

HW5 Selected problem solutions

Part I

Step1

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats as ss
```

```
Data_Path = 'C:/Users/seh00004.UNCNN/Desktop/CSE5520/HW/Data/'
df = pd.read_csv(Data_Path + 'GSE44768_CR_40_54_Combined.csv')
```

```
alz_f = df[df.GSM_ID.str.contains('alz_f')]
nd_f = df[df.GSM_ID.str.contains('nd_f')]
alz_m = df[df.GSM_ID.str.contains('alz_m')]
nd_m = df[df.GSM_ID.str.contains('nd_m')]
```

```
df_alzf_stat, ndf_alzf_pval = ss.ttest_ind(a=nd_f.LAMP2, b=alz_f.LAMP2, equal_var=False)
ndf_ndm_stat, ndf_ndm_pval = ss.ttest_ind(a=nd_f.LAMP2, b=nd_m.LAMP2, equal_var=False)
ndf_alzm_stat, ndf_alzm_pval = ss.ttest_ind(a=nd_f.LAMP2, b=alz_m.LAMP2, equal_var=False)
alz_f_nd_m_stat, alz_f_nd_m_pval = ss.ttest_ind(a=alz_f.LAMP2, b=nd_m.LAMP2, equal_var=False)
alz_f_alzm_stat, alz_f_alzm_pval = ss.ttest_ind(a=alz_f.LAMP2, b=alz_m.LAMP2, equal_var=False)
nd_m_alzm_stat, nd_m_alzm_pval = ss.ttest_ind(a=nd_m.LAMP2, b=alz_m.LAMP2, equal_var=False)
p_val = [ndf_alzf_pval, ndf_ndm_pval, ndf_alzm_pval, alz_f_nd_m_pval, alz_f_alzm_pval, nd_m_alzm_pval]
#print(p_val)
print(ndf_alzf_pval, '\n', ndf_ndm_pval, '\n', ndf_alzm_pval, '\n', alz_f_nd_m_pval, '\n', alz_f_alzm_pval, '\n', nd_m_alzm_pval)
```

```
0.0011632520081142283
0.20222307887306107
0.10454587853574349
0.015605589934773087
0.10711258991742761
0.5195917095241882
```

Step2

```
import dabest
df2 = pd.concat([nd_f, alz_f, nd_m, alz_m])
for i in range(len(df2.GSM_ID)):
    if 'alz_f' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'alz_f'
    elif 'nd_f' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'nd_f'
    elif 'alz_m' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'alz_m'
    elif 'nd_m' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'nd_m'
df2 = df2.sort_index()
```

```
fig, axes = plt.subplots(2, 3, figsize=(18, 18))
```

```
# 1A: Non-Dementia Female vs. Alzheimer's Female
```

```

axes[0,0].set_title("HW5.2.a: Non-Dementia Female vs. Alzheimer's Female", fontweight="bold")
ndf_alzf_dabest = dabest.load(data=df2, x="GSM_ID", y="LAMP2", idx=("nd_f", "alz_f"))
ndf_alzf_dabest.mean_diff.plot(ax=axes[0,0])

# 1B: Non-Dementia Female vs. Non-Dementia Male
axes[0,1].set_title("HW5.2.b: Non-Dementia Female vs. Non-Dementia Male", fontweight="bold")
ndf_ndm_dabest = dabest.load(data=df2, x="GSM_ID", y="LAMP2", idx=("nd_f", "nd_m"))
ndf_ndm_dabest.mean_diff.plot(ax=axes[0,1])

# 1C: Non-Dementia Female vs. Alzheimer's Male
axes[0,2].set_title("HW5.2.d: Non-Dementia Female vs. Alzheimer's Male", fontweight="bold")
ndf_alzm_dabest = dabest.load(data=df2, x="GSM_ID", y="LAMP2", idx=("nd_f", "alz_m"))
ndf_alzm_dabest.mean_diff.plot(ax=axes[0,2])

