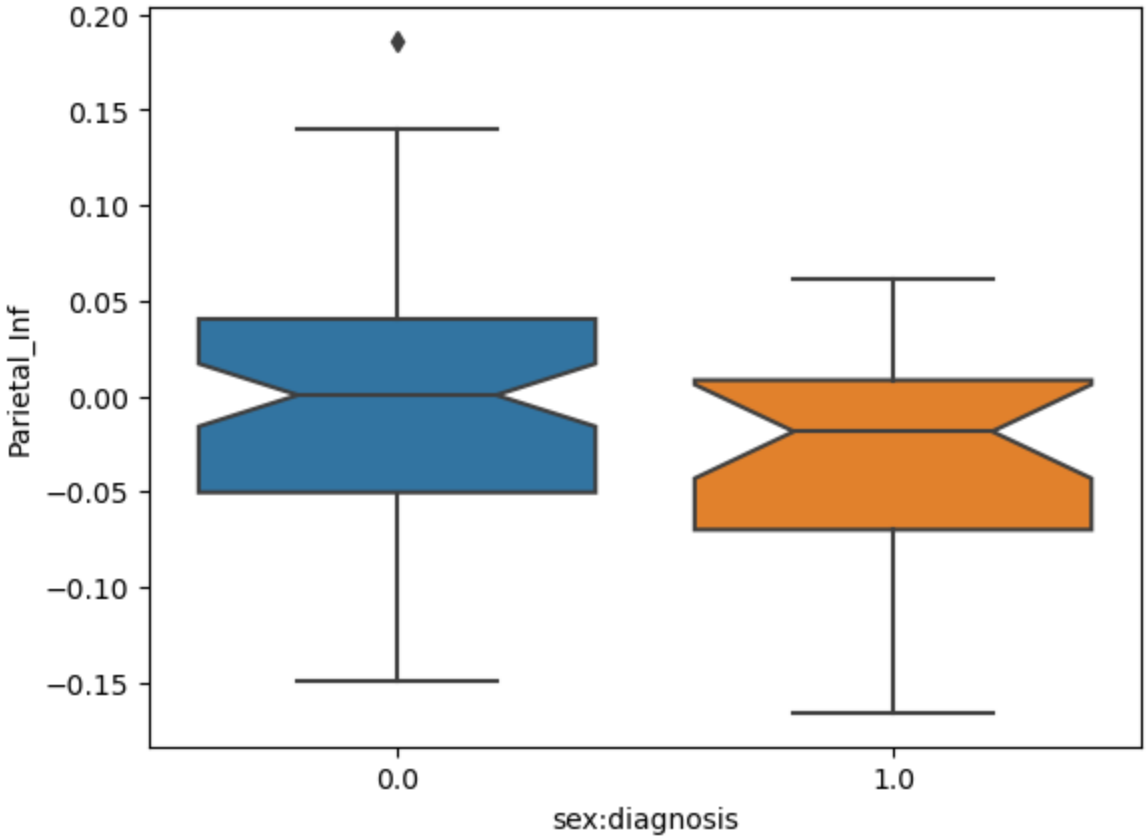


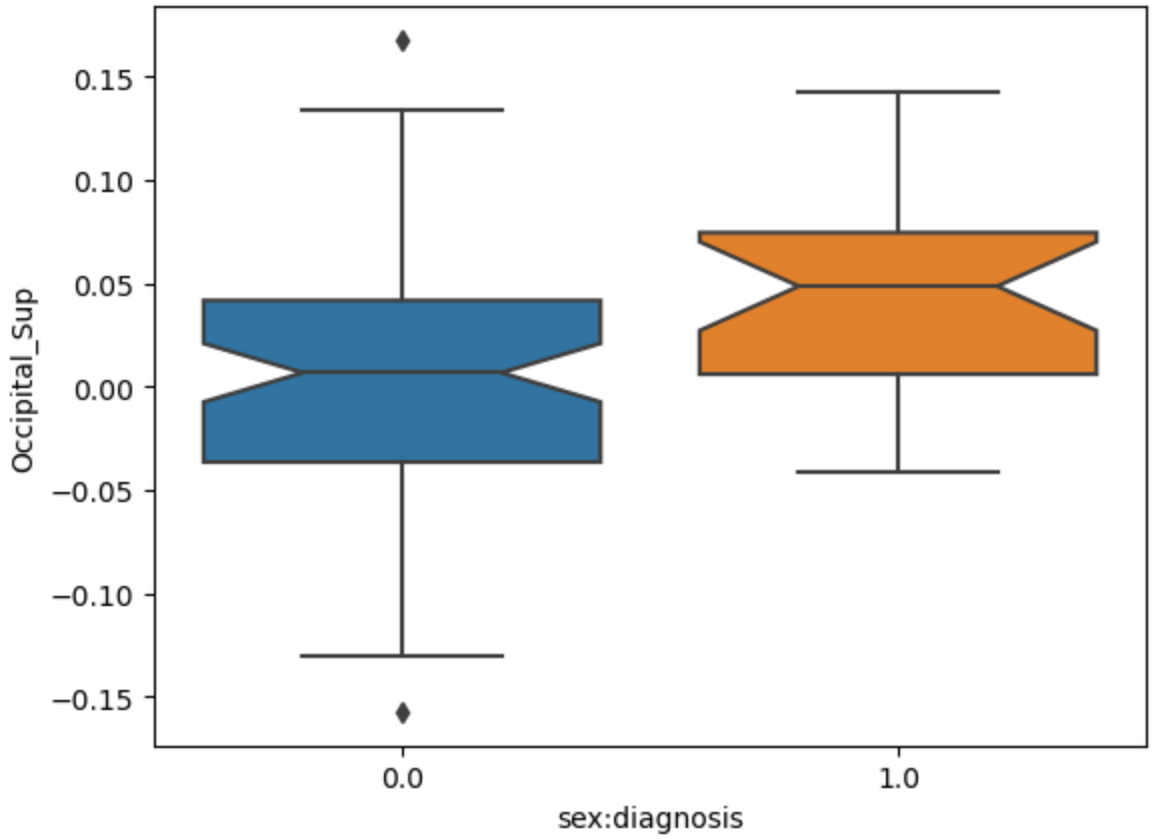


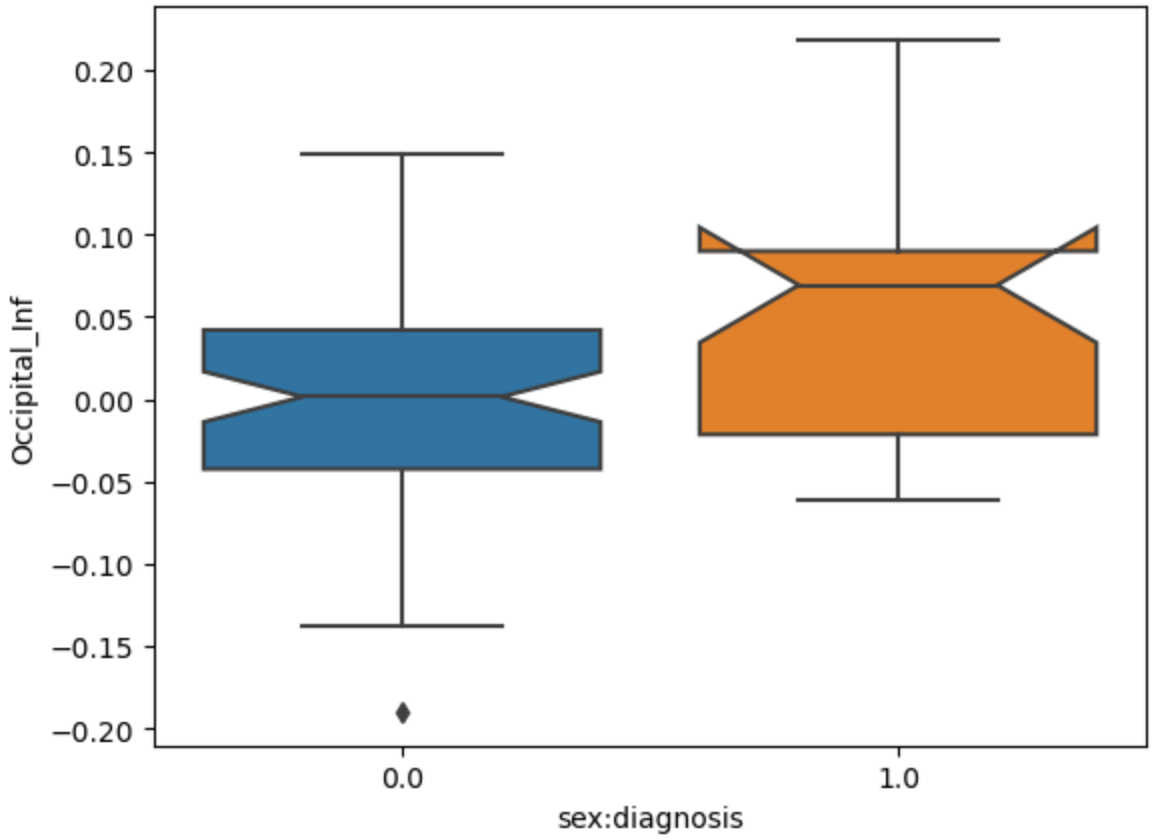

0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
Cingulum_Ant
0.0
1.0
sex:diagnosis

