دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده‌ی مهندسی برق و کامپیوتر

تمرین اول درس مبانی علوم شناختی

نام و نام خانوادگی:

سینا پیرمرادیان

شماره دانشجویی:

۸۱۰۱۰۱۱۲۵

استاد:

دکتر محمدرضا ابوالقاسمی دهاقانی

TABEL OF FIGURE

Introduction:

The first part of the exercise is related to the analysis of different Sigmoid and Gussian Sigmoid methods for fitting data. This part is given in the first code. The second part of the exercise contains 5 questions that we must answer using the dataset provided to us and also using statistical tests. These questions were answered in the second to sixth codes. In the third part of the exercise, it is said to analyze these questions using the ROC chart. The code for this part is given at the end of the second to sixth parts. In the fourth part of the exercise, it is said to make a new hypothesis and test it. The hypothesis of this part is implemented in the part 6 report. The fifth part of the exercise is about the same steps and the combined step, which we must compare and conclude, this part is implemented in the seventh code.
Regarding the last part, which is about the cofounding factor, it can be said that things like the level of fatigue or freshness of the tester can be mentioned.

1. Psychometric Fitting

In this section, we intend to fit a sigmoid curve with the least squares criterion for Hassan and Goli in each Morph Level by using the two types of Sigmoid function mentioned in the title of the exercise. Below are the two types of sigmoid function used.

Type 1:

$$\sigma(x) = \frac{\alpha}{1 + \exp(-\beta x)}$$

Type 2:

$$\sigma(x) = \frac{\alpha}{1 + \exp\big(-\beta(x - \gamma)\big)} + \gamma$$

At the begining, we count the number of Goli and Hassan in each Morph level using Python and display it in a Scatter diagram for each, which is shown in Figure 1.

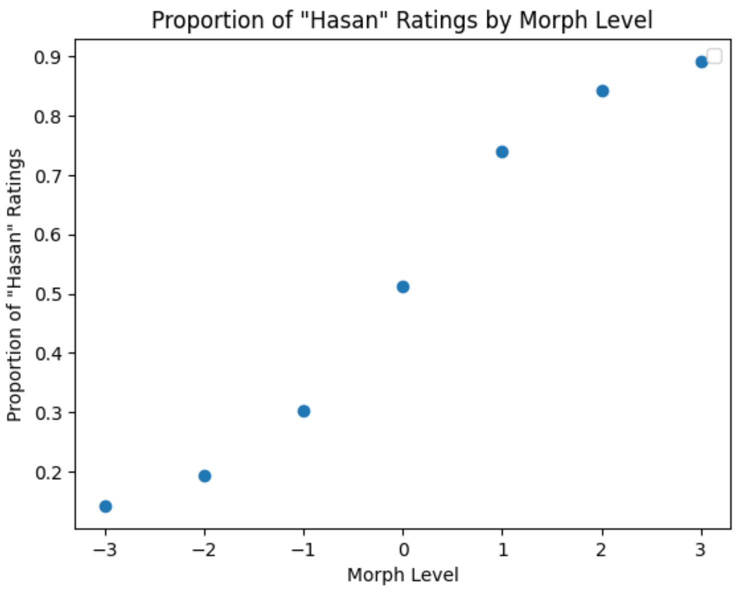

Figure 1

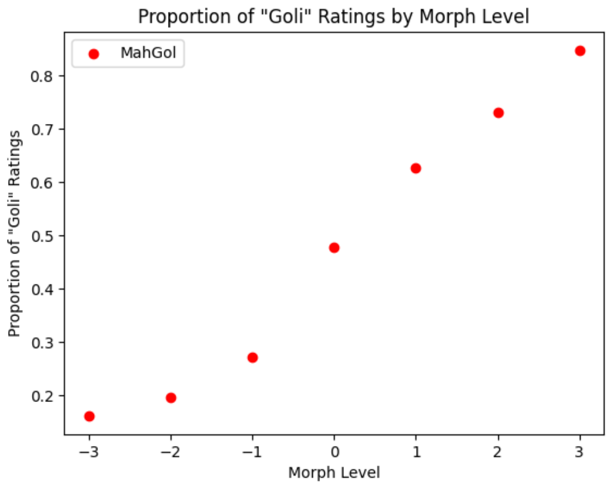

Figure 2

Now, as mentioned, we are going to fit a sigmoid function of the first type separately for Hassan and Goli. and display it on the diagram of the previous part, which was done for Hassan in Figure 2, for Goli in Figure 3. Also, the value of their β parameter is shown in the figure below. Also, the code of this part in Python is shown in Figure 4.

Now it's time to fit the two functions mentioned in the project form on the above examples and compare them with the AIC and BIC criteria. The higher the number of these two criteria means more points for the model and it is better. That is, the higher the penalty of the model, the lower the number of these two criteria. The Sigmoid diagram is given below.

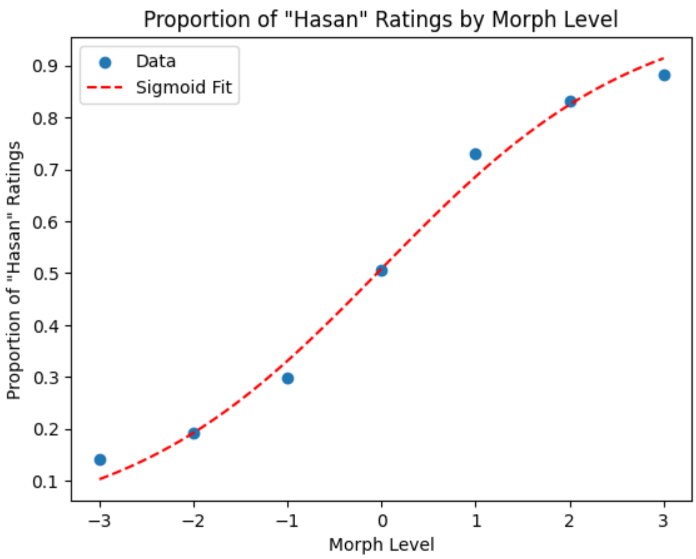

Figure 3

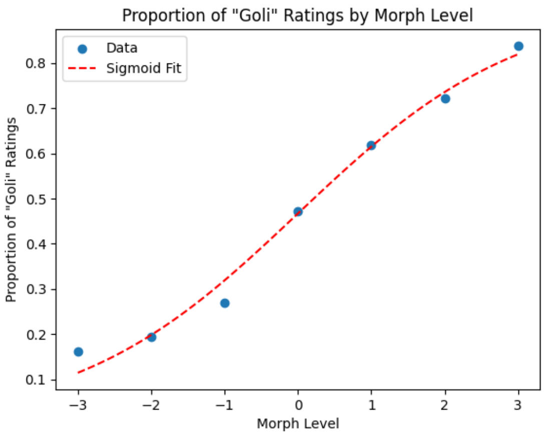
Proportion of "Goli" Ratings by Morph Level
Data
Sigmoid Fit
Proportion of "Goli" Ratings
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
−3
−2
−1
0
1
2
3
Morph Level

