Lab-1

February 15, 2022

Blaine Mason

MATH 216 Lab-1

Lab 1: Descriptive Statistics and Graphs

Purpose: The purpose of this lab assignment is to begin to understand what statistical inference is; display data about a sample of population and make conclusions about the population based on those samples. For example the correlation between gender and likelihood to smoke.

Introduction: The data for today's lab assignment has students in an introductory statistics course. Each student recorded his or her height, weight, gender, smoking preference, usual activity level, and resting pulse. Then they all flipped coins, and those whose coins came up heads ran in place for one minute. Then the entire class recorded their pulses once more.

We see that there will need to be some preprocessing since certain values in the table hold numerical values to represent labels. For instance, in the "Smokes" column; one represents "smokes regularly" and two represents "does not smoke regularly". To make visualizations such as the pie chart, we need to map those values from integers to their respective labels.

We perform analysis on the distribution of height and weight based on gender. We then group the smoking data with gender and format the results into a bar graph.

---

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [10, 5]
```

In order to get the data from a minitab format we needed to export the data from minitab as a .xlsx so pandas can read the table in.

[4]:
```
data = pd.read_excel("Pulse.xlsx",converters={'Gender':str})
```

We display the first five rows of the data

[5]:
```python
data.head()
```

[5]:

	Pulse1	Pulse2	Ran	Smokes	Gender	Height	Weight	Activity
0	64	88	1	2	Male	66.0	140	2
1	58	70	1	2	Male	72.0	145	2
2	62	76	1	1	Male	73.5	160	3
3	66	78	1	1	Male	73.0	190	1
4	64	80	1	2	Male	69.0	155	2

1 Descriptive Statistics

Below we will describe the height and weight of the students:

[36]:
```python
data[["Height", "Weight"]].describe().T
```

[36]:

	count	mean	std	min	25%	50%	75%	max
Height	92.0	68.717391	3.659291	61.0	66.0	69.0	72.0	75.0
Weight	92.0	145.152174	23.739398	95.0	125.0	145.0	155.5	215.0

From the table above we can see the mean Height and Weight are 68.72in and 145.15lbs respectively.

[9]:
```python
print(f'Average Height = {np.mean(data[["Height"]])[0]:.2f}')
print(f'Average Weight = {np.mean(data[["Weight"]])[0]:.2f}')
```

```
Average Height = 68.72
Average Weight = 145.15
```

Now with respect to gender:

[10]:
```python
male_data = data[data["Gender"] == "Male"]
female_data = data[data["Gender"] == "Female"]
```

[28]:
```python
print("======================================================================")
print("Male Height Data")
print(male_data[["Height"]].describe().T)
print("\nMale Weight Data")
print(male_data[["Weight"]].describe().T)
print("======================================================================")
print("Female Height Data")
print(female_data[["Height"]].describe().T)
print("\nFemale Weight Data")
print(female_data[["Weight"]].describe().T)
print("======================================================================")
```

```
======================================================================
Male Height Data
        count       mean       std   min   25%   50%   75%   max
Height   57.0  70.754386  2.582777  66.0  69.0  71.0  73.0  75.0
```

```
Male Weight Data
        count        mean        std    min    25%    50%    75%    max
Weight   57.0  158.263158  18.636108  123.0  145.0  155.0  170.0  215.0
======================================================================
Female Height Data
        count  mean       std   min   25%   50%   75%   max
Height   35.0  65.4  2.562599  61.0  63.0  65.5  68.0  70.0

Female Weight Data
        count   mean        std   min    25%    50%    75%    max
Weight   35.0  123.8  13.372052  95.0  115.5  122.0  130.5  150.0
======================================================================
```

[29]:

```
print('===================================')
print(f'Average Height of Male = {np.mean(male_data[["Height"]])[0]:.2f}')
print(f'Average Weight of Male = {np.mean(male_data[["Weight"]])[0]:.2f}')
print('===================================')
print(f'Average Height of Female = {np.mean(female_data[["Height"]])[0]:.2f}')
print(f'Average Weight of female = {np.mean(female_data[["Weight"]])[0]:.2f}')
print('===================================')
```

