Part 1 – Data Description

Dataset 1

1. Name: Thyroid Disease Dataset – Hypothyroid Data
2. Source: UCI Machine Learning Repository
   https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease
3. Number of Instance: 3163
4. Description: Thyroid Disease Dataset that focused on hypothyroid disease that includes some information of the person and several different thyroid related measure result.
5. Missing Value: 0
6. Number of Attribute: 25
7. Attribute Information:

(1). Disease:	Hypothyroid / Negative
(2). Age:	Continuous
(3). Sex:	Male, Female, unknow
(4). On thyroxine:	T/F
(5). Query on thyroxine:	T/F
(6). On antithyroid medication:	T/F
(7). Thyroid surgery:	T/F
(8). Query hypothyroid:	T/F
(9). Query hyperthyroid:	T/F
(10). Pregnant	T/F
(11). Sick:	T/F
(12). Tumor:	T/F
(13). Lithium:	T/F
(14). Goiter:	T/F
(15). TSH measured:	T/F
(16). TSH:	Continuous
(17). T3 measured:	T/F

(18).	T3:	Continuous
(19).	TT4 measured:	T/F
(20).	TT4:	Continuous
(21).	T4U measured:	T/F
(22).	T4U:	Continuous
(23).	FTI measured:	T/F
(24).	FTI:	Continuous
(25).	TBG measured:	T/F
(26).	TBG:	Continuous

8. Dataset Preview

```
negative,45,F,t,f,f,t,f,f,f,f,f,f,f,y,0,y,2.30,y,206,y,1.05,y,196,n,?
negative,72,F,f,f,f,f,f,f,f,f,t,f,f,y,0,y,2.60,y,110,y,1.02,y,107,n,?
negative,51,F,f,f,f,f,f,f,f,f,f,f,f,y,0,y,2.10,y,93,y,0.87,y,106,n,?
negative,28,F,f,f,f,f,f,f,f,f,f,f,f,y,0,y,1.60,y,79,y,1.07,y,74,n,?
negative,24,F,f,f,f,f,f,f,f,f,f,f,f,y,0.85,y,3,y,121,y,1.25,y,97,n,?
negative,60,M,t,f,f,f,f,f,f,f,f,f,f,n,?,n,?,n,?,n,?,n,?,y,0
negative,74,F,f,f,f,f,t,f,f,f,f,f,f,y,0.30,y,1.20,y,93,y,0.83,y,112,n,?
negative,62,F,f,f,f,f,f,t,f,f,f,f,f,y,0,y,1.40,y,107,y,0.88,y,122,n,?
negative,35,F,f,f,f,f,f,f,f,f,f,f,f,y,0,y,2.30,y,88,y,1.04,y,84,n,?
negative,22,F,f,f,f,f,f,t,f,f,f,f,f,n,?,y,1.70,y,82,y,0.72,y,114,n,?
negative,28,F,f,f,f,f,f,t,t,f,f,f,f,y,0,y,4,y,187,y,1.50,y,124,n,?
negative,16,F,f,f,t,f,f,f,f,f,f,f,f,y,0,y,9.80,y,254,y,1.05,y,241,n,?
negative,64,F,f,f,f,f,t,f,f,f,f,f,f,y,0.90,n,?,y,115,y,0.97,y,119,n,?
negative,72,F,f,f,f,f,f,f,f,f,f,f,f,y,0.90,y,1.40,y,98,y,0.82,y,120,n,?
negative,74,M,f,f,f,f,f,f,f,f,f,f,f,y,0,y,1.40,y,113,y,0.84,y,136,n,?
negative,73,F,t,f,f,f,t,f,f,f,f,f,f,y,11,y,1.30,y,80,y,0.95,y,85,n,?
negative,?,?,f,f,f,f,f,f,f,f,f,f,f,y,1.50,y,1.70,y,75,y,0.84,y,89,n,?
negative,52,F,f,f,f,f,f,f,f,f,f,f,f,y,4.60,y,1.70,y,84,y,0.67,y,127,n,?
```

Dataset 2

1. Name: Fertility Dataset– Fertility Diagnosis
2. Source: UCI Machine Learning Repository

   https://archive.ics.uci.edu/ml/datasets/Fertility
3. Number of Instance: 100

4. Description: 100 volunteers provide a semen sample analyzed according to the WHO 2010 criteria. Sperm concentration are related to socio-demographic data, environmental factors, health status, and life habits
5. Missing Value: 0
6. Number of Attribute: 10
7. Attribute Information:
   (1). Season in which the analysis was performed.
   Winter: -1, Spring: -0.33, Summer: 0.33, Fall: 1
   (2). Age at the time of analysis.
   Between 18-36 years old: 1, others: 0
   (3). Childish diseases (i.e. chicken pox, measles, mumps, polio): No: 0, Yes: 1
   (4). Accident or serious trauma: No: 0, Yes: 1
   (5). Surgical intervention: No: 0, Yes: 1
   (6). High fevers in the last year
   Never: 1, More than three months ago: 0, Less than three months ago: -1
   (7). Frequency of alcohol consumption: 1) several times a day, 2) every day, 3) several times a week, 4) once a week, 5) hardly ever or never (0, 1)
   (8). Smoking habit: Never: -1, Occasional: 0, Daily: 1
   (9). Number of hours spent sitting per day: Continuous, 0~1
   (10). Output: Diagnosis: Normal: N, Altered: O
8. Dataset Preview

```
-0.33,0.69,0,1,1,0,0.8,0,0.88,N
-0.33,0.94,1,0,1,0,0.8,1,0.31,O
-0.33,0.5,1,0,0,0,1,-1,0.5,N
-0.33,0.75,0,1,1,0,1,-1,0.38,N
-0.33,0.67,1,1,0,0,0.8,-1,0.5,O
-0.33,0.67,1,0,1,0,0.8,0,0.5,N
-0.33,0.67,0,0,0,-1,0.8,-1,0.44,N
-0.33,1,1,1,1,0,0.6,-1,0.38,N
1,0.64,0,0,1,0,0.8,-1,0.25,N
1,0.61,1,0,0,0,1,-1,0.25,N
1,0.67,1,1,0,-1,0.8,0,0.31,N
1,0.78,1,1,1,0,0.6,0,0.13,N
1,0.75,1,1,1,0,0.8,1,0.25,N
1,0.81,1,0,0,0,1,-1,0.38,N
```

