

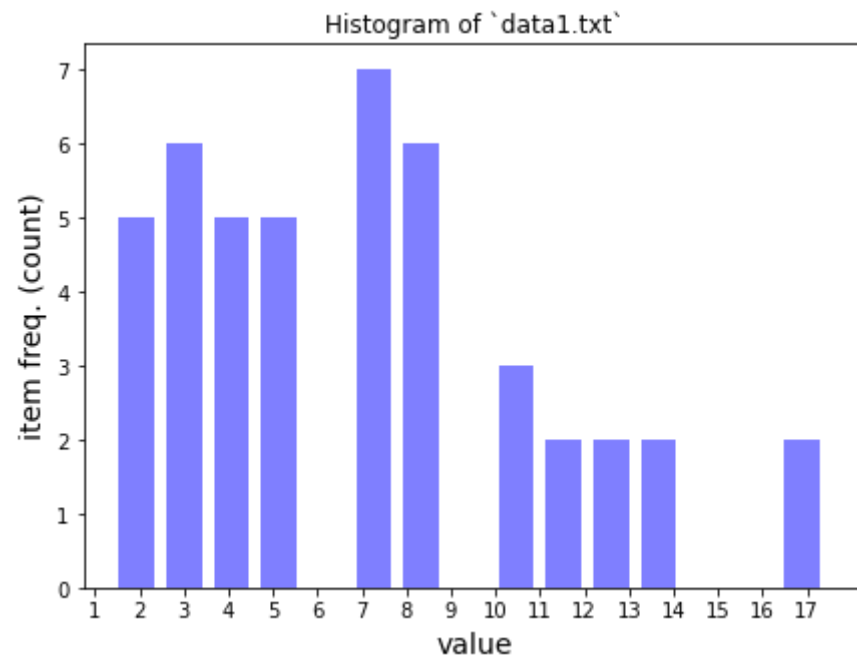
```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
```

Question Q1-1

```
In [2]: df_1 = pd.read_csv('/Users/gta/Desktop/homework-02/Data_HW01/data1.txt', sep='\t', header=None)

df_list = df_1.values.tolist()
flat_list = [each_i for each_j in df_list for each_i in each_j]

plt.figure(figsize=(7,5))
plt.xlabel('value', fontsize=14)
plt.ylabel('item freq. (count)', fontsize=14)
plt.title('Histogram of `data1.txt`')
plt.hist(flat_list, bins=15, color='b', alpha=0.5, align='right', width=0.8)
plt.xticks(range(1, 18))
plt.show()
```

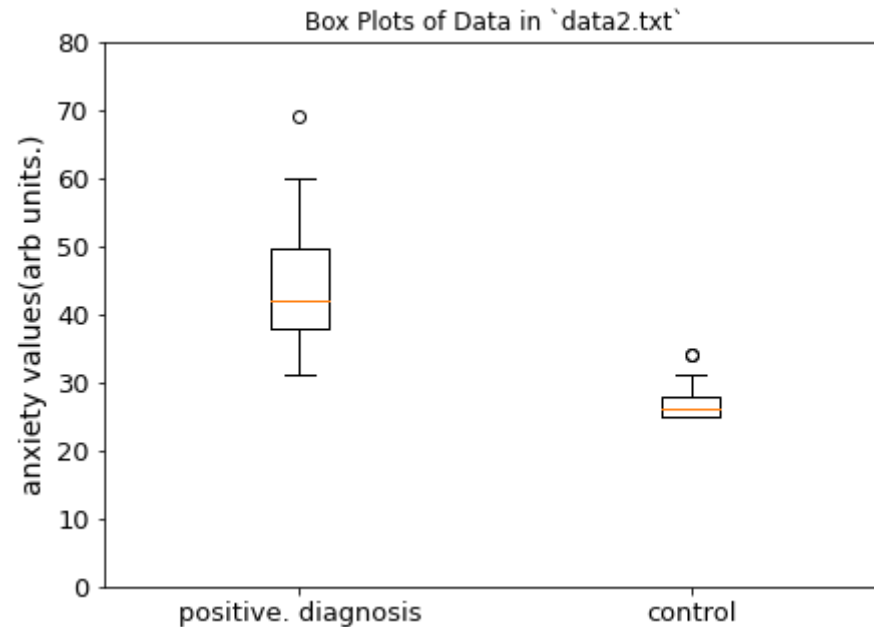


Question Q1-2

```
In [3]: df_2 = pd.read_csv('/Users/gta/Desktop/homework-02/Data_HW01/data2.txt', sep='\t', header=None)

diagnosis_data_mental_pos = df_2.values.tolist()[0]
diagnosis_data_mental_cont = [each for each in df_2.values.tolist()[1] if not math.isnan(each)]

plt.figure(figsize=(7,5))
plt.boxplot([diagnosis_data_mental_pos, diagnosis_data_mental_cont])
plt.ylabel('anxiety values(arb units.)', fontsize=14)
plt.xticks([1, 2], ['positive. diagnosis', 'control'], fontsize=13)
plt.ylim(0, 80, 10)
plt.yticks(fontsize=13)
plt.title('Box Plots of Data in `data2.txt`')
plt.show()
```