```
===================================
Average Height of Male = 70.75
Average Weight of Male = 158.26
===================================
Average Height of Female = 65.40
Average Weight of female = 123.80
===================================
```

Below we will display the 5 most frequent heights and weights with their counts

[31]:

```
print("Most frequent Height and number of students with that Height␣
 ↪respectively:")
data[["Height"]].value_counts()[:5]
```

```
Most frequent Height and number of students with that Height respectively:
```

[31]:

```
Height
68.0      10
69.0      10
72.0       8
66.0       8
73.0       7
dtype: int64
```

[32]:

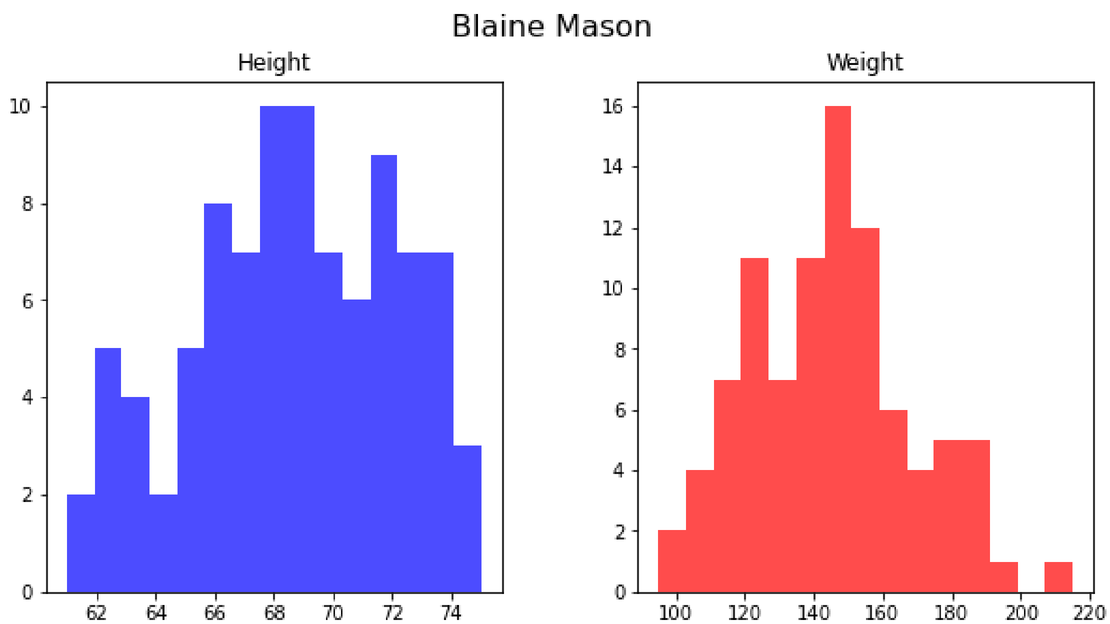
```
print("Most frequent Weight and number of students with that Weight␣
 ↪respectively:")
data[["Weight"]].value_counts()[:5]
```