Part 2 – Visualizations

Dataset 1

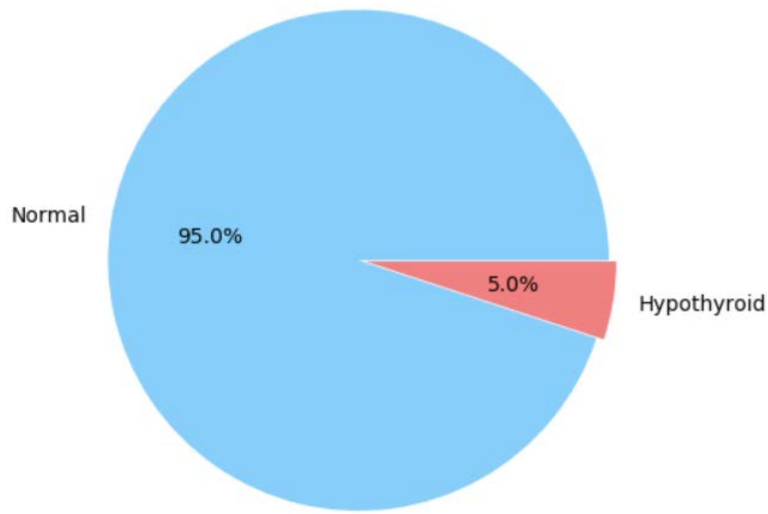

Figure 1-1: Percentage of people that is diagnosed as normal and Hypothyroid.

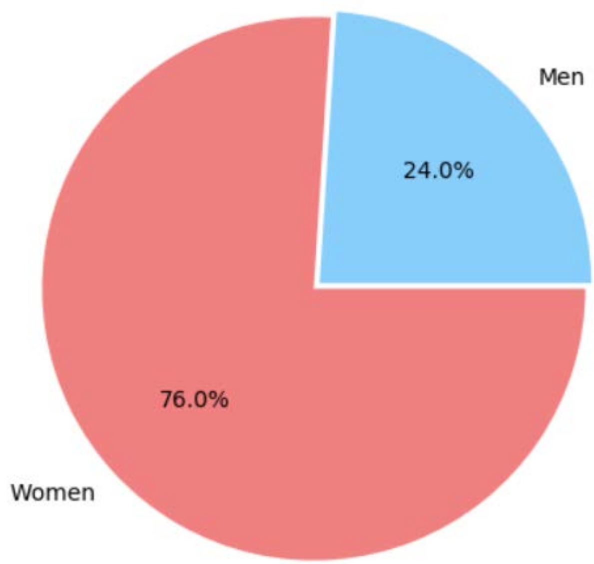

Figure 1-2: Percentage of different sex of all the ones having hypothyroid.

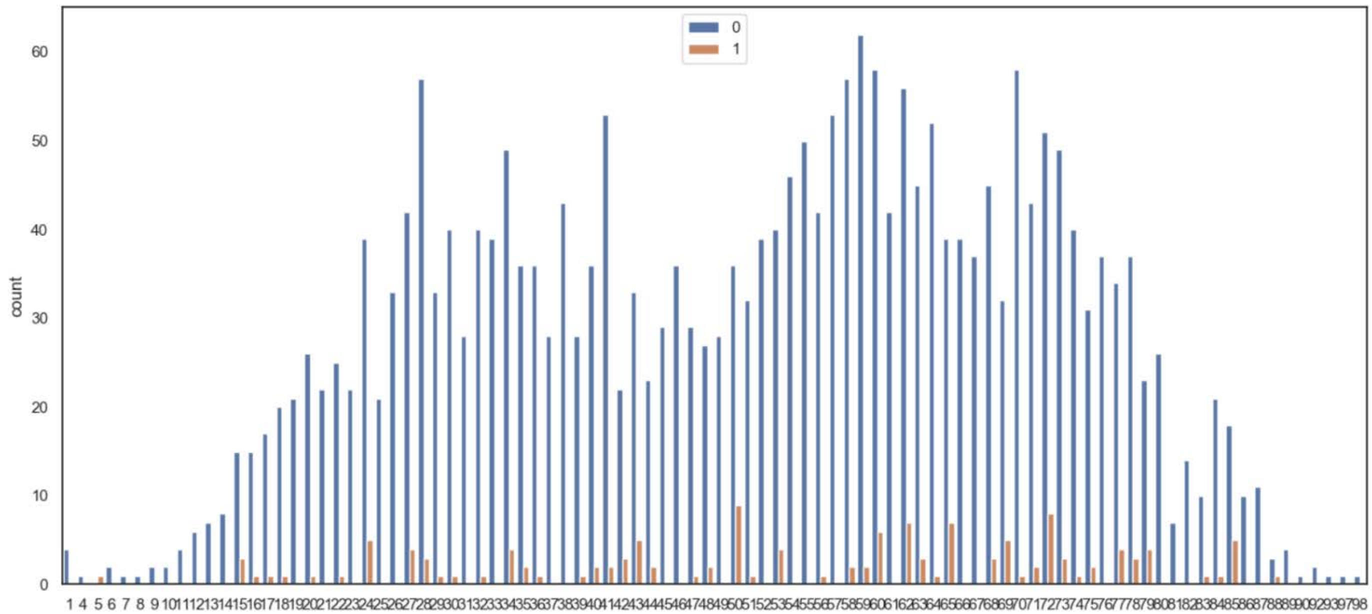

Figure 1-3: Bar chart showing the age distribution. 0 means normal and 1 means hypothyroid. X-axis are the age in years.