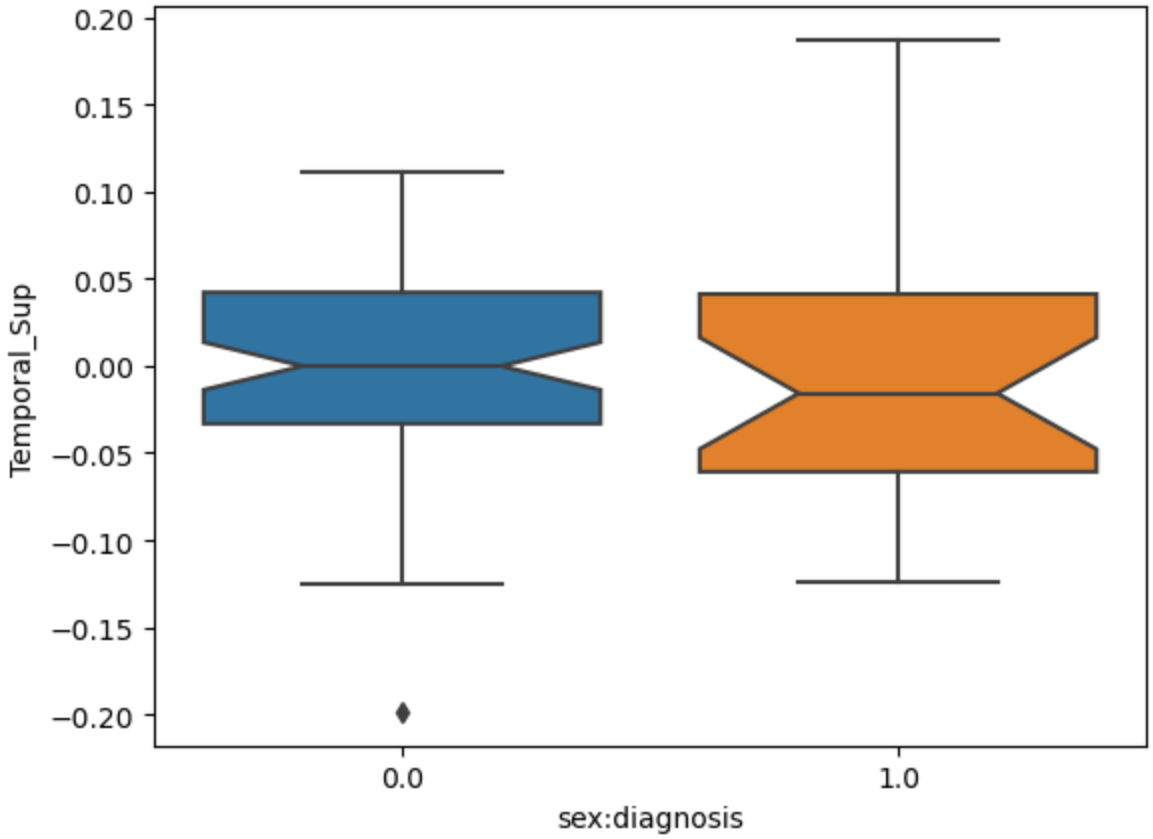

0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Cingulum_Post
0.0
1.0
sex:diagnosis