```
Most frequent Weight and number of students with that Weight respectively:
```

[32]: Weight
```
150    10
155    10
145     5
130     5
125     5
dtype: int64
```

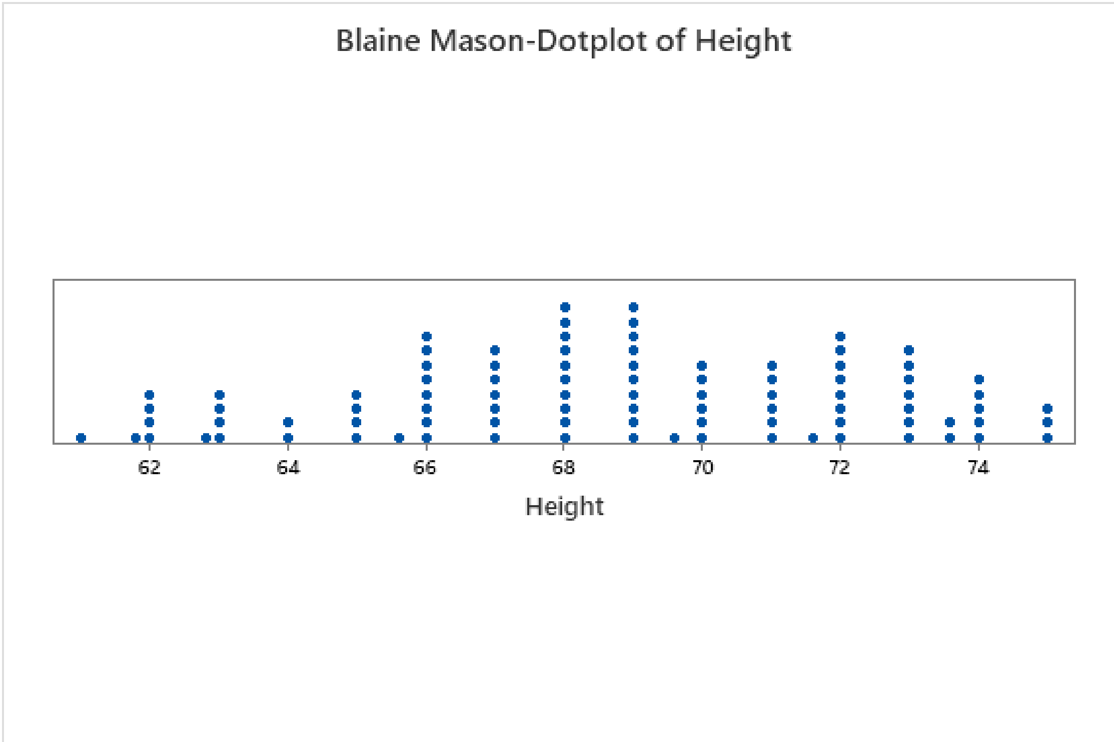
2 Quantitative Data Graphs

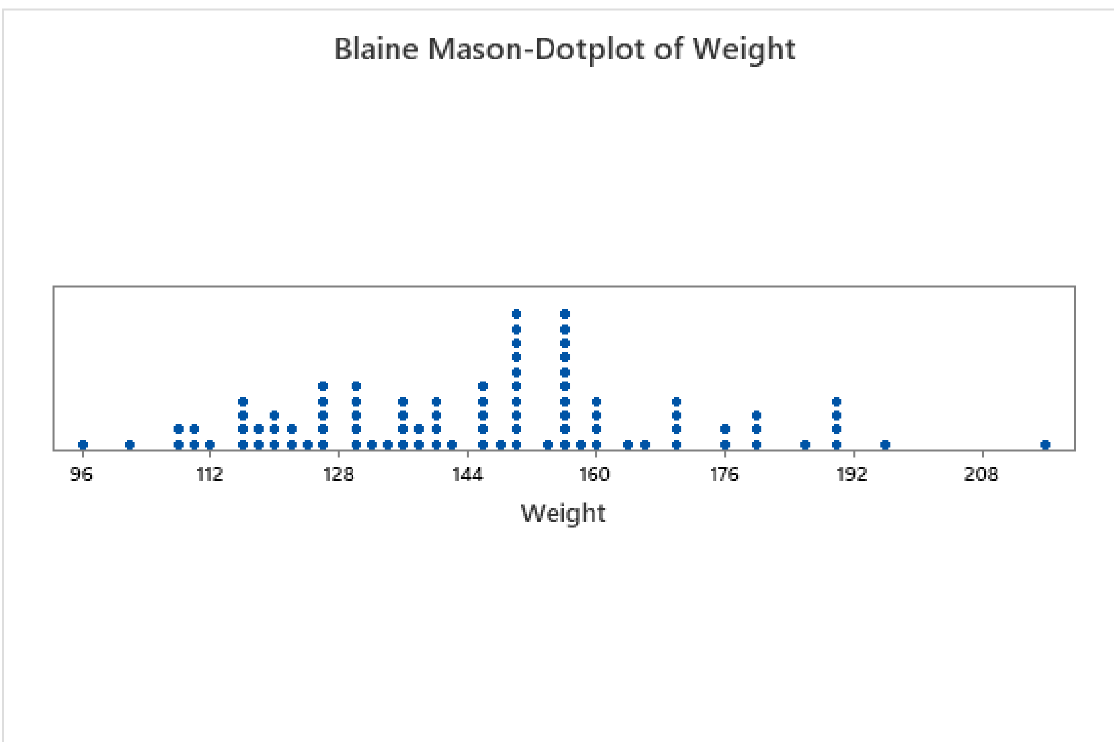
[42]:
```
height_weight = data[["Height", "Weight"]]
fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('Blaine Mason', fontsize=16)
height_weight[["Height"]].hist('Height', ax=ax1, bins=15, color="b", alpha=.7)
height_weight[["Weight"]].hist('Weight', ax=ax2, bins=15, color="r", alpha=.7)
ax1.grid(b=False)
ax2.grid(b=False)
plt.show()
```