Figure 4

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# load the data from the Google Drive link
url = 'https://drive.google.com/u/0/uc?id=1G_j5BtejbtBR18sKt0-epRsXkedghqJA&export=download'
data = pd.read_csv(url)

# define the sigmoid function
def sigmoid(x, beta, alpha):
    y = alpha / (1 + np.exp(-beta*(x)))
    return y

# create a list of trialKeys and levelFace pairs
pairs = [('AbHa', -3), ('AbHa', -2), ('AbHa', -1), ('AbHa', 0), ('AbHa', 1), ('AbHa', 2), ('AbHa', 3)]

# create lists to store the x and y values for each pair
x_values = []
y_values = []

# loop through each pair and calculate the proportion of "Hasan" ratings
for pair in pairs:
    # filter the data by trialKeys and levelFace
    filtered_data = data[(data['trialKeys'] == pair[0]) & (data['levelFace'] == pair[1])]

    # count the number of "Hasan" ratings
    hasan_count = filtered_data['srespChoice'].value_counts()['Hasan'] if 'Hasan' in filtered_data['srespChoice'].value_counts() else 0
    total_count = len(filtered_data)

    # calculate the proportion of "Hasan" ratings
    hasan_proportion = hasan_count / total_count

    # add the x and y values to the lists
    x_values.append(pair[1])
    y_values.append(hasan_proportion)

# fit the sigmoid function to the data
popt, pcov = curve_fit(sigmoid, x_values, y_values)

# create a range of x values for the curve
x_curve = np.linspace(min(x_values), max(x_values), 100)

# calculate the y values for the curve using the fitted parameters
y_curve = sigmoid(x_curve, *popt)

# plot the data and the fitted curve
plt.scatter(x_values, y_values, label='Data')
plt.plot(x_curve, y_curve, 'r--', label='Sigmoid Fit')
plt.title('Proportion of "Hasan" Ratings by Morph Level')
plt.xlabel('Morph Level')
plt.ylabel('Proportion of "Hasan" Ratings')
plt.legend()
plt.show()

# print the value of the sensitivity parameter
print('Sensitivity Parameter (beta):', popt[0])
```

Figure 5

With respect to this part, we are going to do the same work as the previous part, this time for the type 2 sigmoid function. This work and the value of the parameters of this function for Hassan and Goli are shown in Figure 5 and Figure 6, respectively. Also, the code for this part is in Figure 7 has come.

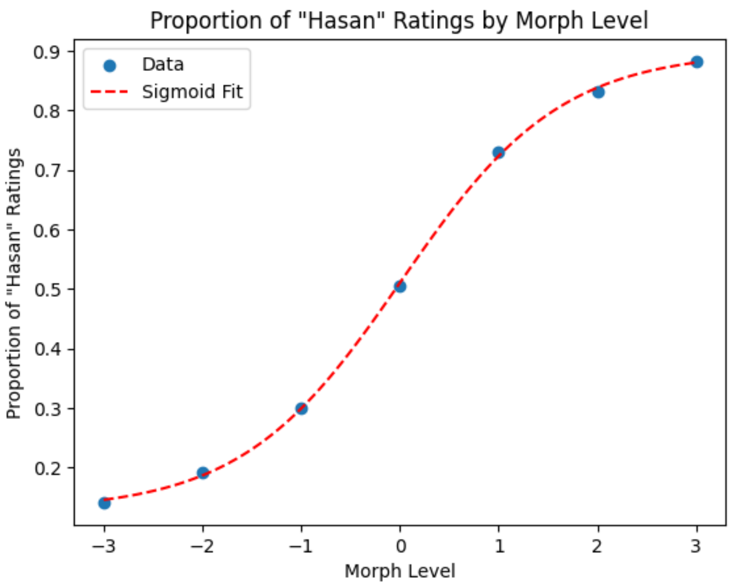

Figure 6

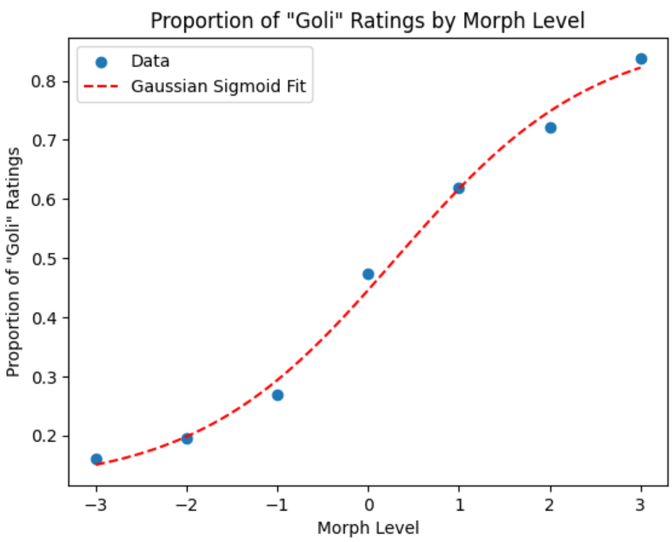

Figure 7

As it was said to obtain the Sigmoid function in the previous parts, the least squares criterion was used. Figure 9 is shown.

```python
# Fit the sigmoid model and calculate the AIC and BIC scores
# x = data["morphing"]
# y = data["response"]
popt_sig, _ = curve_fit(sigmoid, x, y, p0=[0.5, 0], method='lm')
nll_sig = neg_log_likelihood(popt_sig, x, y, sigmoid)
k_sig = 3
n = len(x)
aic_sig = 2*k_sig - 2*nll_sig
bic_sig = k_sig*np.log(n) - 2*nll_sig

# Fit the Gaussian CDF model and calculate the AIC and BIC scores
popt_gauss, _ = curve_fit(gauss_sigmoid, x, y)
nll_gauss = neg_log_likelihood(popt_gauss, x, y, gauss_sigmoid)
k_gauss = 3
aic_gauss = 2*k_gauss - 2*nll_gauss
bic_gauss = k_gauss*np.log(n) - 2*nll_gauss

# Print the results
print("Sigmoid model:")
print(f"Parameters: {popt_sig}")
print(f"AIC: {aic_sig*100:.2f}")
print(f"BIC: {bic_sig*100:.2f}")
print("")
print("Gaussian Sigmoid model:")
print(f"Parameters: {popt_gauss}")
print(f"AIC: {aic_gauss*100:.2f}")
print(f"BIC: {bic_gauss*100:.2f}")
```

Figure 8