Question Q1-3

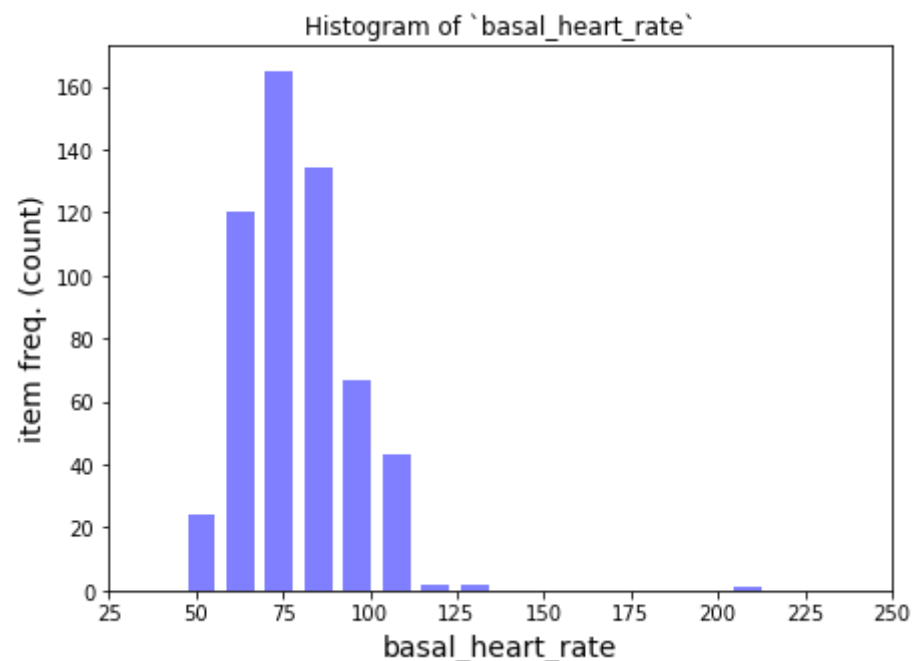
When the data provided is analyzed via the box plots, the data suggests that the anxiety value of the patients with a positive diagnosis of a mental disorder is higher than the control group.

Question Q2-1

```
In [4]: df_3 = pd.read_excel('/Users/gta/Desktop/homework-02/Data_HW01/CardiacData.xls', skiprows=[0])

df_basal_heart_rate_list = df_3[df_3.columns.tolist()[0]].values.tolist()

plt.figure(figsize=(7,5))
plt.xlabel('basal_heart_rate', fontsize=14)
plt.ylabel('item freq. (count)', fontsize=14)
plt.title('Histogram of `basal_heart_rate`')
plt.hist(df_basal_heart_rate_list, bins=15, color='b', alpha=0.5, align='right', width=8)
plt.xlim(25, 250, 25)
plt.show()
```