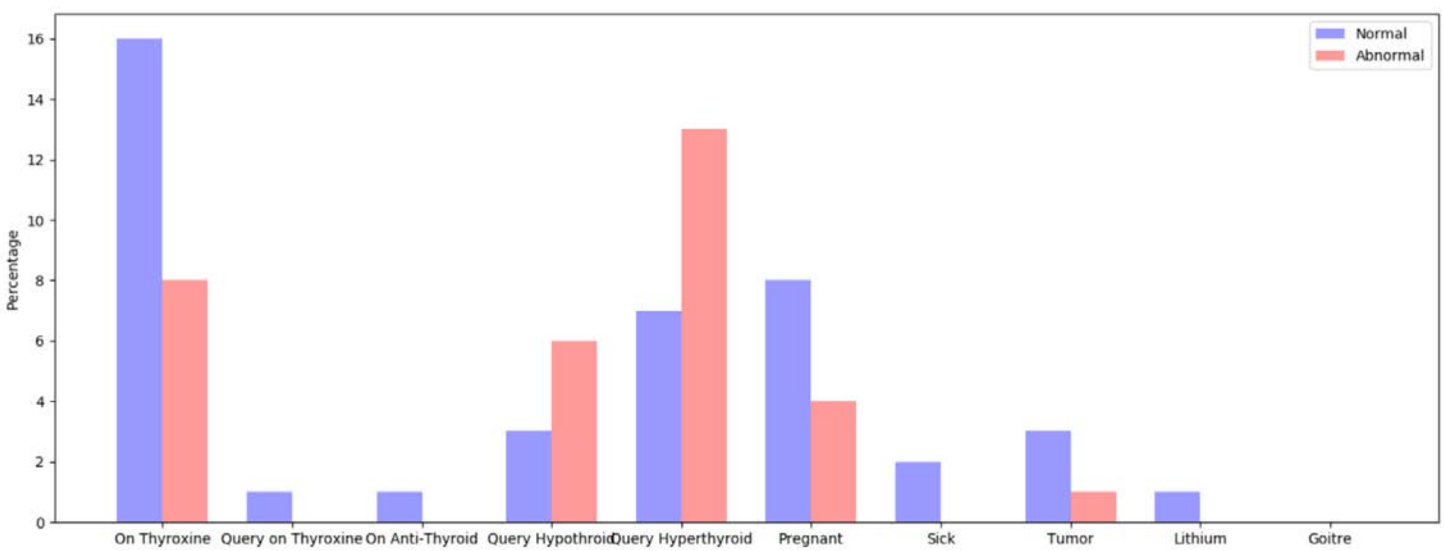

Figure 1-4: Bar chart showing the percentage of different condition and percentage of normal and hypothyroid.

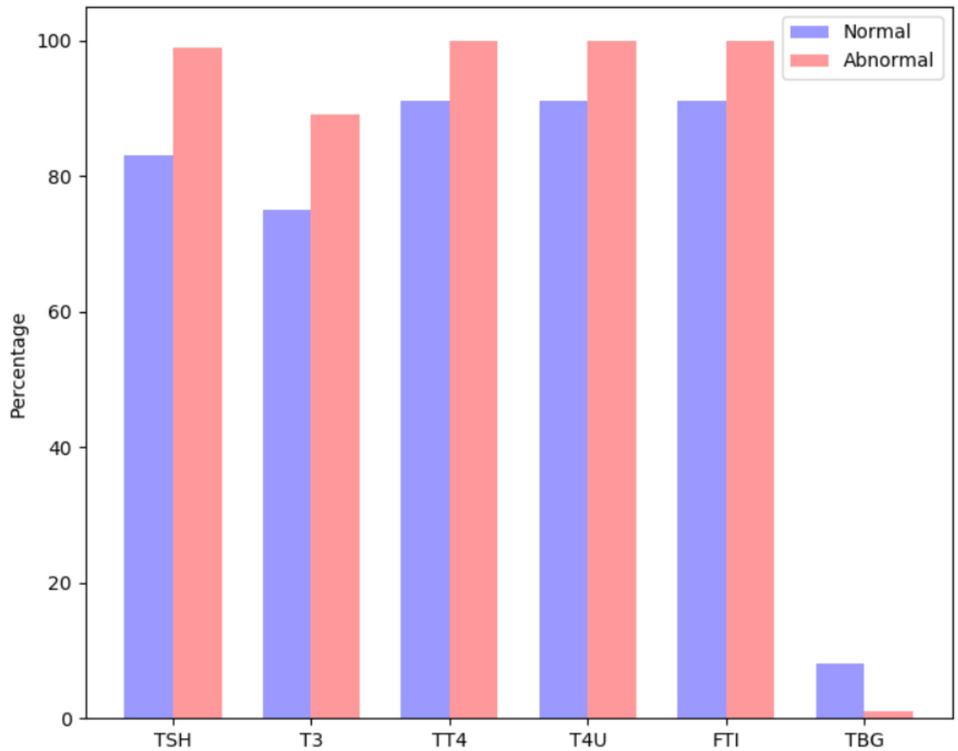

Figure 1-5: Bar chart showing the percentage of having each measure.

Dataset 2

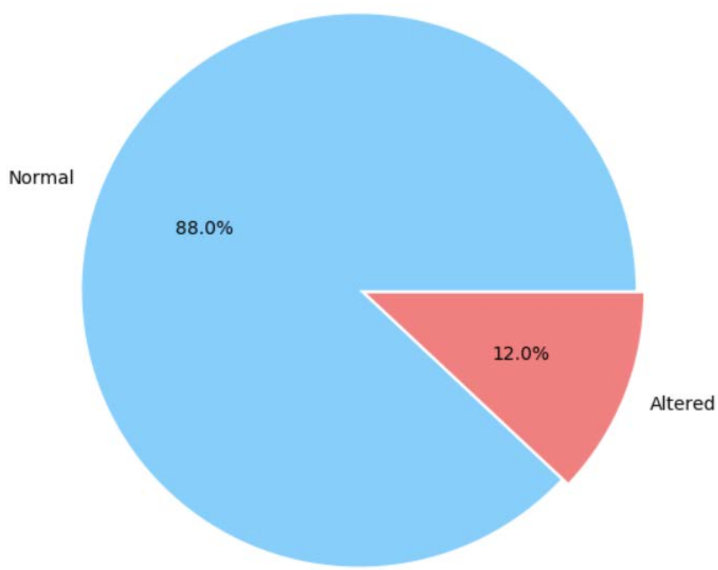

Figure 2-1: Percentage of people that is diagnosed as normal and altered.