```
Sigmoid model:
Parameters: [1.01666258 0.72811132]
AIC: -123.65
BIC: -139.88

Gaussian Sigmoid model:
Parameters: [0.77163051 1.23581573 0.01305661 0.12741509]
AIC: -119.71
BIC: -135.93
```

Figure 9

The AIC score calculated for this model is -123.85 for the graph on the right, which is related to Goli/Mahnaz, and -139.88, which is related to Hasan/Abbas, on the left graph. The BIC score for this graph on the right is -119.71 and for the graph on the left it is -135.95.

Now that we have drawn the graphs and calculated the scores using the AIC and BIC methods, we can see that both the Gussian graph fits the points better and the scores related to this graph are better than the sigmoid model. Therefore, it can be claimed that this model has worked better than the sigmoid model.

2. Tesing 5 Hypotheses

In this section, a number of propositions have been proposed, which we are trying to find their answers using the available data and prove them by using appropriate statistical analysis and also by fitting the sigmoid function and obtaining the β parameter.

First Question:

We want to check whether identification of identities are different in different frequency bands or not?

To answer this question, first sort the available data based on frequency bands (HF, LF, IF) and then get the number of Goli for each Morph level in each frequency range and on it a Sigmoid function of the type First, fit and obtain the average β parameters for the candidates in each frequency range and display it on a bar chart, the code of this part is shown in Figure 10 and its bar chart is shown in Figure 11.

hypotheses 1 ◀ frequency *level*

```
from sklearn.metrics import roc_auc_score

data = pd.read_csv('/content/data.csv')
freq_levels = ['IF', 'LF', 'HF']
data_freqLevel = {level: data[data.levelFreq==level] for level in freq_levels}
print(data_freqLevel)


# fit sigmoid on sorted data
betas = []

for level in freq_levels:
    x = data_freqLevel[level]['levelFace']
    y = (data_freqLevel[level]['srespChoice'] == 'Goli')
    popt, pcov = curve_fit(sigmoid, x, y)
    betas.append(popt[1])

# plot Beta coeff in bar plot

plt.bar(freq_levels, betas)
plt.xlabel('Frequency Level')
plt.ylabel('Sensitivity (beta)')
plt.title('Sensitivity to detecting identity from different spatial frequency bands')
plt.show()
```

Figure 10

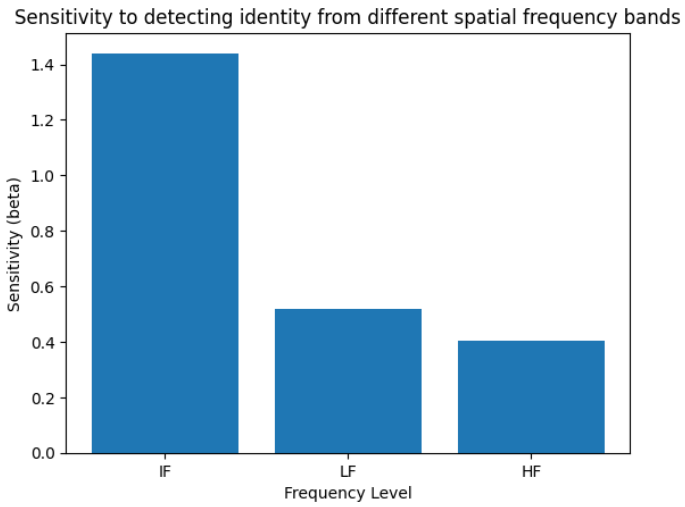

Figure 11

Also, for the statistical analysis of this part, we perform a statistical analysis using Z Test, the results of which are shown in Figure 12.

Z test analysis

```
[59]  x1, x2, x3 = Ztest(HF_s/HF_total, HF_total,
      print('p-value for  HF and LF: ' + str(x1))
      print('HF:  ' + str(x2))
      print('LF : ' + str(x3))

      p-value for  HF and LF: 2.0288047811034525e-10
      HF: 0.7579299812030075
      LF : 0.7868303571428571

[60]  x1, x2, x3 = Ztest(HF_s/HF_total, HF_total,
      print('p-value for  HF and LF: ' + str(x1))
      print('HF:  ' + str(x2))
      print('LF : ' + str(x3))

      p-value for  HF and LF: 2.1071044722879627e-228
      HF: 0.7579299812030075
      LF : 0.8909774436090225

[61]  x1, x2, x3 = Ztest(LF_s/LF_total, LF_total,
      print('p-value for  HF and LF: ' + str(x1))
      print('HF:  ' + str(x2))
      print('LF : ' + str(x3))

      p-value for  HF and LF: 1.3665226565283558e-150
      HF: 0.7868303571428571
      LF : 0.8909774436090225
```

Figure 12

So, in response to the question asked in the exercise, whether the sensitivity parameters are the same for different stages of the test or not, the results of the test were obtained such that for AbHa tests these parameters are almost the same and for MahGol tests these parameters are different from each other (at least A pair of parameters are different).

As you can see, if our coefficient is closer to zero, it is better, because in this case, the element of chance has no effect in our analysis, and the answer to the question is yes. Here, too, identity recognition has been done better in the IF frequency.

Also, the area under the ROC diagram in this case is shown in Figure 13.

```
beta
[1.4401525335783758]
AUC score
0.8690378289473684
beta
[1.4401525335783758, 0.5184904889641667]
AUC score
0.7458196271929827
beta
[1.4401525335783758, 0.5184904889641667, 0.4024688411001252]
AUC score
0.7121025219298246
```

Figure 13

Second question:

Are people more skilled in identifying images of their same and opposite gender?

In this part, first we merge subjectInfo and data, then we sort the obtained data frame based on gender, and then I have to see how each of the two genders performed in the detection of Hassan and Goli, and make a bar chart for each of them. We draw separately, the code of this part is shown in figure 14, and the actions of men and women for Goli are shown in figure 15 and for Hassan in figure 16.

```
data = pd.read_csv('/content/merged_data.csv', on_bad_lines='skip')

sex = ['M', 'F']
data_sex = {level: data[data.sex==level] for level in sex}
# print(data_sex)

# fit sigmoid on sorted data
betas = []
for level in sex:
    x = data_sex[level]['levelFace']
    y = data_sex[level]['srespChoice'] == 'Goli'
    popt, pcov = curve_fit(sigmoid, x, y)
    betas.append(popt[1])

# plot Beta coeff in bar plot

plt.bar(sex, betas)
plt.xlabel('Genders')
plt.ylabel('Sensitivity (beta)')
plt.title('Sensitivity to detecting identity from genders')
plt.show()
```

Figure 14

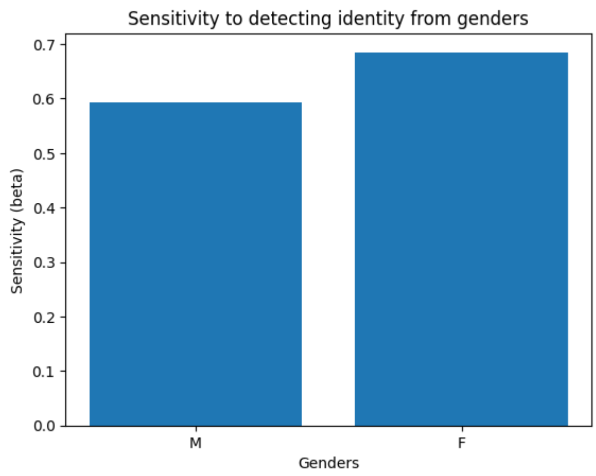

Figure 15

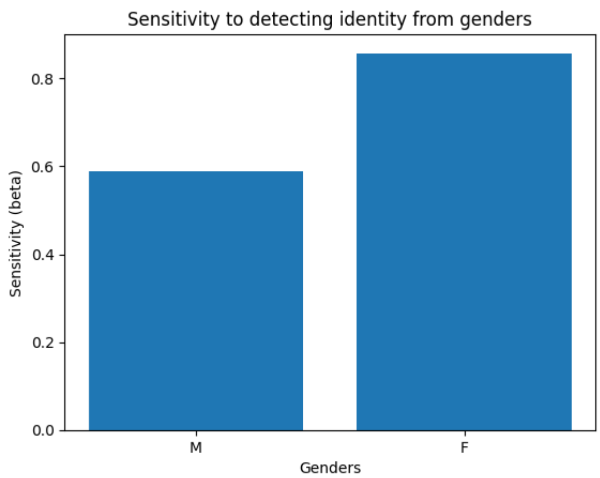

Figure 16

Also, for the statistical analysis of this part, we perform a statistical analysis using Z Test, the results of which are shown in Figure 17.

Z test analysis

