HOMEWORK 1

SETUP

The program was run and compiled in Ubuntu 18.04 but should work in most other versions. The following commands were done in order to get the program working:

```
> sudo apt-get update
> sudo apt-get install freeglut3
> sudo apt-get install freeglut3-dev
> sudo apt-get install binutils-gold
> sudo apt-get install g++ cmake
> sudo apt-get install libglew-dev
> sudo apt-get install g++
> sudo apt-get install mesa-common-dev
> sudo apt-get install build-essential
> sudo apt-get install libglew1.5-dev libglm-dev
> sudo apt-get install mesa-utils
> sudo apt-get –f install
```

PROGRAM COMPILATION AND RUNNING

The following command should be executed to compile the program:

```
> g++ histogram.cc -lm -lglut -lGL -lGLU -o histogram
```

This should create a "histogram" program which is ran doing the following command:

```
> ./histogram
```

SOLUTIONS/SCREENSHOTS

The public dataset was taken from http://archive.ics.uci.edu/ml/machine-learning-databases/poker/poker-hand-testing.data which is a dataset of thousands of different poker hands. Each card is represented in a hand by a number where the values are as follows:

SUITS:
1 = hearts, 2 = spades, 3 = diamonds, 4 = clubs
VALUES:
1 = ace, 2-10 = face value, 11 = jack, 12 = queen, 13 = king

To determine the final value of a card (of which there are 52 unique cards) we use the algorithm: ((SUITS – 1) * 13) + VALUES. This gives us a card value from 1 to 52 which we then plot on the histogram for all card values in the dataset. As seen in the screenshot, the y-axis is the probability of the card and the x-axis is the poker hand calculation as already mentioned. The orange bars are these values which make sense since the chance of drawing a card from a deck of 52 is equal for each card. The mean (shown at the top of the graph) is shown as approximately 26 which makes sense for an even distribution, being in the middle of the graph. The variance is about 225 which gives a standard deviation of about 15

which also makes sense since our data is spread very wide. The black lines are the univariate gaussian distribution with the corresponding mean and variance which we can see is centered around the mean and have a large variance (slowly declines on each side). Finally, the Bhattacharyya distance is calculated between the histogram values and the univariate gaussian distribution. But because we are taking the Bhattacharyya distance between a dataset and its own univariate gaussian distribution we expect the distance to be very small (near 0). It is indeed small at 0.065115, which some of it can be caused by floating point error. Since the distance is near zero we can say that there is a lot of overlap between the two distributions which makes sense since one is derived from the other.