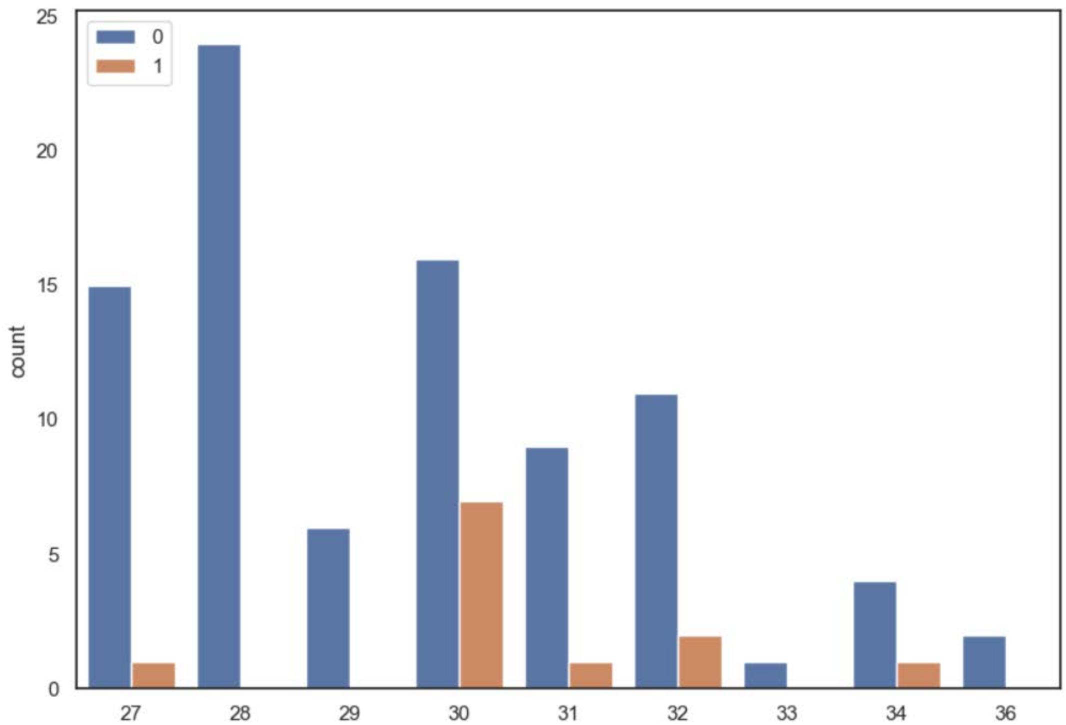

Figure 2-2: Bar chart showing the distribution of age, 0 means normal and 1 means abnormal. X-axis present the age in years

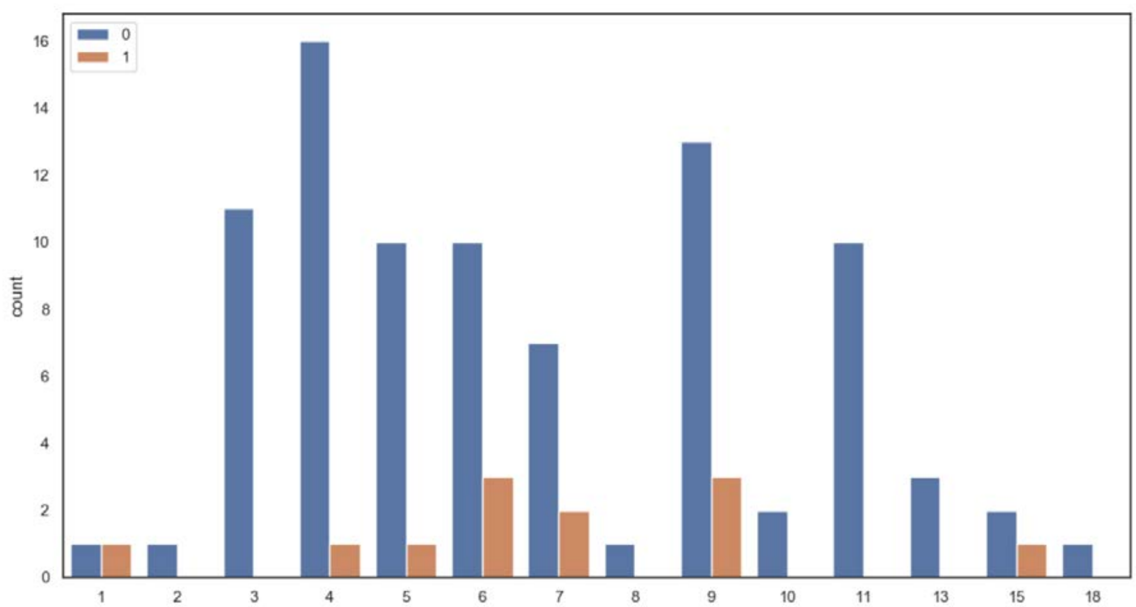

Figure 2-3: Bar chart showing the distribution of sitting hours, 0 means normal and 1 means abnormal. X-axis present sitting hours