```
[57]
data['gender'] = data.apply(hetro_homo, axis=1)
data['res'] = data.apply(add_result_col_to_Data, axis=1)

success = data.groupby(['gender']).sum()
total = data.groupby(['gender']).count()
hetro_s, hetro_total = success['res']['hetro'], total['res']['hetro']
homo_s ,homo_total = success['res']['homo'], total['res']['homo']

x1, y1, z1 = Ztest(hetro_s/hetro_total, hetro_total, homo_s/homo_total,
print('p-value for homo/hetro: ' + str(x1))
print('hetro probablity: ' + str(y1))
print('homo probablity: ' + str(z1))
```

```
p-value for homo/hetro: 4.405732809590035e-10
hetro probablity: 0.8224206349206349
homo probablity: 0.8008365508365508
```

Figure 17

Based on the data obtained from the statistical analysis, men have performed equally in both cases and women have performed better in recognizing the identity of the opposite sex.

Also, the areas under the ROC diagram are shown in Figures 18 and 19.

```
beta
[0.5921940447891488]
AUC score
0.7522467320261438
beta
[0.5921940447891488, 0.6851745514520942]
AUC score
0.7895833333333333
```

Figure 18

With a good probability, it can be claimed that the sensitivity parameter is the same for men in all three stages of the test (LF, HF, IF).

```
beta
[0.5889951877507543]
AUC score
0.7522467320261438
beta
[0.5889951877507543, 0.8574595998887298]
AUC score
0.7895833333333333
```

Figure 19

Third question:

Can candidates recognize identities in a specific spectral band if they use their left hand or not?

For this, we have to use the Z score test. Then we check whether the person guessed correctly or not. After making sure that the correct guess or wrong guess is correct, we go to the person's gender and check whether the line is related to the same sex or not. Finally, we get a probability for homosexuals and non-homosexuals. Now we have to check whether the probability of homo is equal to hetero. The null hypothesis is the same as the mentioned equality, which states that the probability of correct guess is the same for both and has nothing to do with the gender of the question. The assumption of a but is the opposite, it means that a person is biased in one direction. Now we have to do the Z test.

In this part, we first separate the data based on each frequency band and compare the β parameter for each band for the right and left hand, the code for this part is shown in Figure 20, its bar chart is in Figure 21, 22 and 23.

hypotheses 3 : Identifying images by left hand in level Freq

```python
data_IF = data[data.levelFreq=='IF']
data_LF = data[data.levelFreq=='LF']
data_HF = data[data.levelFreq=='HF']

Hand_level = ['Left', 'Right']
data_IF_sorted = {level: data_IF[data_IF.Hand==level] for level in Hand_level}
data_LF_sorted = {level: data_LF[data_LF.Hand==level] for level in Hand_level}
data_HF_sorted = {level: data_HF[data_HF.Hand==level] for level in Hand_level}

# fit sigmoid on sorted data

betas_IF = []
for level in Hand_level:
    x = data_IF_sorted[level]['levelFace']
    y = data_IF_sorted[level]['srespChoice'] == 'Goli'
    popt, pcov = curve_fit(sigmoid, x, y)
    betas_IF.append(popt[1])
betas_LF = []
for level in Hand_level:
    x = data_LF_sorted[level]['levelFace']
    y = data_LF_sorted[level]['srespChoice'] == 'Goli'
    popt, pcov = curve_fit(sigmoid, x, y)
    betas_LF.append(popt[1])
betas_HF = []
for level in Hand_level:
    x = data_HF_sorted[level]['levelFace']
    y = data_HF_sorted[level]['srespChoice'] == 'Goli'
    popt, pcov = curve_fit(sigmoid, x, y)
    betas_HF.append(popt[1])
# plot Beta coeff in bar plot

plt.bar(Hand_level, betas_IF)
plt.xlabel('Hand used')
plt.ylabel('Sensitivity (beta)')
plt.title('Sensitivity to detecting identity from IF band')
plt.show()
```

Figure 20

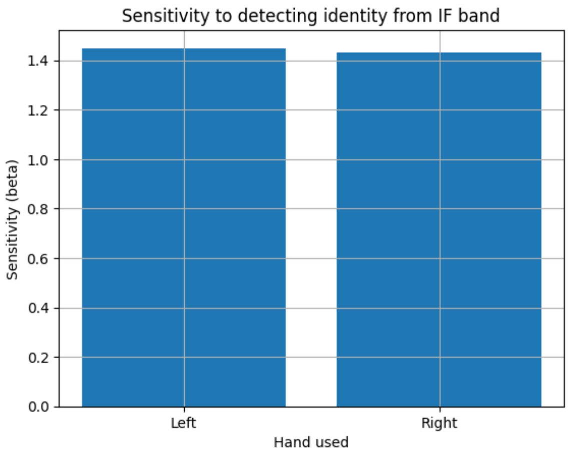

Figure 21

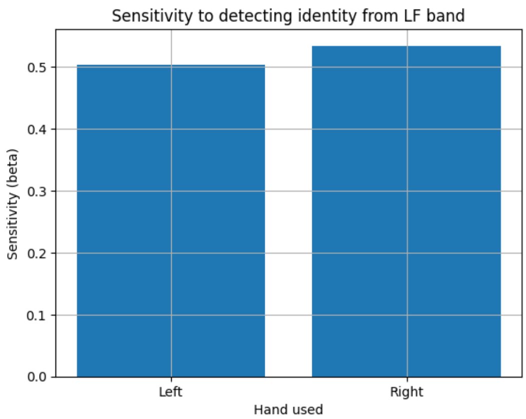

Figure 22

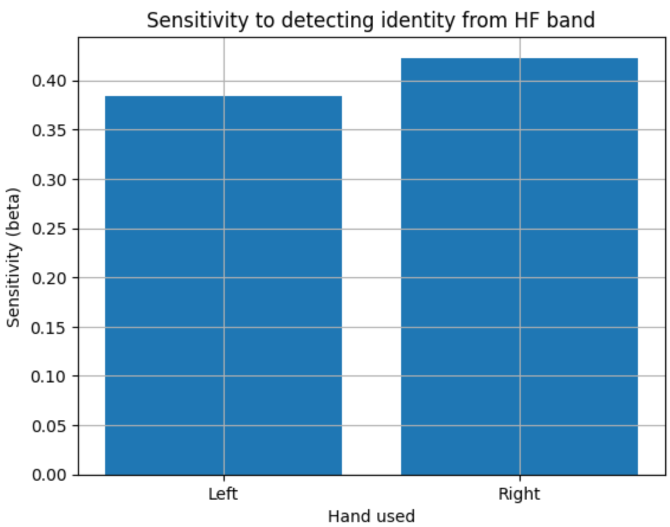

Figure 23

Also, for the statistical analysis of this part, we perform a statistical analysis using Z Test, the results of which are shown in Figure 24.

Z test analysis