Based on the notes we took in class, since we are working with more 92 samples we chose to use 15 bins for my histograms. This allows me to see similar results as the frequency tables, but if we were to use less bins we would not see the same results.

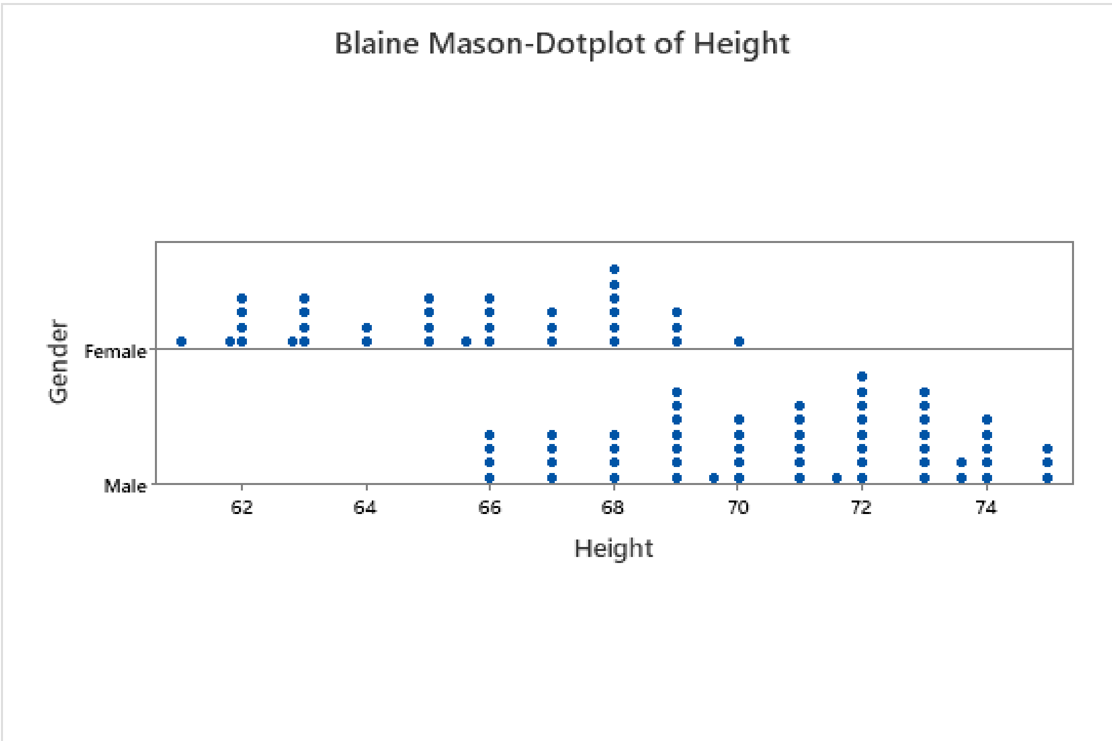
[39]:
```
from PIL import Image
display(Image.open('plots/Dotplot of Height.png'))
display(Image.open('plots/Dotplot of Weight.png'))
```

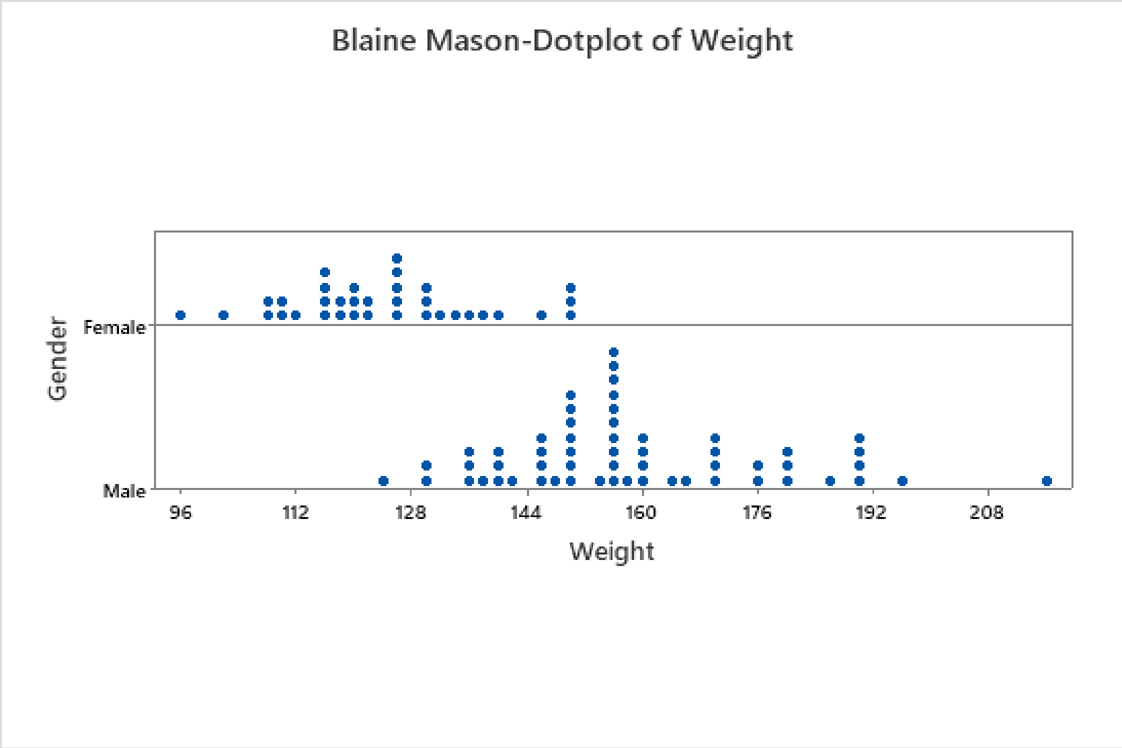

Blaine Mason-Dotplot of Height
62
64
66
68
70
72
74
Height


Blaine Mason-Dotplot of Weight
96
112
128
144
160
176
192
208
Weight