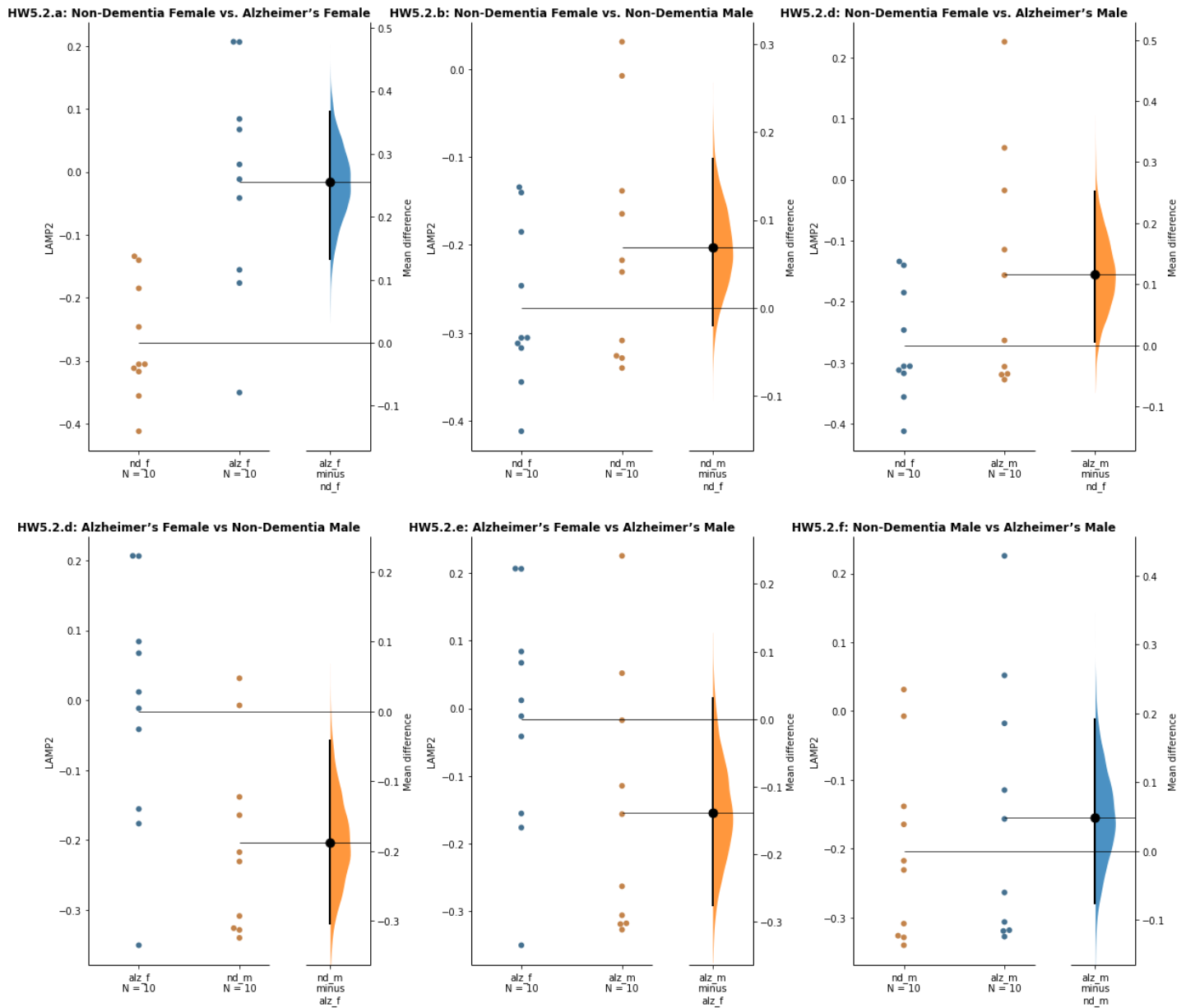
# 1D: Alzheimer's Female vs Non-Dementia Male
axes[1,0].set_title("HW5.2.d: Alzheimer's Female vs Non-Dementia Male", fontweight="bold")
alz_f_ndm_dabest = dabest.load(data=df2, x="GSM_ID", y="LAMP2", idx=("alz_f", "nd_m"))
alz_f_ndm_dabest.mean_diff.plot(ax=axes[1,0])