```
[73]
p_value, p1, p2 = Ztest(R_HF_s/R_HF_total, R_HF_total,
print('p-value Right and Left hand HF: ' + str(p_value)
print('Right hand HF probality: ' + str(p1))
print('Left hand HF probality: ' + str(p2))
```

```
p-value Right and Left hand HF: 0.6290161441255026
Right hand HF probality: 0.759515977443609
Left hand HF probality: 0.7563439849624061
```

```
[74]
p_value, p1, p2 = Ztest(R_LF_s/R_LF_total, R_LF_total,
print('p-value Right and Left hand LF: ' + str(p_value)
print('Right hand LF probality: ' + str(p1))
print('Left hand LF probality: ' + str(p2))
```

```
p-value Right and Left hand LF: 0.8369052039454088
Right hand LF probality: 0.7874765037593985
Left hand LF probality: 0.7861842105263158
```

```
[75]
p_value, p1, p2 = Ztest(R_IF_s/R_IF_total, R_IF_total,
print('p-value Right and Left hand IF: ' + str(p_value)
print('Right hand IF probality: ' + str(p1))
print('Left hand IF probality: ' + str(p2))
```

```
p-value Right and Left hand IF: 0.35006606095978343
Right hand IF probality: 0.8887453007518797
Left hand IF probality: 0.8932095864661654
```

Figure 24

In different frequency ranges, based on the statistical analysis provided, the best performance has been obtained for the left hand in the IF frequency, followed by LF and HF respectively.

The values in the picture above represent the p-value, the probability of correct detection of the non-homogeneous test and the probability of correct detection of the same-sex test, respectively. Considering the low p-value, it can be concluded that the null hypothesis is strongly rejected. It means that the difference between two possibilities is meaningful and not a chance. So, it can be claimed that the probability of correct diagnosis is higher in the non-homogeneous test. In other words, people perform better in recognizing faces of the opposite sex.

Also, the area under the ROC diagram is shown in Figure 25.

```
beta
[1.4493544011349229]
AUC score
0.871984649122807
beta
[1.4493544011349229, 1.4299351066314872]
AUC score
0.8660910087719298
beta
[0.5032179193579617]
AUC score
0.7457510964912281
beta
[0.5032179193579617, 0.5343072363308027]
AUC score
0.7458881578947367
beta
[0.38367797166801976]
AUC score
0.7103892543859649
beta
[0.38367797166801976, 0.4226499338692281]
AUC score
0.7138157894736843
```

Figure 25

The area under the two graphs are very different from each other, and this difference cannot be meaningful and just happened by chance.

Fourth question:

Can candidates recognize identities in a specific spectral band if they use their dominant hand or not?

In this part, we first sort the data based on different frequency bands, then we obtain the value of the β parameter of the candidates for using the dominant hand in each frequency band and draw its bar graph, this is done in figures 26 and 27. and the code of this part is available in Figure 28.

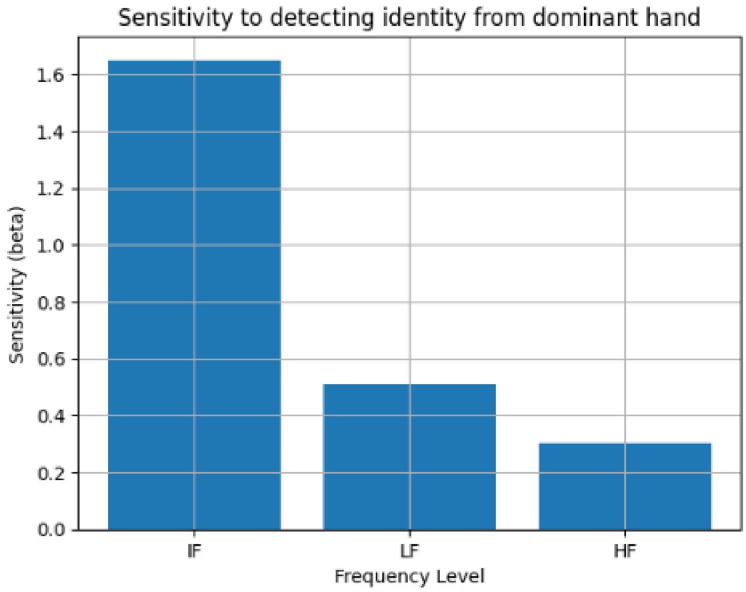

Figure 26

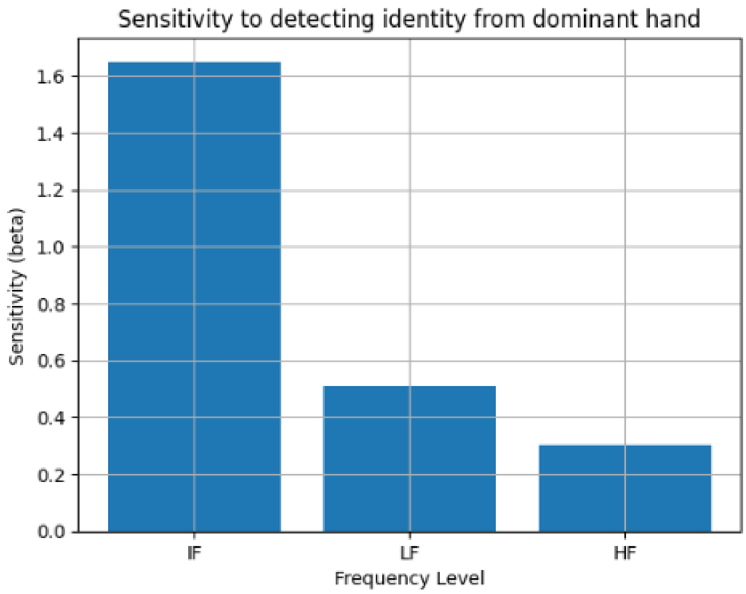

Figure 27