The dot plots do appear similar to the histograms above. This would be due to the fact that dot plots are displaying in a strictly categorical fashion. If one were to increase the bin size of the histograms there would be a convergence to mirror the dot plots.

[40]:
```
display(Image.open('plots/Dotplot of HeightG.png'))
display(Image.open('plots/Dotplot of WeightG.png'))
```



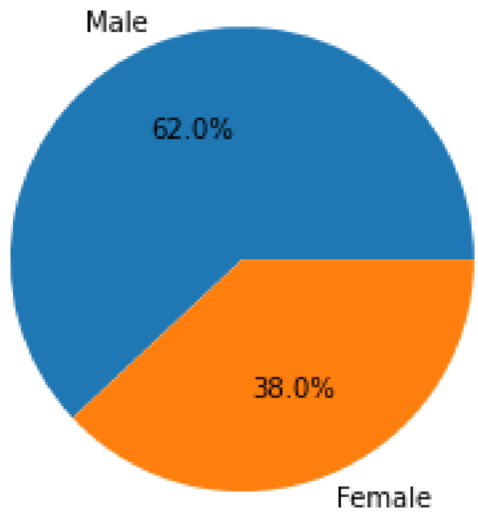


We know that height and weight is often found to be normally distributed. In these two dot plots we see that these both take a shape of a distribution that has data mostly around the mean and a small number in the tails. The difference I see between male and female is the mean for female height and weight is less than the mean for male height and weight.