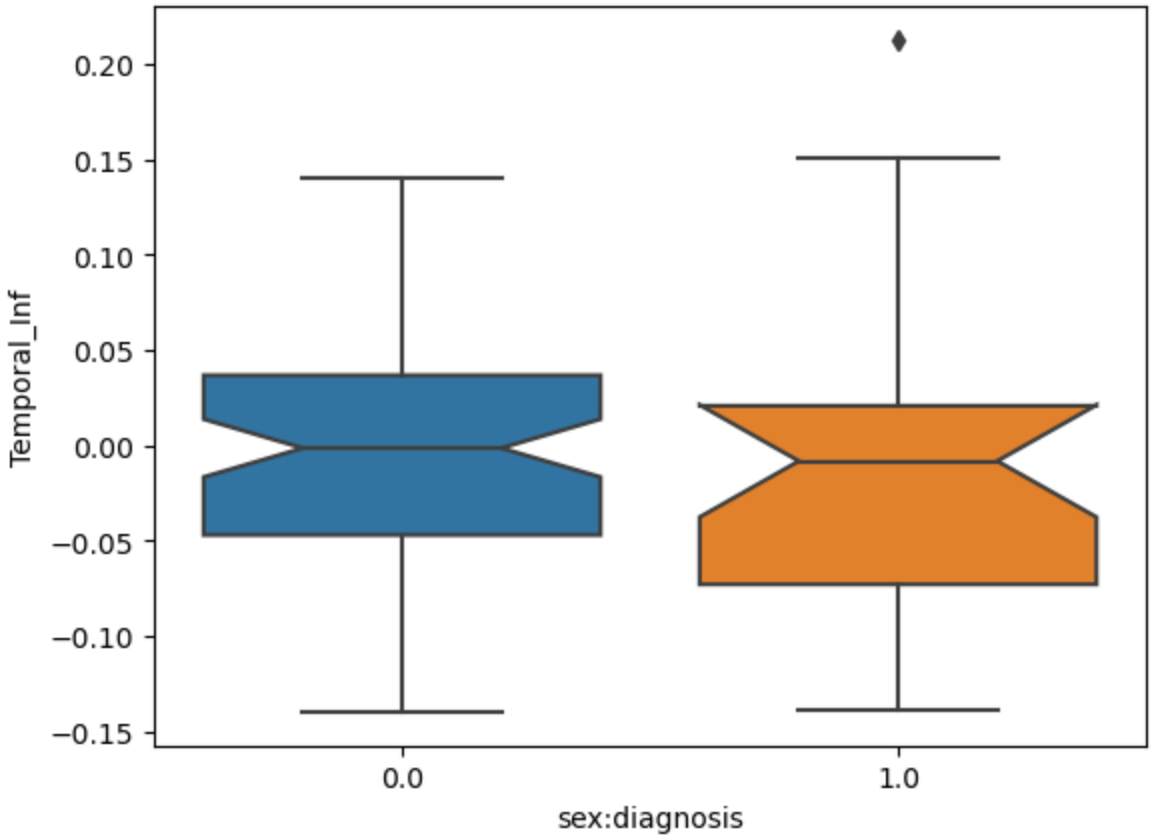

0.10
0.05
0.00
−0.05
−0.10
−0.15
Parietal_Sup
0.0
1.0
sex:diagnosis


0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Parietal_Inf
0.0
1.0
sex:diagnosis


0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Occipital_Sup
0.0
1.0
sex:diagnosis


0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
Occipital_Inf
0.0
1.0
sex:diagnosis





IGNORE BELOW

In [ ]:

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

Y = data.drop(columns=['age', 'sex', 'diagnosis'])
```

```
X = data.loc[:, ['age', 'sex', 'diagnosis']]

reg = LinearRegression()
```

In [ ]:
```
display(X)
display(Y)
```

It is expected that there are following interactions: age and sex; age and diagnosis; sex and diagnosis. Interaction terms are thus added.

In [ ]:
```
xs = [
    'intercept', 'age', 'sex', 'diagnosis', 'age_sex', 'age_diagnosis', 'sex_diagnosis'
]