```
betas = []
# fit sigmoid on sorted data
for level in freq_levels:
    x = data_freq[level]['levelFace']
    x1 = data_freq[level]['Hand'] != data_freq[level]['dom']
    x2 = data_freq[level]['srespChoice'] == 'Goli'
    y = x1 &  x2
    popt, pcov = curve_fit(sigmoid, x, y)
    betas.append(popt[1])
# plot Beta coeff in bar plot

plt.grid()
plt.bar(freq_levels, betas)

plt.xlabel('Frequency Level')
plt.ylabel('Sensitivity (beta)')
plt.title('Sensitivity to dominant hand')

plt.show()
```

Figure 28

Also, for the statistical analysis of this part, we perform a statistical analysis using Z Test, the results of which are shown in Figure 29.

It can be said that if the dominant hand is used, the HF and LF stage can have relatively the same success rate. But the p value for the IF test with any other test is a very small value that rejects the null hypothesis. That is, if the dominant hand is used, the performance in IF is significantly different from other stages, and considering its high success rate (89%), this performance is on the positive side. That is, with the dominant hand, our performance in IF is much better than the other tests (HF and LF).

Z test analysis

```
[65] x1, x2, x3 = Ztest(D_HF_s/D_HF_total, D_HF_total, N_HF_s/N
     print('p-value dominant and non-dominant HF: ' + str(x1))
     print('dominant dominant HF probability: ' + str(x2))
     print('non-dominant dominant HF probability: ' + str(x3))

p-value dominant and non-dominant HF: 0.7883948181043998
dominant dominant HF probability: 0.7588110902255639
non-dominant dominant HF probability: 0.7570488721804511
```

```
[66] x1, x2, x3 = Ztest(D_LF_s/D_LF_total, D_LF_total, N_LF_s/N
     print('p-value dominant and non-dominant LF: ' + str(x1))
     print('dominant dominant LF probability: ' + str(x2))
     print('non-dominant dominant LF probability: ' + str(x3))

p-value dominant and non-dominant LF: 0.8957770477181338
dominant dominant LF probability: 0.7864191729323309
non-dominant dominant LF probability: 0.7872415413533834
```

```
[67] x1, x2, x3 = Ztest(D_IF_s/D_IF_total, D_IF_total, N_IF_s/N
     print('p-value dominant and non-dominant IF: ' + str(x1))
     print('dominant IF probability: ' + str(x2))
     print('non-dominant IF probability: ' + str(x3))

p-value dominant and non-dominant IF: 0.4911062874657419
dominant IF probability: 0.8893327067669173
non-dominant IF probability: 0.8926221804511278
```

```
[68] x1, x2, x3 = Ztest(D_HF_s/D_HF_total, D_HF_total, D_LF_s/D
     print('p-value dominant HF and LF: ' + str(x1))
     print('dominant HF probability: ' + str(x2))
     print('dominant LF probability: ' + str(x3))

p-value dominant HF and LF: 1.7305323060980966e-05
dominant HF probability: 0.7588110902255639
dominant LF probability: 0.7864191729323309
```

```
[69] x1, x2, x3 = Ztest(D_IF_s/D_IF_total, D_IF_total, D_LF_s/D
     print('p-value dominant IF and LF: ' + str(x1))
     print('dominant IF probability: ' + str(x2))
     print('dominant LF probability: ' + str(x3))

p-value dominant IF and LF: 3.833433030162703e-74
dominant IF probability: 0.8893327067669173
dominant LF probability: 0.7864191729323309
```

```
[70] x1, x2, x3 = Ztest(D_HF_s/D_HF_total, D_HF_total, D_IF_s/D
     print('p-value dominant HF and IF: ' + str(x1))
     print('dominant HF probability: ' + str(x2))
     print('dominant IF probability: ' + str(x3))

p-value dominant HF and IF: 8.992370214846495e-111
dominant HF probability: 0.7588110902255639
dominant IF probability: 0.8893327067669173
```

Figure 29

If we want to do an analysis using ROC, we can see that the area of almost all the graphs drawn with the dominant hand is almost close to the area of the graphs with the non-dominant hand. Therefore, it can be said that this small difference indicates that the difference between them is not significant. That is, their performance is not much different.

```
[17024 rows x 13 columns]}
beta
[1.4328179397893055]
AUC score
0.8690378289473684
beta
[1.4328179397893055, 0.537673260413891]
AUC score
0.7458196271929827
beta
[1.4328179397893055, 0.537673260413891, 0.41805188123686976]
AUC score
0.7121025219298246
```

Figure 30

```
beta
[1.4401525335783758]
AUC score
0.8690378289473684
beta
[1.4401525335783758, 0.5184904889641667]
AUC score
0.7458196271929827
beta
[1.4401525335783758, 0.5184904889641667, 0.4024688411001252]
AUC score
0.7121025219298246
```

Figure 31

Fifth question:

Are women better than men in identifying identities?

In this part, we act similar to the second part, we have given bar graphs for Hassan and Goli in figures 32 and 33, and the code of this part is available in figure 34.

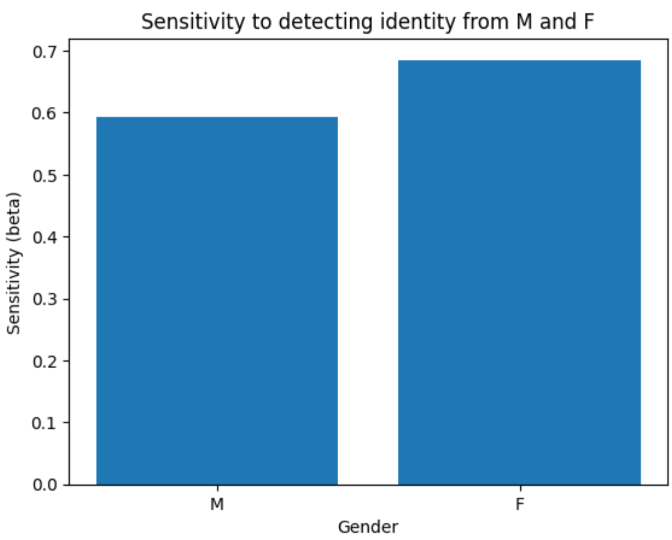

Figure 32

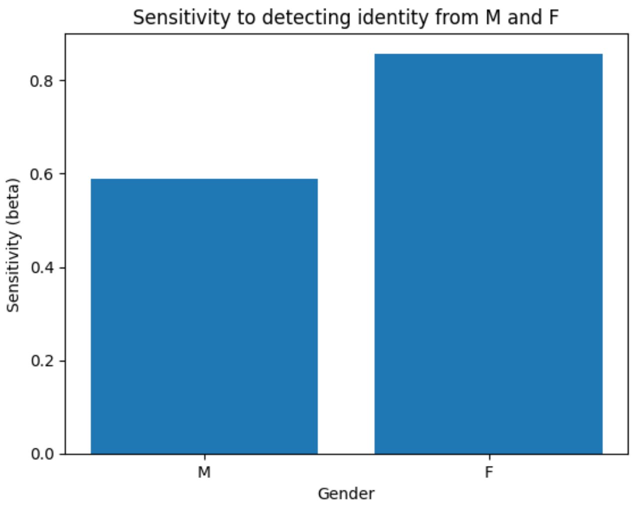

Figure 33

```python
data = pd.read_csv('/content/merged_data.csv', on_bad_lines='skip')



sex = ['M', 'F']
data_sex = {level: data[data.sex==level] for level in sex}
# print(data_sex)

# fit sigmoid on sorted data
betas = []
for level in sex:
    x = data_sex[level]['levelFace']
    y = data_sex[level]['srespChoice'] == 'Hasan'
    popt, pcov = curve_fit(sigmoid, x, y)
    betas.append(popt[1])