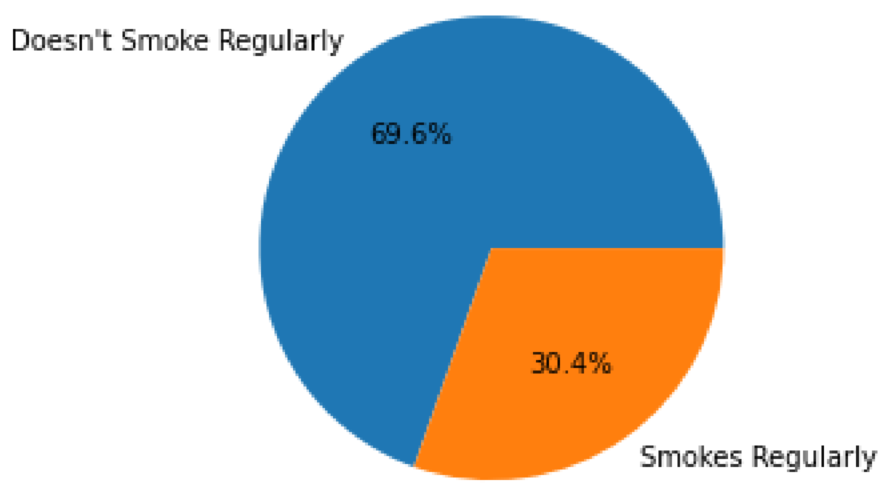
3 Qualitative Data graphs

We would claim that gender is nominal data since it is "named" data, but we cannot place it in any particluar order. Smoking habits is an example of ordinal data since there is a description of the habits, but can be placed in order. For example: "Smokes very little", "Smokes sometimes", "Smokes", "Smokes regularly", "Smokes very often".

[64]:
```
genders = data[["Gender"]].value_counts()
labels=["Male", "Female"]
plt.pie(genders,labels=labels, autopct='%1.1f%%');
```

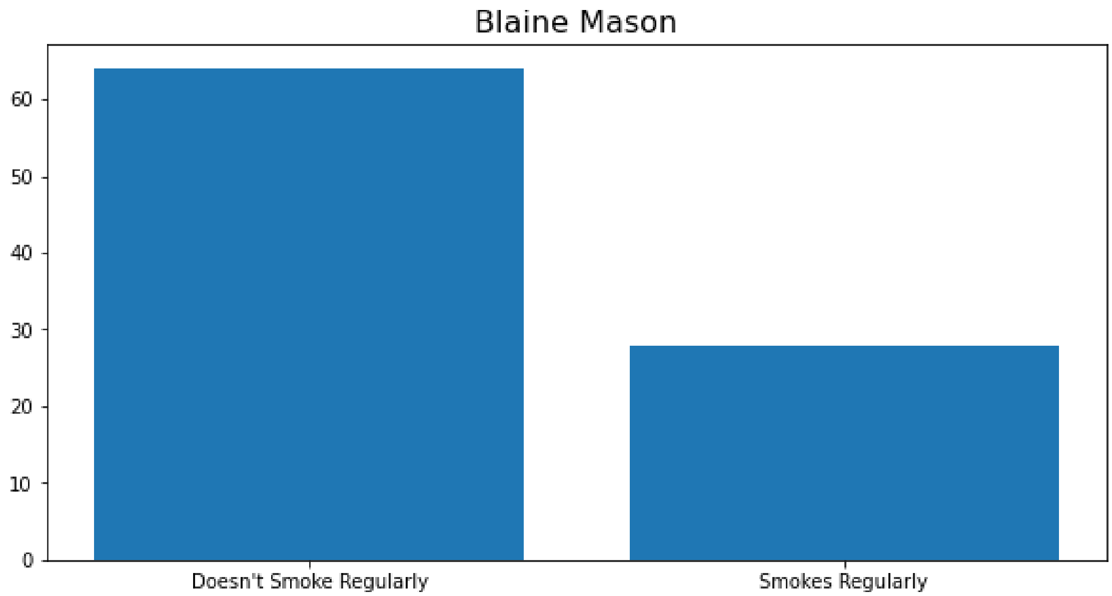



