Project Report
EE 381, Spring 2023

First Name:Samnang_ Last Name: __Lath

Student ID: ___028349722___

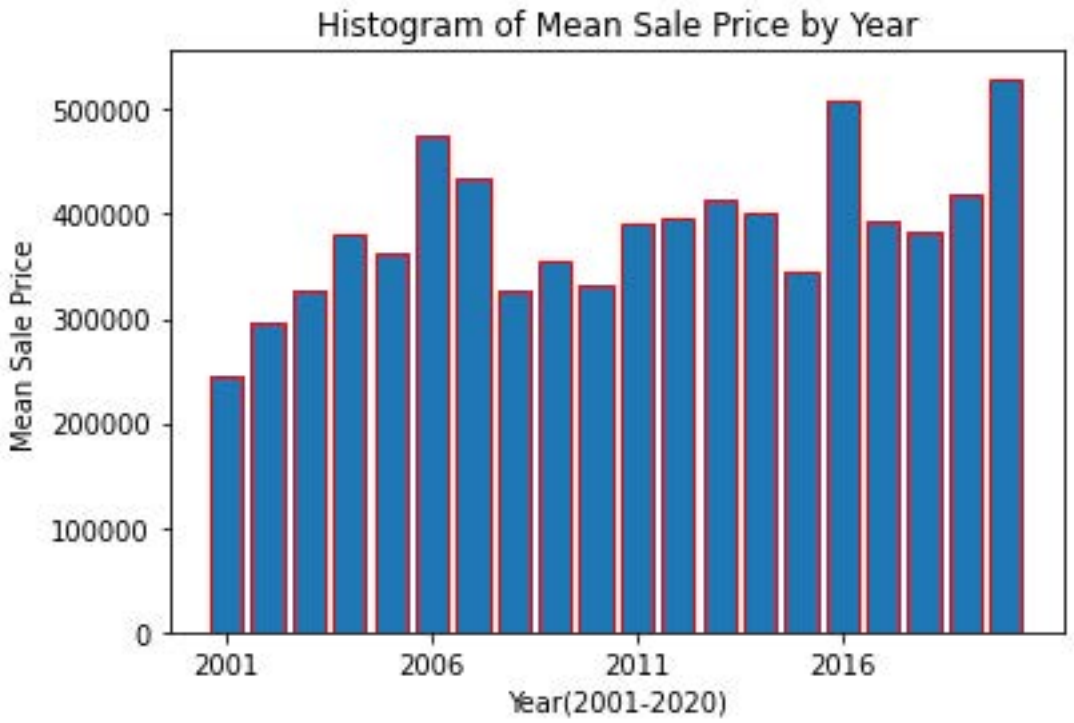
I did this final project own my own and did not share with anyone via discord, emails, verbal, or any other means, if I do, I understand that it is considered as a cheating, and there will be an academic disciplinary action on my academic dishonesty, it will stay forever on my record.

Sign _SL___ Date __05/03/2023_____

Mean Price: Fill the table below. (2 point)

Year	**Mean Price**
2001	246235.0352
2005	364030.1261
2010	331657.4726
2015	345883.7639
2020	529887.7341

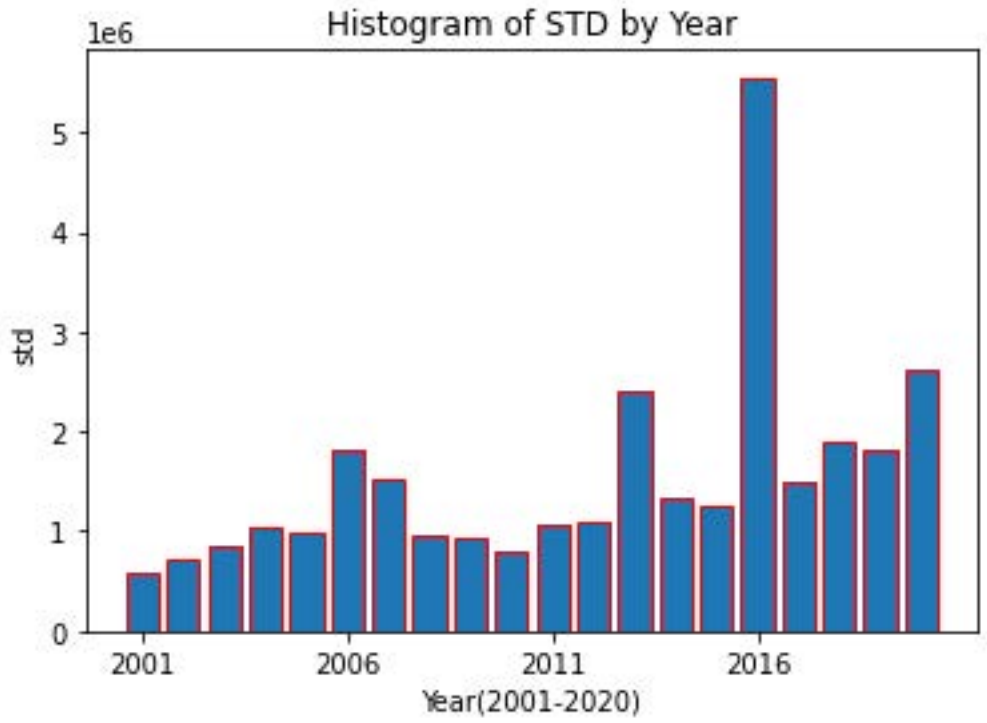
Mean Price: Insert a bar graph below showing yearly mean prices. (2 points)



Standard deviation (STD): Fill the table below. (2 point)