# plot Beta coeff in bar plot

plt.bar(sex, betas)
plt.xlabel('Gender')
plt.ylabel('Sensitivity (beta)')
plt.title('Sensitivity to detecting identity from M and F')
plt.show()
```

Figure 34

Also, for the statistical analysis of this part, we perform a statistical analysis using Z Test, the results of which are shown in Figure 35.

Z test analysis

```
data['result'] = data.apply(add_result_col_to_Data, axis=1)
success = data.groupby(['sex']).sum()
total = data.groupby(['sex']).count()

M_s, M_total = success['result'].loc['M'], total['result'].loc['M']
F_s, F_total = success['result'].loc['F'], total['result'].loc['F']

x1, x2, x3 = Ztest(M_s/M_total, M_total, F_s/F_total, F_total), M_s/
print('p-value Male and Female: ' + str(x1))
print('Male probability: ' + str(x2))
print('Female probability: ' + str(x3))
```

```
p-value Male and Female: 1.0787338302080971e-22
Male probability: 0.7904849439775911
Female probability: 0.8251488095238095
```

Figure 35

According to the presented statistical analysis, women have performed better than men in recognizing identity.

we see that the p value is very small and rejects the null hypothesis that the success rate of both sexes is equal. In other words, it cannot be claimed that the gender of the candidates has no effect on the success rate. Now that it is not ineffective, it should be seen which gender was able to perform better. According to the table above, the success rate of women is higher. Therefore, it can be said that the success rate of women is definitely higher than that of men. That is, women perform better in face recognition and can better distinguish between similar faces.

Also, the area under the ROC diagram is shown in Figures 36 and 37.

```
beta
[0.5921940447891488]
AUC score
0.7522467320261438
beta
[0.5921940447891488, 0.6851745514520942]
AUC score
0.7895833333333333
```

Figure 36

```
beta
[0.5889951877507543]
AUC score
0.7522467320261438
beta
[0.5889951877507543, 0.8574595998887298]
AUC score
0.7895833333333333
```

Figure 37

The ROC diagram in this case is shown in Figure 38.

If we want to present our analysis through the ROC chart, we can see that the area under the chart for the female gender is much more than the male gender chart. Therefore, it can be claimed that the female gender has a significantly better performance than the male gender.

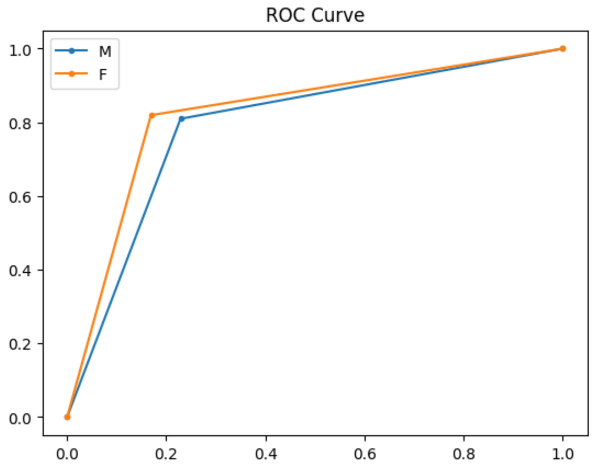

Figure 38

3. Make a Hypothesis:

In this part, we are going to check the data again by presenting a new question and statistical analysis on it.

Question: Is the rate of identity recognition different for men and women in different frequencies?

For this case, we first sort the data based on frequency and check the male and female operators separately in each frequency.

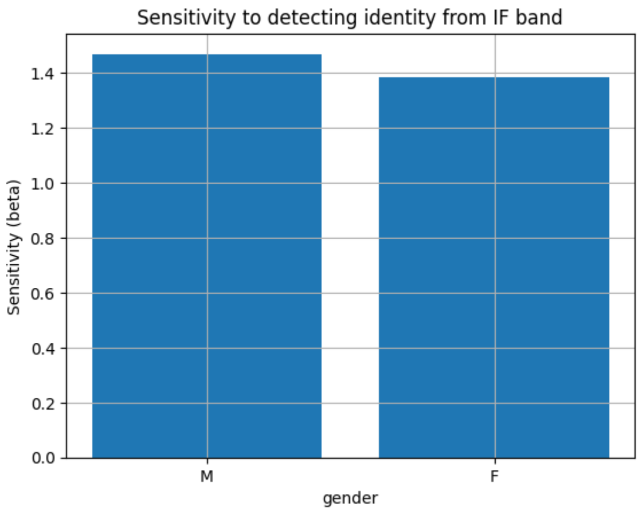

Figure 39

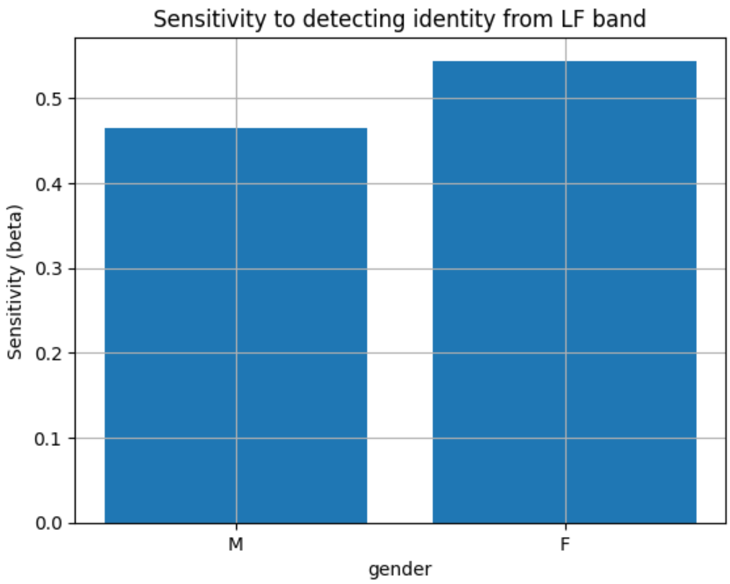

Figure 40

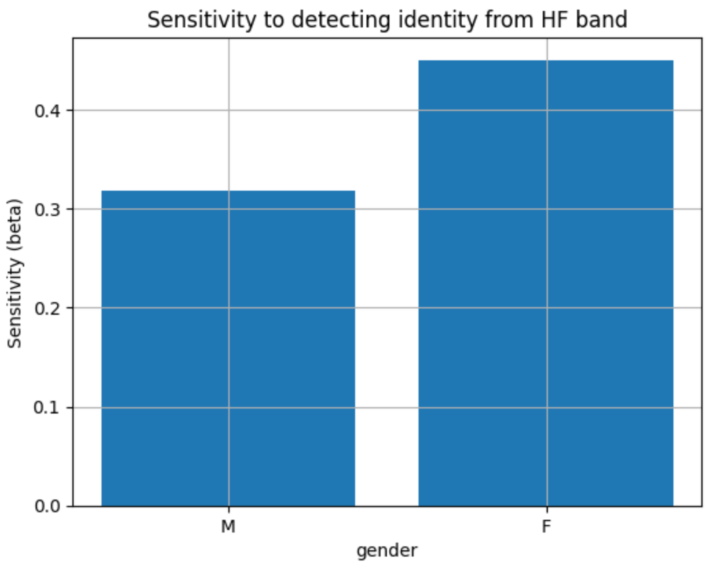

Figure 41

Also, its statistical analysis performed by z test is shown in Figure 42.