```
[70]: smokes = data[["Smokes"]].value_counts()
      labels=["Doesn't Smoke Regularly", "Smokes Regularly"]
      plt.pie(genders,labels=labels, autopct='%1.1f%%');
```



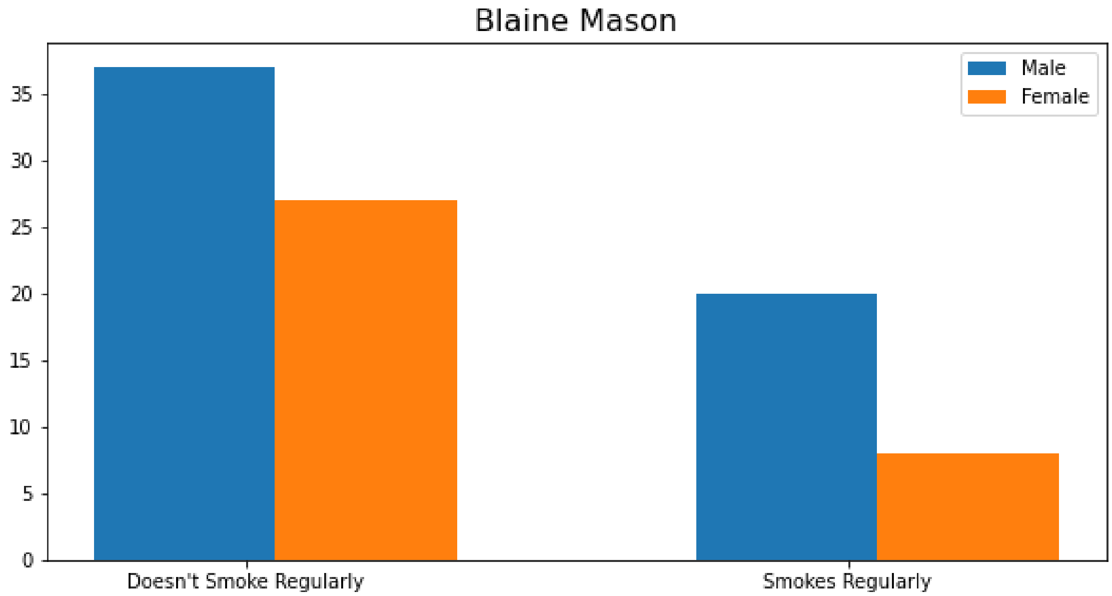

```
[155]: fig, ax = plt.subplots()
       plt.title('Blaine Mason', fontsize=16)
```

```
ax.bar(x=labels, height=[data["Smokes"].value_counts().iloc[0],data["Smokes"].
↪value_counts().iloc[1]]);
```



The bar graph without the y-axis would show proportions mainly, but we would argue both the pie chart and bar graphs without labels would describe percentage by showing proportions.

[153]:
```
X_axis = np.arange(len(labels))
plt.bar(X_axis, height=[male_data["Smokes"].value_counts().
↪iloc[0],male_data["Smokes"].value_counts().iloc[1]], width=0.3, label="Male")
plt.bar(X_axis + .3, height=[female_data["Smokes"].value_counts().
↪iloc[0],female_data["Smokes"].value_counts().iloc[1]], width=0.3,␣
↪label="Female")
plt.xticks(X_axis + .2/2, labels)
plt.legend()
plt.title("Blaine Mason", fontsize=16)
plt.show()
```



From the bar graph above we can see 20 Males smoke regularly and 27 Females do not smoke regularly. I am unsure we can make a conclusion since the number of females and makes differ greatly.