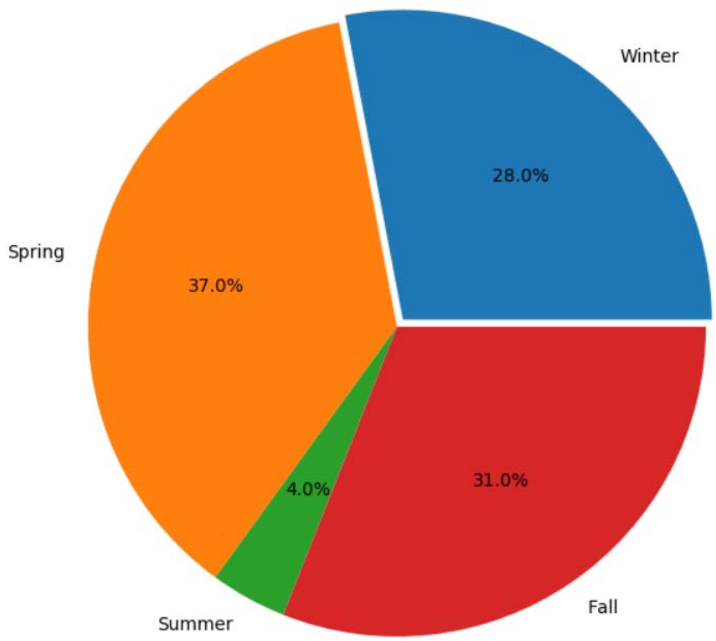

Figure 2-4: Pie charts showing the percentage of the season the sperm is collected

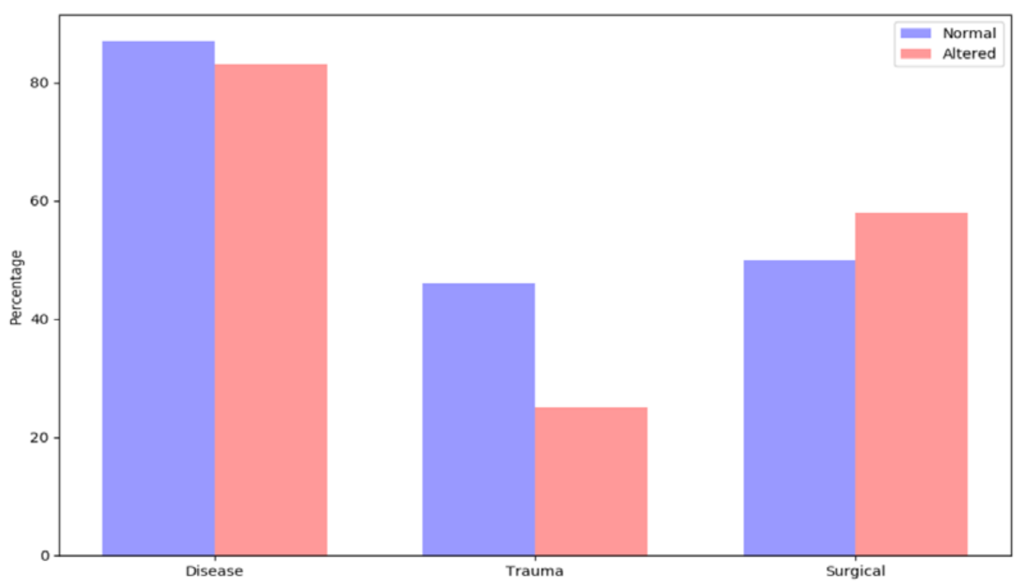

Figure 2-5: Bar chart showing the percentage of different condition and percentage of normal and altered.

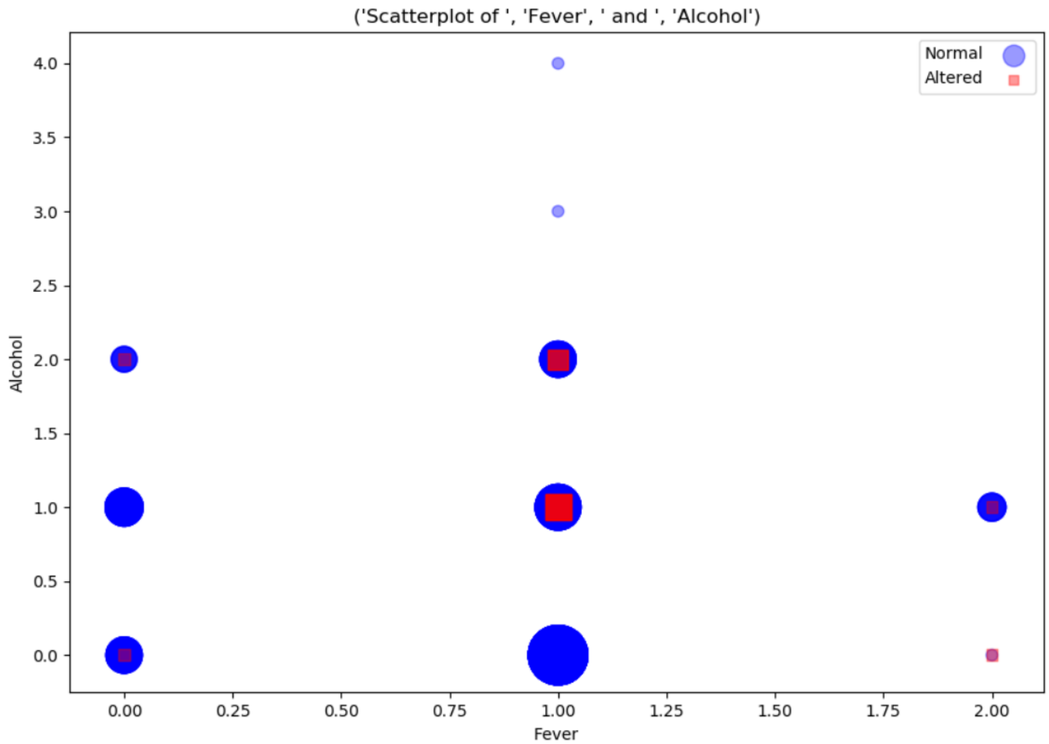

Figure 2-6: The scatter plot presents the relation between fever, alcohol and the condition of the semen.

Part 3 – Observations

Dataset 1

Figure 1-1

Percentage of people that is diagnosed as normal and Hypothyroid. Blue means normal and red means hypothyroid. We can see that most people do not have hypothyroid, and the percentage of people having is about 5 percent.