poly = PolynomialFeatures(2, interaction_only=True, include_bias=True)
X = pd.DataFrame(poly.fit_transform(X), columns=xs)
```

In [ ]:
```
display(X)
display(Y)
```

In [ ]:
```
# # do for each brain region
# for i in Y.columns:
#     print('Region: ', i)
#     # ground truth of brain region
#     y = Y[i]
#     # fit
#     reg.fit(X,y)
#     print('Score: ', reg.score(X,y))
#     # get weights
#     b = pd.Series(reg.coef_, index=xs)
#     print('Weights: ')
#     display(b)
#     # get y hat to remove effect of age and sex
#     y_hat = y - np.dot(X[X.columns[X.columns != 'diagnosis']], b[b.index != 'diagnosis
#     df = pd.concat([X['diagnosis'], y_hat], axis=1)
#     display(df)
#     sns.boxplot(data=df, x='diagnosis', y=i, notch=True)
#     plt.show()
```

We can fit a linear model and remove confounding effects to see whether diagnosis has an effect on biomarkers.

In [ ]:
```
# fit linear model
reg.fit(X,Y)
```

In [ ]:
```
reg.score(X, Y)
```

In [ ]:
```
# weights
B = pd.DataFrame(reg.coef_.T, index=xs, columns=Y.columns)
B
```

In [ ]:
```
# remove confounding effects
Y_hat = Y - np.dot(X[X.columns[X.columns != 'diagnosis']], B[B.index != 'diagnosis'])
Y_hat
```

In [ ]:
```
df = pd.concat([X['diagnosis'], Y_hat], axis=1)
df
```

We can visualize the effect of the diagnosis.

In [ ]:
```
for i in df.drop(columns=['diagnosis']).columns:
    # plot boxplot
    sns.boxplot(data=df, x='diagnosis', y=i, notch=True)
    plt.show()
```

We can use a two-tailed two-sample t-test to see whether the difference in biomarker is significant between the diseased group and healthy controls.

In [ ]:
```
biomarkers_control = df.loc[df['diagnosis'] == 0, df.columns[df.columns != 'diagnosis']]
biomarkers_disease = df.loc[df['diagnosis'] == 1, df.columns[df.columns != 'diagnosis']]
display(biomarkers_control)
display(biomarkers_disease)
```

In [ ]:
```
result = stats.ttest_ind(
    biomarkers_control,
    biomarkers_disease,
    axis=0,
    alternative='two-sided'
)
result
```

Before making conclusions for significance, we need to correct for multiple testing using the Benjamini-Hochberg procedure. Here, we set the false discovery rate to 0.05.

In [ ]:
```
FDR = 0.05
```

In [ ]:
```
# get df of pvals
df_bh = pd.Series(result.pvalue, index=biomarkers_control.columns, name='pvals').to_fram
# sort by pval
df_bh = df_bh.sort_values('pvals')
# rank
df_bh['rank'] = df_bh.rank()
# get BH critical value
df_bh['crit'] = df_bh.apply(lambda x: (x['rank']/len(df_bh))*FDR, axis=1)

df_bh
```

In [ ]:
```
# get highest rank where pval < crit
rank_max = df_bh[df_bh['pvals'] < df_bh['crit']]['rank'].max()
# all significant pvals
df_bh[df_bh['rank'] <= rank_max]
```

We can see that the following regions are significantly affected by the disease: Temporal_Inf, Cingulum_Post, Occipital_Inf, Frontal_Inf, Temporal_Sup, Cingulum_Ant, Parietal_Sup, Frontal_Sup

We can also explore the interaction effect between sex and diagnosis.

In [ ]:
```
# remove other effects
Y_hat = Y - np.dot(X[X.columns[X.columns != 'sex_diagnosis']], B[B.index != 'sex_diagnos
Y_hat
```

In [ ]:
```
df = pd.concat([X['sex_diagnosis'], Y_hat], axis=1)
df
```

In [ ]:
```
for i in df.drop(columns=['sex_diagnosis']).columns:
    # plot boxplot
    sns.boxplot(data=df, x='sex_diagnosis', y=i, notch=True)
    plt.show()
```

In [ ]:
```
biomarkers_control = df.loc[df['sex_diagnosis'] == 0, df.columns[df.columns != 'sex_diag
biomarkers_disease = df.loc[df['sex_diagnosis'] == 1, df.columns[df.columns != 'sex_diag
display(biomarkers_control)
display(biomarkers_disease)
```

In [ ]:
```
result = stats.ttest_ind(
    biomarkers_control,
    biomarkers_disease,
    axis=0,
    alternative='two-sided'
)
result
```

In [ ]:
```
# get df of pvals
df_bh = pd.Series(result.pvalue, index=biomarkers_control.columns, name='pvals').to_fram
# sort by pval
df_bh = df_bh.sort_values('pvals')
# rank
df_bh['rank'] = df_bh.rank()
# get BH critical value
df_bh['crit'] = df_bh.apply(lambda x: (x['rank']/len(df_bh))*FDR, axis=1)

df_bh
```

In [ ]:
```
# get highest rank where pval < crit
rank_max = df_bh[df_bh['pvals'] < df_bh['crit']]['rank'].max()
# all significant pvals
df_bh[df_bh['rank'] <= rank_max]
```

The following regions have significant interaction effect between sex and diagnosis: Cingulum_Post, Frontal_Inf, Occipital_Sup, Occipital_Inf, Frontal_Sup

In [ ]:
```
df = pd.concat([X, Y], axis=1)
df
```

In [ ]:
```
for i in df.drop(columns=xs).columns:
    # plot boxplot
    sns.boxplot(data=df, x='sex', y=i, hue='diagnosis', notch=True)
    plt.show()
```

In [ ]:
```

```