Year	**STD**
2001	587961.3711
2005	978403.1256
2010	790797.6331
2015	1242075.0601
2020	2621786.6525

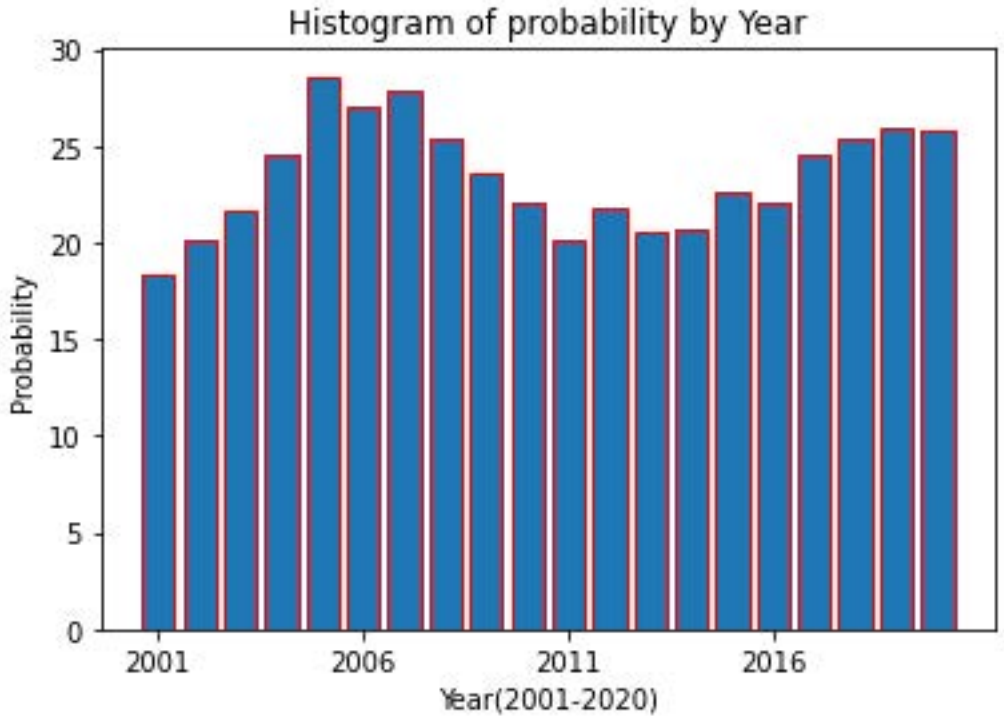
Standard deviation: Insert a bar graph below showing yearly standard deviations. (2 point)



Probability of price ranging from $200,000 to $300,000 inclusive: Fill the table below. (4 points)

Year	**Probability**
2001	18.31 %
2005	28.6 %
2010	22.02 %
2015	22.66 %
2020	25.8 %

Insert a bar graph below showing yearly probability of price ranging from $200,000 to $300,000 inclusive. (4 point)



Python Code: Provide your code below. (4 points)

```
import numpy as np
import matplotlib.pyplot as plt

######  Example 1.11 ##########
# load text file
fname = 'Sales_01_20.csv' # comma separated values
data1 = np.loadtxt(fname, delimiter=',', skiprows=1)
splityear = np.unique(data1[:, 0])
print(splityear)
means=[]

# calculate mean for each splityear
for year in splityear:
   boolarr = np.where(data1[:, 0] == year, True, False)
   yearsale = data1[boolarr, 1]
   mean_value= np.mean(yearsale)
   roundedmeans=round(mean_value,4)
   means.append (roundedmeans)#append in the list to use it later for matplotlib
   print ("In year ", year, " the mean is: ",roundedmeans)
std =[]
print("-------------------------------------------")
for year in splityear:
   boolarr = np.where(data1[:, 0] == year, True, False) #check for true arr by comparing year in data with year
   yearsale = data1[boolarr, 1]
   std_value=np.std(yearsale)
   roundedstd=round(std_value,4)
   std.append(roundedstd)
   print ("In year ", year, " std is: ", roundedstd)
print("-------------------------------------------")
probability=[]
for year in splityear:
   boolarryear = np.where(data1[:, 0] == year, True, False)
   boolarr2030 = np.where((data1[:, 1] >= 200000) & (data1[:, 1] <= 300000), True, False)
   num_sales_in_range = np.sum(np.logical_and(boolarryear, boolarr2030))
   total_sales_for_year = np.sum(boolarryear)
   yearlyprob = (num_sales_in_range / total_sales_for_year)*100
   rounded = round(yearlyprob, 2)
   probability.append(rounded)
   print("probability of ", year, "is", rounded,"%")


fig, ax = plt.subplots()   # Create a figure containing a single axes
ax.bar(splityear, means, edgecolor='red')
ax.set_xlabel('Year(2001-2020)')
ax.set_ylabel('Mean Sale Price')
ax.set_title('Histogram of Mean Sale Price by Year')
ax.set_xticks(np.arange(min(splityear), max(splityear), 5))
plt.show()
fig, ax = plt.subplots()
ax.bar(splityear, std, width=0.8, edgecolor='red')
ax.set_xlabel('Year(2001-2020)')
ax.set_ylabel('std')
ax.set_title('Histogram of STD by Year')
ax.set_xticks(np.arange(min(splityear), max(splityear), 5))
plt.show()
fig, ax = plt.subplots()
ax.bar(splityear, probability, width=0.8, edgecolor='red')
ax.set_xlabel('Year(2001-2020)')
ax.set_ylabel('Probability')
ax.set_title('Histogram of probability by Year')
ax.set_xticks(np.arange(min(splityear), max(splityear), 5))
plt.show()
```