# 1E: Alzheimer's Female vs Alzheimer's Male
axes[1,1].set_title("HW5.2.e: Alzheimer's Female vs Alzheimer's Male", fontweight="bold")
alz_f_alzm_dabest = dabest.load(data=df2, x="GSM_ID", y="LAMP2", idx=("alz_f", "alz_m"))
alz_f_alzm_dabest.mean_diff.plot(ax=axes[1,1])

# 1F: Non-Dementia Male vs Alzheimer's Male
axes[1,2].set_title("HW5.2.f: Non-Dementia Male vs Alzheimer's Male", fontweight="bold")
ndm_alzm_dabest = dabest.load(data=df2, x="GSM_ID", y="LAMP2", idx=("nd_m", "alz_m"))
ndm_alzm_dabest.mean_diff.plot(ax=axes[1,2])

plt.subplots_adjust(wspace=0.9)
plt.show()

```



Step3

```
sorted_pvals = sorted(p_val)
siz=float(len(sorted_pvals))
alpha=0.05
```

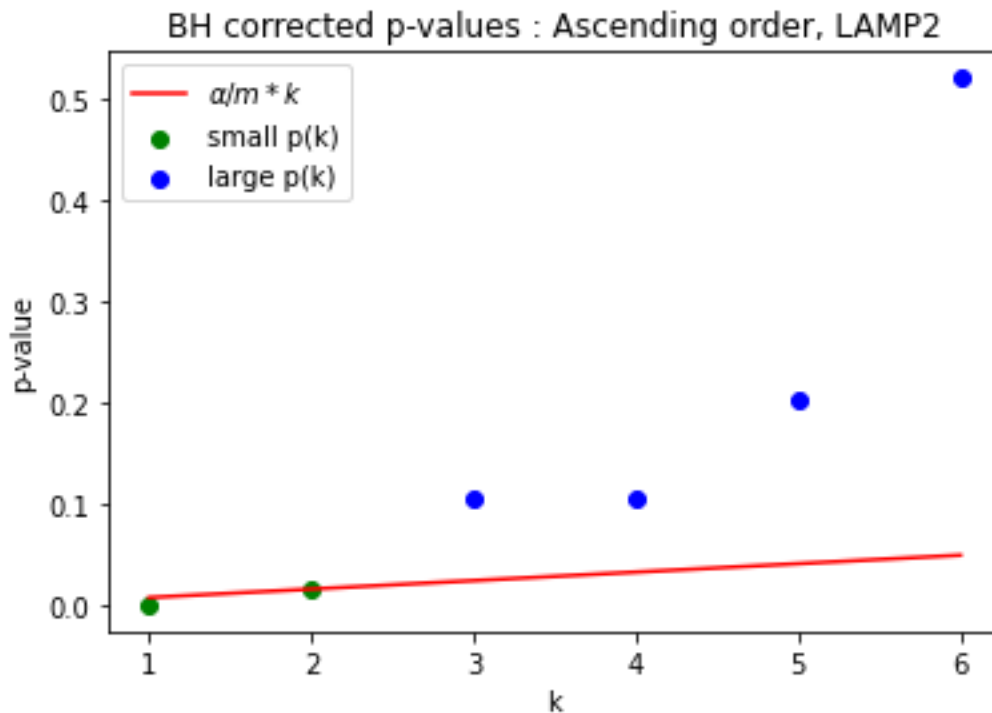
```
x=[float(i) for i in range(1,7)]
xx=np.asarray(x)
yy=alpha/siz*xx
temp=yy-sorted_pvals
#print(temp)
ind=np.where(temp > 0)
max_ind=(max(max(ind)))
m_ind=np.arange(max_ind+1)
u_ind=np.arange(max_ind+1,int(siz))
#print(m_ind,u_ind)
```