Figure 1-2

Percentage of sex of all the ones having hypothyroid. Blue means men and red means women. From this figure, we can see that more than 3/4 of people having hypothyroid are female, because of that, we can assume that there might be some reason leading to female having higher rate than male to have hypothyroid.

Figure 1-3

Bar chart showing the age distribution. 0 means normal and 1 means hypo-thyroid. X-axis are the age in years. From a broader view, the distribution of the normal people is similar to the Gaussian distribution, which have less data is both high and low end and have more data in the middle. On the other hand, the distribution of hypothyroid is more distributed, there are no significant special points that have more counts than other.

Figure 1-4

Bar chart showing the percentage of different condition and percentage of normal and hypothyroid. The x-axis shows different conditions, which are: On Thyroxine, Query on Thyroxine, On Anti-Thyroid, Query Hypothyroid, Query Hyperthyroid, Pregnant, Sick, Tumor, Lithium and Goiter. The y-axis presents the percentage of people under each condition. The blue charts represent normal data and the red chart represent Hypothyroid data.

From this figure, we can briefly tell the relation between the percentage of normal and abnormal under each condition. By observing this figure, I found out that there isn't a condition that has great effect on having hypothyroid or not due to the reason of the difference between the percentage of them are not too significant. The only two factors that may be considered are "On Thyroxine" and "Query Hyperthyroid". For the factor "On Thyroxine", the percentage of normal is almost twice as the percentage of abnormal, therefore, we can guess that people that are on thyroxine are more likely to not have hypothyroid. For the factor "Query Hyperthyroid", the percentage of having hyperthyroid is about 1.5 times of the percentage of normal, we can make a guess that people who have hyperthyroid are more likely to query hyperthyroid than people who don't have hyperthyroid.

Figure 1-5

Bar chart showing the percentage of having each measure. In this bar chart I found out that the percentage of normal and abnormal in each test measure are really close. For my assumption, this might mean that these measures are required for knowing if a person have hypothyroid or not, which is necessary measure in this dataset. As for the TBG measure, which is extremely less compare to other measures, I guess it may be a measure that is not common is thyroid related test, causing that both normal and abnormal have a low percentage of this measure.

Dataset 2

Figure 2-1

Percentage of people that is diagnosed as normal and altered.

Blue means normal and red means altered.

From this figure, we can tell that only 12% of people are have abnormal sperm, most of the people are having normal sperm.

Figure 2-2

Bar chart showing the distribution of age, 0 means normal and 1 means abnormal. X -axis present the age in years. In this I found out that in this dataset, there are more younger people than older people. Surprisingly, the number of men at age 28 is the highest and none of them have altered sperm. However, from an overall view, there seems to be no pattern of normal or altered sperm when comparing them with age.

Figure 2-3

Bar chart showing the distribution of sitting hours, 0 means normal and 1 means abnormal. X-axis present sitting hours. From the chart, we can see most people are having sitting hours around three to nine hours a day. Similar to the figure 1-3, the distribution of abnormal is pretty distributed, there are no significant special points that have more counts than other. It is also hard to find a pattern of normal or altered sperm when comparing them with sitting hours.

Figure 2-4

Pie chart showing the percentage of the season the sperm is collected. From the figure, we can say that Spring, Fall and Winter each present almost one third of the total data and Summer almost have not data. I guess that this might related to how these sperm are collected, for example, the sperm collection progress may start in fall and continued for a year, and most people who are willing to participate attends the progress early.

Figure 2-5

Bar chart showing the percentage of different condition and percentage of normal and altered. The x-axis shows different conditions, which are: Disease, Trauma and Surgical. The y-axis presents the percentage of people under each condition of each group. The blue charts represent normal data and the red chart represent altered data.

From this figure, we can see the relation between each factor and the percentage of having normal or abnormal sperm. By observing this bar chart, I assume that these factors don't have great effect in the normality of a men's sperm, the reason of that is because under each condition, the percentage of normal and abnormal are similar. Take disease as example, both normal and abnormal have high percentage of having disease, therefore we assume that disease don't have effect in the normality of a men's sperm.

Figure 2-6

The scatter plot presents the relation between fever, alcohol and the condition of the semen. The x-axis present if the person has fever recently. 0 means not having fever in the last year, 1 means having fever more than three months age and 2 means having fever less than three months ago. The y-axis presents the frequency of having alcohol, 0 means hardly ever or never have alcohol, the larger the number is the more frequent the person have alcohol, the largest number 4 means having alcohol several times a day. Color blue represent the normal one and red represent altered one. The larger the scatter plot is indicated higher number of people.

By observing this scatter plot, we can found out that people who have fever more than three months ago are having the highest rate of have abnormal sperm, but we cannot see any relation between the frequency of having alcohol, we can only see that people, including normal and abnormal, tends not to drink alcohol, the number of people decreased as the frequency of drinking alcohol raised. As for the percentage of people having fever, we can see that most people have fever more than three months age, and only have a few people have fever in the past three months. I guess that the main reason of not able to find any significant relation on this dataset is mainly because the number of dataset is too small, which is only a hundred data in this dataset, if we are able to gain more data and have it visualized, be might be able to find some more significant relations among each factors.

Part 4 – Appendix

For this project, I complete all the program in Python, the following part is the code that I used to produce these visualized charts.