Question Q2-2

```
In [5]: df_3.columns.tolist()

# 'Heart Wall Motion Anomaly Observed (yes=0)',
# 'Baseline Cardiac Ejection Fraction'

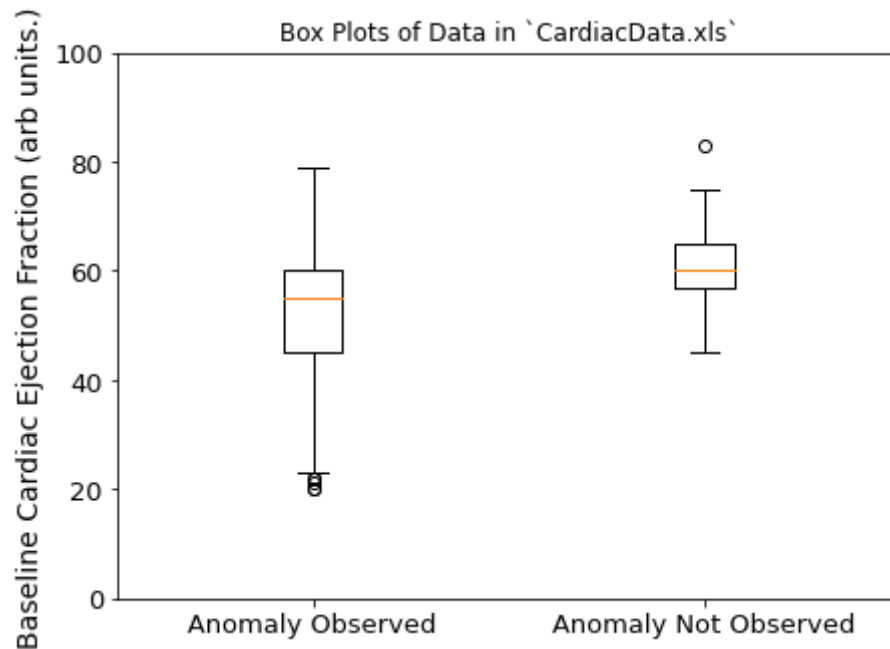
df_3_subset = df_3[['Baseline Cardiac Ejection Fraction', 'Heart Wall Motion Anomaly Observed (yes=0)']]
```

```

baseline_cardiac_ejection_fraction_anomaly_1 = df_3_subset[df_3_subset['Heart Wall Motion Anomaly Observed' == 'Yes']]
baseline_cardiac_ejection_fraction_anomaly_0 = df_3_subset[df_3_subset['Heart Wall Motion Anomaly Observed' == 'No']]

plt.figure(figsize=(7,5))
plt.boxplot([baseline_cardiac_ejection_fraction_anomaly_0, baseline_cardiac_ejection_fraction_anomaly_1])
plt.ylabel('Baseline Cardiac Ejection Fraction (arb units.)', fontsize=14)
plt.xticks([1, 2], ['Anomaly Observed', 'Anomaly Not Observed'], fontsize=13)
plt.ylim(0, 100, 15)
plt.yticks(fontsize=13)
plt.title('Box Plots of Data in `CardiacData.xls`')
plt.show()

```



Question Q2-3

In [6]:

```

baseline_cardiac_ejection_fraction_anomaly_1_mean = np.mean(baseline_cardiac_ejection_fraction_anomaly_1)
baseline_cardiac_ejection_fraction_anomaly_1_min = np.min(baseline_cardiac_ejection_fraction_anomaly_1)
baseline_cardiac_ejection_fraction_anomaly_1_max = np.max(baseline_cardiac_ejection_fraction_anomaly_1)
baseline_cardiac_ejection_fraction_anomaly_1_median = np.percentile(baseline_cardiac_ejection_fraction_anomaly_1, 50)
baseline_cardiac_ejection_fraction_anomaly_1_25th_pct = np.percentile(baseline_cardiac_ejection_fraction_anomaly_1, 25)
baseline_cardiac_ejection_fraction_anomaly_1_75th_pct = np.percentile(baseline_cardiac_ejection_fraction_anomaly_1, 75)

print('Summary Statistics for `baseline_cardiac_ejection_fraction_anomaly_*not*_observed`')

```

```

print('Mean: {}'.format(baseline_cardiac_ejection_fraction_anomaly_1_mean))
print('Min: {}'.format(baseline_cardiac_ejection_fraction_anomaly_1_min))
print('Max: {}'.format(baseline_cardiac_ejection_fraction_anomaly_1_max))
print('Median: {}'.format(baseline_cardiac_ejection_fraction_anomaly_1_median))
print('25th Percentile: {}'.format(baseline_cardiac_ejection_fraction_anomaly_1_25th_pct))
print('75th Percentile: {}'.format(baseline_cardiac_ejection_fraction_anomaly_1_75th_pct))

```

```

Summary Statistics for `baseline_cardiac_ejection_fraction_anomaly_*not*_observed`
Mean: 60.7431906614786
Min: 45
Max: 83
Median: 60.0
25th Percentile: 57.0
75th Percentile: 65.0