```

selected_elements = []
selected_elements = [sorted_pvals[i] for i in m_ind]
print(selected_elements)
unselected_elements = []
unselected_elements = [sorted_pvals[i] for i in u_ind]
#print(unselected_elements)

plt.xlabel("k")
plt.ylabel("p-value")
plt.title("BH corrected p-values : Ascending order, LAMP2")
plt.scatter([i for i in np.arange(1,len(selected_elements)+1)], selected_elements, label="small p(k)", color='g')
plt.scatter([i for i in np.arange(len(selected_elements)+1,7)], unselected_elements, label="large p(k)", color='b')
plt.plot([i for i in range(1, 7)], [(alpha/siz)*k for k in range(1, 7)], "r", label=r"$\alpha/m * k$")
plt.legend(frameon=True)
plt.show()

```



Step4

```

ndf_alzf_stat, ndf_alzf_pval = ss.ttest_ind(a=nd_f.BSN, b=alz_f.BSN, equal_var=False)
ndf_ndm_stat, ndf_ndm_pval = ss.ttest_ind(a=nd_f.BSN, b=nd_m.BSN, equal_var=False)
ndf_alzm_stat, ndf_alzm_pval = ss.ttest_ind(a=nd_f.BSN, b=alz_m.BSN, equal_var=False)
alz_f_nd_m_stat, alz_f_nd_m_pval = ss.ttest_ind(a=alz_f.BSN, b=nd_m.BSN, equal_var=False)
alz_f_alzm_stat, alz_f_alzm_pval = ss.ttest_ind(a=alz_f.BSN, b=alz_m.BSN, equal_var=False)
nd_m_alzm_stat, nd_m_alzm_pval = ss.ttest_ind(a=nd_m.BSN, b=alz_m.BSN, equal_var=False)
p_val= [ndf_alzf_pval,ndf_ndm_pval,ndf_alzm_pval,alz_f_nd_m_pval,alz_f_alzm_pval,nd_m_alzm_pval]
#print(p_val)
print(ndf_alzf_pval,'\n',ndf_ndm_pval,'\n',ndf_alzm_pval,'\n',alz_f_nd_m_pval,'\n',alz_f_alzm_pval,'\n',nd_m_alzm_pval)

```

```

0.03216540899668569
0.6241373504007757
0.7055896137725121
0.04885633580221915
0.04686251843546239
0.9179576528461968

```

```

df2 = pd.concat([nd_f, alz_f, nd_m, alz_m])
for i in range(len(df2.GSM_ID)) :
    if 'alz_f' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'alz_f'
    elif 'nd_f' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'nd_f'
    elif 'alz_m' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'alz_m'
    elif 'nd_m' in df2.GSM_ID.iloc[i]:
        df2.GSM_ID.iloc[i] = 'nd_m'
df2=df2.sort_index()

```

```

fig, axes = plt.subplots(2, 3, figsize=(18, 18))

```

```

# 1A: Non-Dementia Female vs. Alzheimer's Female

```

```

axes[0,0].set_title("HW5.2.a: Non-Dementia Female vs. Alzheimer's Female", fontweight="bold")
ndf_alzf_dabest = dabest.load(data=df2, x="GSM_ID", y="BSN", idx=("nd_f", "alz_f"))
ndf_alzf_dabest.mean_diff.plot(ax=axes[0,0])

```

```

# 1B: Non-Dementia Female vs. Non-Dementia Male

```

```

axes[0,1].set_title("HW5.2.b: Non-Dementia Female vs. Non-Dementia Male", fontweight="bold")
ndf_ndm_dabest = dabest.load(data=df2, x="GSM_ID", y="BSN", idx=("nd_f", "nd_m"))
ndf_ndm_dabest.mean_diff.plot(ax=axes[0,1])

```

```

# 1C: Non-Dementia Female vs. Alzheimer's Male

```

```

axes[0,2].set_title("HW5.2.d: Non-Dementia Female vs. Alzheimer's Male", fontweight="bold")
ndf_alzm_dabest = dabest.load(data=df2, x="GSM_ID", y="BSN", idx=("nd_f", "alz_m"))
ndf_alzm_dabest.mean_diff.plot(ax=axes[0,2])

```

```

# 1D: Alzheimer's Female vs Non-Dementia Male

```

```

axes[1,0].set_title("HW5.2.d: Alzheimer's Female vs Non-Dementia Male", fontweight="bold")
alz_f_ndm_dabest = dabest.load(data=df2, x="GSM_ID", y="BSN", idx=("alz_f", "nd_m"))
alz_f_ndm_dabest.mean_diff.plot(ax=axes[1,0])

```