Dataset 1

In first step of this part, I first process the raw data and organize it into an easier format for further visualization

```
hypo_file = open("data/hypothyroid.data", "r")
lines = hypo_file.readlines()
hypo_file.close()
hypo = []

# function for turning things into binary
def organize(temp):
    temp = [tem.replace('f', '0').replace('t', '1').replace('M', '1')
                .replace('F', '0').replace('y', '1').replace('n', '0')
            for tem in temp]
    return temp

# organize each
for i in lines:
    temp = (i.rstrip('\n').split(','))
    if (temp[1] != '?') and ( temp[2] != '?'):
        if temp[0] == 'hypothyroid':
            temp[0] = '1'
        else:
            temp[0] = '0'
        temp = organize(temp)
        hypo.append(temp)

# write into new data
new_file = open("data/HypothyroidProcessed.txt", "w")
for i in hypo:
    for j in i:
        new_file.write(str(j))
        new_file.write(" ")
    new_file.write('\n')
new_file.close()
```

After this, the data attribute will become like this:

[00] Result	0 false	1 true
[01] Age	age (int)	
[02] Sex	0 Female	1 Male
[03] on thyroxine	0 false	1 true
[04] query on thyroxine	0 false	1 true
[05] on Anti-thyroid	0 false	1 true
[06] Surgery	0 false	1 true
[07] Query Hypothyroid	0 false	1 true
[08] Query Hyperthyroid	0 false	1 true
[09] Pregnant	0 normal	1 altered
[10] Sick	0 false	1 true
[11] Tumor	0 false	1 true
[12] Lithium	0 false	1 true
[13] Goiter	0 false	1 true
[14] TSH	0 false	1 true
[15] Num of TSH	int or '?'	
[16] T3	0 false	1 true
[17] Num of T3	int or '?'	
[18] TT4	0 false	1 true
[19] Num of TT4	int or '?'	
[20] T40	0 false	1 true
[21] Num of T40	int or '?'	
[22] FTI	0 false	1 true
[23] Num of FTI	int or '?'	
[24] TBG	0 false	1 true
[25] Num of TBG	int or '?'	

Next, we will start to visualize the data

```
import matplotlib.pyplot as plt
import numpy as np
from collections import Counter
hypo_file = open("data/HypothyroidProcessed.txt", "r")
lines = hypo_file.readlines()
hypo_file.close()
hypo = []
nameSet = ['Result', 'Age', 'Sex', 'On Thyroxine', 'Query on Thyroxine', 'On Anti-Thyroid', 'Query Hypothroid',
           'Query Hyperthyroid', 'Pregnant', 'Sick', 'Tumor', 'Lithium', 'Goitre',
           'TSH Measure', 'TSH', 'T3 Measure', 'T3', 'TT4 Measure', 'TT4',
           'T40 Measure', 'T40', 'FTI Measure', 'FTI', 'TBG Measure', 'TBG']
# Remove space and '\n' and split by space, store data in hypo array
for i in lines:
    temp = i.rstrip('\n').split(' ')
    for x in range(14):
        temp[x] = int(temp[x])
    for x in range(14, 26, 2):
        temp[x] = int(temp[x])
        if temp[x]:
            temp[x+1] = float(temp[x+1])
    hypo.append(temp)

# Separate Normal data and Hypothyroid Data
normal_data = []
hypothyroid_data = []
for i in hypo:
    if i[0]:
        hypothyroid_data.append(i)
    else:
        normal_data.append(i)
```

```
# count the percentage of each item
def statistic (data, target):
    for i in data:
        for j in range(2, 14):
            if i[j]:
                target[j-2] += 1
        for j in range(14, 25, 2):
            if i[j]:
                target[j-2] += 1
    for i in range(len(target)):
        target[i] = (target[i]*100/len(data))
    return target
```

```
# Sum the number of disease, trauma and surgical
normal_statistic = [0]*26
hypothyroid_statistic = [0]*26

normal_statistic = statistic(normal_data, normal_statistic)
hypothyroid_statistic = statistic(hypothyroid_data, hypothyroid_statistic)
print normal_statistic
print hypothyroid_statistic
```

```
def bar_chart(dataset1, dataset2):
    fig, ax = plt.subplots()
# parameter setting
    index = np.arange(10)
    bar_width = 0.35
    opacity = 0.4
# input data
    ax.bar(index, dataset1, bar_width, alpha=opacity, color='b', label='Normal')
    ax.bar(index + bar_width, dataset2, bar_width, alpha=opacity, color='r', label='Abnormal')
# Label the axes
    ax.set_ylabel('Percentage')
    ax.set_xticks(index + bar_width / 2)
    ax.set_xticklabels(('On Thyroxine', 'Query on Thyroxine', 'On Anti-Thyroid', 'Query Hypothyroid',
            'Query Hyperthyroid', 'Pregnant', 'Sick', 'Tumor', 'Lithium', 'Goitre'))
    ax.legend()
# Show chart
    fig.tight_layout()
    plt.show()


def pie_chart(data, labels):
    colors = ['lightskyblue', 'lightcoral']
    explode = (0.03, 0)  # explode 1st slice
    # Plot
    plt.pie(data, explode=explode, labels=labels, autopct='%1.1f%%', colors=colors)
    plt.axis('equal')
    plt.show()
```

```
def count_chart(dataset):
    sns.set(style="white")
    ax = sns.countplot(x = 'age', hue = 'type', data=dataset)
    plt.show()


count_chart({'age': [i[1] for i in hypo], 'type': [i[0] for i in hypo]})
pie_chart([len(normal_data), len(hypothyroid_data)],['Normal', 'Hypothyroid'])
bar_chart(normal_statistic[1:11:], hypothyroid_statistic[1:11:], ('On Thyroxine', 'Query on Thyroxine',
         'On Anti-Thyroid', 'Query Hypothyroid', 'Query Hyperthyroid', 'Pregnant', 'Sick', 'Tumor', 'Lithium', 'Goitre'))
pie_chart([hypothyroid_statistic[0], len(hypothyroid_data)-hypothyroid_statistic[0]], ['Men', 'Women'])
bar_chart(normal_statistic[12:23:2], hypothyroid_statistic[12:23:2],('TSH', 'T3', 'TT4', 'T4U', 'FTI', 'TBG') )
```