```

In [7]:

```

baseline_cardiac_ejection_fraction_anomaly_0_mean = np.mean(baseline_cardiac_ejection_fraction_anomaly_0)
baseline_cardiac_ejection_fraction_anomaly_0_min = np.min(baseline_cardiac_ejection_fraction_anomaly_0)
baseline_cardiac_ejection_fraction_anomaly_0_max = np.max(baseline_cardiac_ejection_fraction_anomaly_0)
baseline_cardiac_ejection_fraction_anomaly_0_median = np.percentile(baseline_cardiac_ejection_fraction_anomaly_0, 50)
baseline_cardiac_ejection_fraction_anomaly_0_25th_pct = np.percentile(baseline_cardiac_ejection_fraction_anomaly_0, 25)
baseline_cardiac_ejection_fraction_anomaly_0_75th_pct = np.percentile(baseline_cardiac_ejection_fraction_anomaly_0, 75)

print('Summary Statistics for `baseline_cardiac_ejection_fraction_anomaly_observed`')
print('Mean: {}'.format(baseline_cardiac_ejection_fraction_anomaly_0_mean))
print('Min: {}'.format(baseline_cardiac_ejection_fraction_anomaly_0_min))
print('Max: {}'.format(baseline_cardiac_ejection_fraction_anomaly_0_max))
print('Median: {}'.format(baseline_cardiac_ejection_fraction_anomaly_0_median))
print('25th Percentile: {}'.format(baseline_cardiac_ejection_fraction_anomaly_0_25th_pct))
print('75th Percentile: {}'.format(baseline_cardiac_ejection_fraction_anomaly_0_75th_pct))

```

```

Summary Statistics for `baseline_cardiac_ejection_fraction_anomaly_observed`
Mean: 51.21594684385382
Min: 20
Max: 79
Median: 55.0
25th Percentile: 45.0
75th Percentile: 60.0

```

Question Q2-4

In [8]:

```

df_3.columns.tolist()

# Column names.
# 'Any New Cardiac Event (yes=0)'

```

```
# 'Stress ECG Positive (yes=0)'

# Cross tab between 'Any New Cardiac Event (yes=0)', and 'Stress ECG Positive (yes=0)'
subset_col = ['Any New Cardiac Event (yes=0)', 'Stress ECG Positive (yes=0)']
df_4_subset = df_3[subset_col]
df_4_subset.groupby(subset_col)['Stress ECG Positive (yes=0)'].count()
```

```
Out[8]: Any New Cardiac Event (yes=0)  Stress ECG Positive (yes=0)
0                                     0                      46
      1                             1                      43
1                                     0                      90
      1                             1                     379
Name: Stress ECG Positive (yes=0), dtype: int64
```

```
In [9]: # Return all values from 'Any New Cardiac Event (yes=0)' when `Stress ECG Positive (yes=0)' == 1`
new_cardiac_event_stress_ecg_pos_no = df_4_subset[df_4_subset['Stress ECG Positive (yes=0)'] == 1]['Any New Ca

# Return all values from 'Any New Cardiac Event (yes=0)' when `Stress ECG Positive (yes=0)' == 0`
new_cardiac_event_stress_ecg_pos_yes = df_4_subset[df_4_subset['Stress ECG Positive (yes=0)'] == 0]['Any New Ca

# -----

# Return all values when 'Any New Cardiac Event (yes=0) == 0' when `Stress ECG Positive (yes=0)' == 1`
new_cardiac_event_yes_stress_ecg_pos_no = [each for each in new_cardiac_event_stress_ecg_pos_no if each == 0]

# Return all values when 'Any New Cardiac Event (yes=0) == 1' when `Stress ECG Positive (yes=0)' == 1`
new_cardiac_event_no_stress_ecg_pos_no = [each for each in new_cardiac_event_stress_ecg_pos_no if each == 1]

# Two lines of code below is the subset where `Stress ECG Positive (yes=0)' == 0` and
# counts if new cardiac events are found
# Return all values when 'Any New Cardiac Event (yes=0) == 0' when `Stress ECG Positive (yes=0)' == 0`
new_cardiac_event_yes_stress_ecg_pos_yes = [each for each in new_cardiac_event_stress_ecg_pos_yes if each == 0]

# Return all values when 'Any New Cardiac Event (yes=0) == 1' when `Stress ECG Positive (yes=0)' == 0`
new_cardiac_event_no_stress_ecg_pos_yes = [each for each in new_cardiac_event_stress_ecg_pos_yes if each == 1]

# -----

plt.figure(figsize=(3,5))
width = [0.5]

# subset where `Stress ECG Positive (yes=0)' == 0` and
# charts if new cardiac events are found (yes vs. no)
plt.bar(x='No', width=width, height=len(new_cardiac_event_no_stress_ecg_pos_yes), alpha=0.5, align='center', cc
```

```
plt.bar(x='Yes', width=width, height=len(new_cardiac_event_yes_stress_ecg_pos_yes), alpha=0.5, align='center',
plt.ylim(0, 100, 25)
plt.yticks(fontsize=14)
plt.ylabel('Count of Any New \n Cardiac Events \n (count)', fontsize=14)
plt.xticks(fontsize=14)
plt.xlabel('is_new_cardiac_event', fontsize=14)
plt.title('Distribution of New Cardiac \n Events for *Positive* SCG Stress \n')
plt.show()
```