```

# 1E: Alzheimer's Female vs Alzheimer's Male

```

```

axes[1,1].set_title("HW5.2.e: Alzheimer's Female vs Alzheimer's Male", fontweight="bold")
alz_f_alzm_dabest = dabest.load(data=df2, x="GSM_ID", y="BSN", idx=("alz_f", "alz_m"))
alz_f_alzm_dabest.mean_diff.plot(ax=axes[1,1])

```

```

# 1F: Non-Dementia Male vs Alzheimer's Male

```

```

axes[1,2].set_title("HW5.2.f: Non-Dementia Male vs Alzheimer's Male", fontweight="bold")
ndm_alzm_dabest = dabest.load(data=df2, x="GSM_ID", y="BSN", idx=("nd_m", "alz_m"))
ndm_alzm_dabest.mean_diff.plot(ax=axes[1,2])

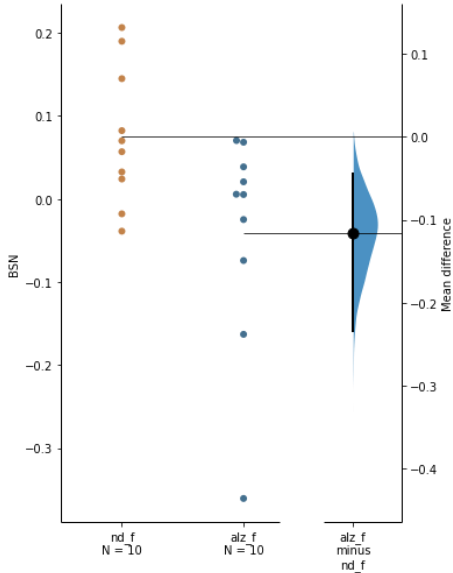
```

```

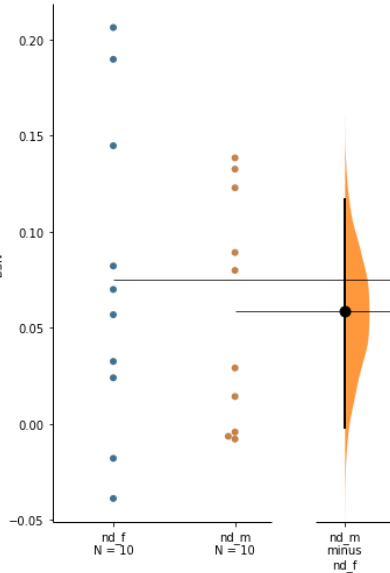
plt.subplots_adjust(wspace=0.9)
plt.show()

```

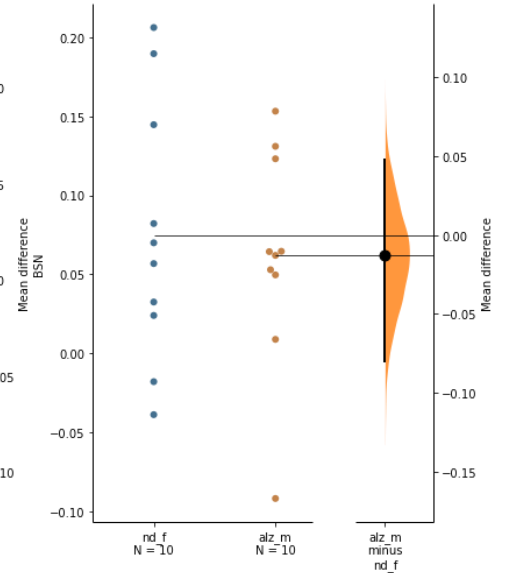
HW5.2.a: Non-Dementia Female vs. Alzheimer's Female



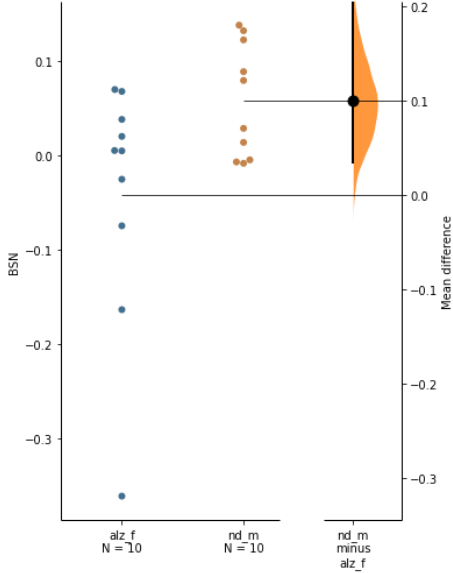
HW5.2.b: Non-Dementia Female vs. Non-Dementia Male



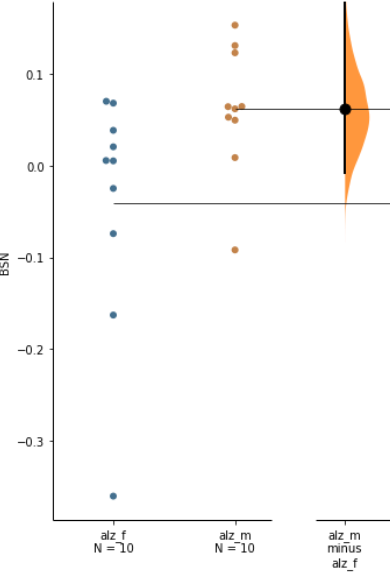
HW5.2.d: Non-Dementia Female vs. Alzheimer's Male



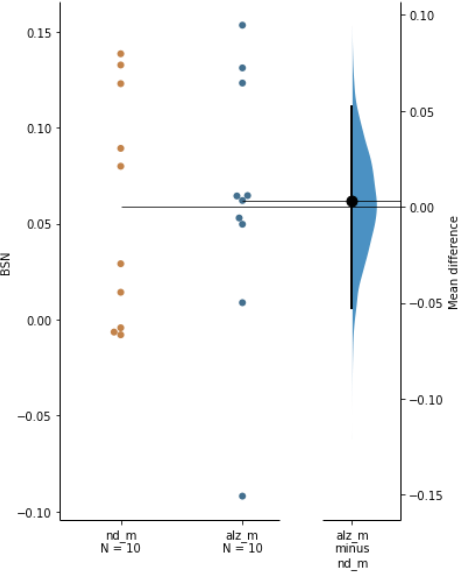
HW5.2.d: Alzheimer's Female vs Non-Dementia Male



HW5.2.e: Alzheimer's Female vs Alzheimer's Male