Dataset 2

In first step of this part, I first process the raw data and organize it into an easier format for further visualization

```
# read the data in lines
fertile_file = open("data/fertility.txt", "r")
lines = fertile_file.readlines()
fertile_file.close()
temp=[]
fertile = []
# organize data in each lines
for i in lines:
    temp = (i.rstrip('\n').split(','))
    temp[0] = int(round((float(temp[0])+1)*1.5))
    temp[1] = int(float(temp[1])*18+18)
    temp[5] = abs(int(int(temp[5])+1)-2)
    temp[6] = abs(int(float(temp[6])*5)-5)
    temp[7] = int(temp[7])+1
    temp[8] = int(float(temp[8])*18)
    if (temp[9] == 'N'):
        temp[9] = 0
    else:
        temp[9] = 1
    fertile.append(temp)

# write the data into a new file
new_file = open("data/FertilityProcessed.txt", "w")
for i in fertile:
    for j in i:
        new_file.write(str(j))
        new_file.write(" ")
    new_file.write('\n')
new_file.close()
```

After this, the data attribute will become like this:

[0] Season	0 Winter,	1 Spring,	2 Summer,	3 Fall		
[1] Age	age (integer)					
[2] Disease	0 false	1 true				
[3] Trauma	0 false	1 true				
[4] Surgical	0 false	1 true				
[5] Fever	0 never	1 more than 3 months	2 less than 3 month			
[6] Alcohol	4 n/day	3 1/day	2 n/week	1 1/week	0 never	
[7] Smoke	0 never	1 sometime	2 daily			
[8] Sit Hour	hours(integer)					
[9] Result	0 normal	1 altered				

Next, we will start to visualize the data

```
import matplotlib.pyplot as plt
import numpy as np
from collections import Counter

# Read processed file
fertile_file = open("data/FertilityProcessed.txt", "r")
lines = fertile_file.readlines()
fertile_file.close()
fertile = []
nameSet = ['Season', 'Age', 'Disease', 'Trauma', 'Surgical', 'Fever', 'Alcohol', 'Smoke', 'Sitting Hour', 'Result']
# Remove space and '\n' and split by space, store data in fertile array
for i in lines:
    temp = i.rstrip('\n').split(' ')
    temp = map(int, temp)
    fertile.append(temp)

# Separate Normal and Altered data
normal_data = []
altered_data = []
for i in fertile:
    if i[9]:
        altered_data.append(i)
    else:
        normal_data.append(i)


# count the percentage of disease, trauma and surgical
def statistic (data, target):
    for i in data:
        for j in range(2, 5):
            if i[j]:
                target[j-2] += 1
    for i in range(len(target)):
        target[i] = (target[i]*100/len(data))
    return target
```

```
# Sum the number of disease, trauma and surgical
normal_statistic = [0, 0, 0]
altered_statistic = [0, 0, 0]

normal_statistic = statistic(normal_data, normal_statistic)
altered_statistic = statistic(altered_data, altered_statistic)


#  Function for getting every n-th item in fertile
def get_n_item(item, array=fertile):
    tmp = []
    for x in array:
        tmp.append(x[int(item)])
    return tmp
```

```python
def scatter_plot(a, b):
    x_data_n = get_n_item(a, normal_data)
    x_data_a = get_n_item(a, altered_data)
    y_data_n = get_n_item(b, normal_data)
    y_data_a = get_n_item(b, altered_data)
    x_label = nameSet[a]
    y_label = nameSet[b]
# Create the plot object
    fig, ax = plt.subplots()
# create a list of the sizes, here multiplied by 50 for scale
    c = Counter(zip(x_data_n, y_data_n))
    s = [50 * c[(xx, yy)] for xx, yy in zip(x_data_n, y_data_n)]
# s size, c color, marker type of dot, alpha transparency
    ax.scatter(x_data_n, y_data_n, s = s, c = 'blue', alpha = 0.4, marker='o', label = "Normal")
    c = Counter(zip(x_data_a, y_data_a))
    s = [50 * c[(xx, yy)] for xx, yy in zip(x_data_a, y_data_a)]
    ax.scatter(x_data_a, y_data_a, s = s, c = 'red', alpha = 0.4, marker='s', label = "Altered")
# Label the axes and provide a title
    title = 'Scatterplot of ',x_label,' and ',y_label
    ax.set_title(title)
    ax.set_xlabel(nameSet[a])
    ax.set_ylabel(nameSet[b])
    ax.legend(loc='best', markerscale=0.5, markerfirst=0)
# Show chart
    fig.tight_layout()
    plt.show()
```

```python
def bar_chart(dataset1, dataset2):
    fig, ax = plt.subplots()
# parameter setting
    index = np.arange(3)
    bar_width = 0.35
    opacity = 0.4
# input data
    ax.bar(index, dataset1, bar_width, alpha=opacity, color='b', label='Normal')
    ax.bar(index + bar_width, dataset2, bar_width, alpha=opacity, color='r', label='Altered')
# Label the axes
    ax.set_ylabel('Percentage')
    ax.set_xticks(index + bar_width / 2)
    ax.set_xticklabels(('Disease', 'Trauma', 'Surgical'))
    ax.legend()
# Show chart
    fig.tight_layout()
    plt.show()

def pie_chart(data, labels):
    colors = ['lightskyblue', 'lightcoral']
    explode = (0.03, 0)  # explode 1st slice
    # Plot
    plt.pie(data, explode=explode, labels=labels, autopct='%1.1f%%', colors=colors)
    plt.axis('equal')
    plt.show()
```

```
season_data = [0,0,0,0]
for i in fertile:
    if (i[0] == 0):
        season_data[0]+=1
    elif (i[0] ==1):
        season_data[1]+=1
    elif (i[0] ==2):
        season_data[2]+=1
    else:
        season_data[3]+=1

print season_data

def count_chart(dataset):
    sns.set(style="white")
    ax = sns.countplot(x = 'Hours',hue='Result', data=dataset)
    plt.show()


pie_chart([len(normal_data), len(altered_data)],['Normal', 'Altered'])
bar_chart(normal_statistic, altered_statistic)
scatter_plot(5,6)
count_chart({'Age': [i[1] for i in fertile], 'Result':[i[9] for i in fertile]})
pie_chart(season_data,['Winter', 'Spring', 'Summer','Fall'])
count_chart({'Hours':[i[8] for i in fertile], 'Result':[i[9] for i in fertile]})
```