```
p_value, p1, p2 = z_test_func(R_HF_s/R_HF_tot, R_HF_tot,
print('p-value Right and Left hand HF: ' + str(p_value))
print('Right hand HF probality: ' + str(p1))
print('Left hand HF probality: ' + str(p2))

p-value Right and Left hand HF: 1.5593127469928302e-13
Right hand HF probality: 0.7775669642857143
Left hand HF probality: 0.7280724789915967

[91]
p_value, p1, p2 = z_test_func(R_LF_s/R_LF_tot, R_LF_tot,
print('p-value Right and Left hand HF: ' + str(p_value))
print('Right hand HF probality: ' + str(p1))
print('Left hand HF probality: ' + str(p2))

p-value Right and Left hand HF: 0.001046497288379553
Right hand HF probality: 0.7938616071428571
Left hand HF probality: 0.7728466386554622

p_value, p1, p2 = z_test_func(R_IF_s/R_IF_tot, R_IF_tot,
print('p-value Right and Left hand HF: ' + str(p_value))
print('Right hand HF probality: ' + str(p1))
print('Left hand HF probality: ' + str(p2))

p-value Right and Left hand HF: 8.55437039726457e-12
Right hand HF probality: 0.9040178571428571
Left hand HF probality: 0.8705357142857143
```

Figure 42

The area under the ROC diagram in this case is shown in Figure 43.

```
beta
[1.4687573724756087]
AUC score
0.8466605392156862
beta
[1.4687573724756087, 1.3827603556201722]
AUC score
0.8829427083333334
beta
[0.46391917464809396]
AUC score
0.7313112745098038
beta
[0.46391917464809396, 0.5438643047864428]
AUC score
0.75234375
beta
[0.31842258542199015]
AUC score
0.6787683823529412
beta
[0.31842258542199015, 0.450447372703915]
AUC score
0.7334635416666667
```

Figure 43

The ROC diagram in this case is shown in Figure 44.

If we want to present our analysis through the ROC chart, we can see that the area under the chart for the female gender is much more than the male gender chart. Therefore, it can be claimed that the female gender has a significantly better performance than the male gender.

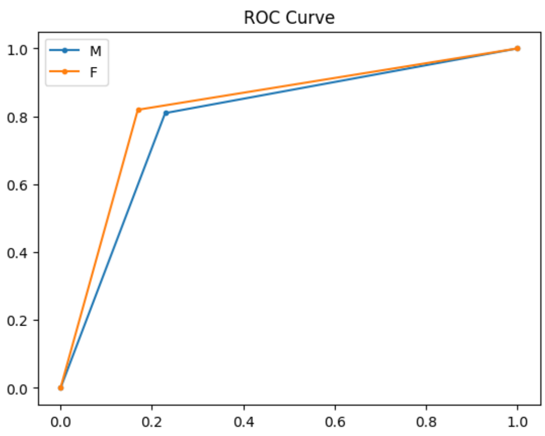

Figure 44

4. Mixture analysis, Mixed Trials

Question : we will compare the sensitivity of subjects in the trials of the first three blocks with the trials of the fourth block.
by knowing that the next set of images would be from a specific category, such as LF images, our performance might change compared to the mixed blocks.
In this code, grouping and separation based on BlockType.
In this part as well as in other parts, it is enough to evaluate the user's performance in various stages and the combination stage and test it using Z test to see if the difference between the performances is significantly far from each other or if this difference is random and indicates It is not true.

```
x1, x2, x3 = z_test_func(M_s/M_t, M_t, S_s/S_t, S_t), M_s/M_t, S_s/S_t
print('p-value Mixed , Same: ' + str(x1))
print('Mixed probability: ' + str(x2))
print('Same probability: ' + str(x3))
```

```
p-value Mixed , Same: 8.396562244611553e-10
Mixed probability: 0.8013001253132832
Same probability: 0.8225250626566416
```

Figure 45

By performing the Z test, we find that the P value is a very small number, so we can firmly reject the initial assumption that the performance of people in these two types of stages is the same. Since the assessment is less in the combined stage, it can be said that the candidate has spent more energy by spending a lot of time on the same steps, and when entering the combined stage, the candidate's low energy has led to more cognitive errors.

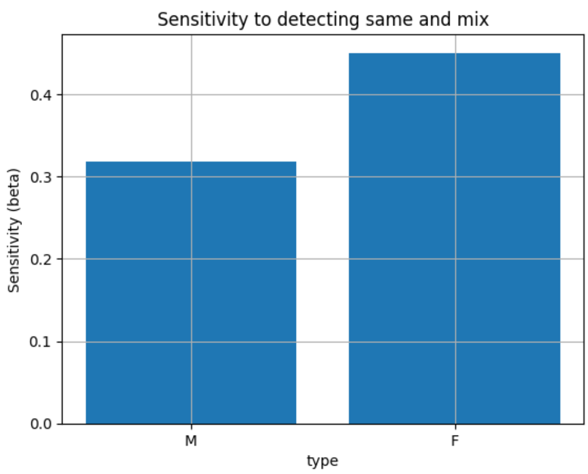

Figure 46

Also, if we calculate area under ROC diagram, we will get to the following number. As can be seen from the data, it can be said that in the combined state, the area under the graph is less than the area under the graph for the same states.

5. Confounds: Aiming to Unbiased Interpretation

Question : identify any confounds that may have existed in the research design the Trial.

Based on the information provided, there are a few potential confounds that may have existed in the research design of this trial:

1.Order effects: The order in which the participants complete the different blocks of the trial (with different spatial frequency filters) may impact their performance. For example, if participants complete the low spatial frequency block first, they may become more sensitive to low spatial frequencies and this may influence their performance on subsequent blocks. This could confound the results, as any differences in performance across spatial frequencies may be due to the order in which the blocks were completed rather than the actual spatial frequency manipulation.

2.Learning effects: Participants may become more familiar with the faces as they progress through the trial, which may influence their ability to detect the identities accurately. This could confound the results, as any differences in performance across spatial frequencies may be due to learning effects rather than the actual spatial frequency manipulation.

3.Response bias: Participants may have a response bias towards selecting male or female names, regardless of the identity of the presented face. This could confound the results, as any differences in performance across spatial frequencies may be due to response bias rather than the actual spatial frequency manipulation.

To address these potential confounds, the researchers could counterbalance the order in which participants complete the different blocks of the trial, so that half of the participants complete the low spatial frequency block first, and the other half complete the high spatial frequency block first. Additionally, the researchers could include a control group that does not receive any spatial frequency filters, to compare performance with and without the filters. Finally, the researchers could analyze the data for response bias and include this as a covariate in the statistical analysis if necessary.