HW5.2.f: Non-Dementia Male vs Alzheimer's Male



```
sorted_pvals = sorted(p_val)
siz=float(len(sorted_pvals))
alpha=0.05
```

```
x=[float(i) for i in range(1,7)]
xx=np.asarray(x)
yy=alpha/siz*xx
temp=yy-sorted_pvals
#print(temp)
ind=[]
#s_ind=[i+1 for i in temp[i] > 0)
if len(temp[temp>0]):
    print(' ind')
    max_ind=(max(max(ind)))
    m_ind=np.arange(max_ind+1)
    u_ind=np.arange(max_ind+1,int(siz))
else:
```

```

m_ind=[]
u_ind=np.arange(0,int(siz))

#print(m_ind,u_ind)

selected_elements = []
selected_elements = [sorted_pvals[i] for i in m_ind]
print(selected_elements)
unselected_elements = []
unselected_elements = [sorted_pvals[i] for i in u_ind]
#print(unselected_elements)

plt.xlabel("k")
plt.ylabel("p-value")
plt.title("BH corrected p-values : Ascending order, BSN")
plt.scatter([i for i in np.arange(1,len(selected_elements)+1)], selected_elements, label="small p(k)", color='g')
plt.scatter([i for i in np.arange(len(selected_elements)+1,7)], unselected_elements, label="large p(k)", color='b')
plt.plot([i for i in range(1, 7)], [(alpha/siz)*k for k in range(1, 7)], "r", label=r"$\alpha/m * k$")
plt.legend(frameon=True)
plt.show()

```

