Binomial Distribution

Problem statement

80% of all the visitors to Lavista Museum end up buying souvenirs from the souvenir shop at the Museum. On the coming Sunday, if a random sample of 10 visitors is picked:

1. Find the probability that every visitor will end up buying from the souvenir shop
2. Find the probability that a maximum of 7 visitors will buy souvenirs from the souvenir shop

Let's check first whether we satisfy the assumptions of the binomial distribution.

• There are only two possible outcomes (success or failure) for each trial. A visitor will buy souvenirs from the souvenir shop or not (yes or no).

• Number of trials (n) is fixed - There are 10 visitors in the sample.

• Each trial is independent of the other trials - It is reasonable to assume that the buying activity of visitors is independent.

• The probability of success (p) is the same for each trial - The probability of success for each visitor is 0.8.

```
from scipy.stats import binom
n=10
p=0.8
prob_every_visitor=binom.pmf(n,n,p)
print(prob_every_visitor)
```

```
0.10737418240000006
```

```
prob_max_7=binom.cdf(7,n,p)
print(prob_max_7)
```

```
0.32220047359999987
```

Problem statement

IT industry records the amount of time a software engineer needs to fix a bug in the initial phase of software development in 'debugging.csv'.

Let X = Time needed to fix bugs

X is a continuous random variable. Let's see the distribution of X and answer the below questions.

1. Find the probability that a randomly selected software debugging requires less than three hours
2. Find the probability that a randomly selected software debugging requires more than two hours
3. Find the 50th percentile of the software debugging time

Reading the Data into the Dataframe

```
import pandas as pd
df=pd.read_csv("debugging.csv")
time_taken=df['Time Taken to fix the bug']
prob_lessthan_3=(time_taken<3).mean()
print(prob_lessthan_3)
```

```
0.49761677788369874
```

```
prob_morethan_2=(time_taken>2).mean()
print(prob_morethan_2)
```

```
0.7521448999046711
```

```
percentile=time_taken.median()
print(percentile)
```

```
3.005
```

Problem statement

A testing agency wants to analyze the complexity of SAT Exam 2020. They have collected the SAT scores of 1000 students in "sat_score.csv". Let's answer some of the questions that will help to decide the complexity of SAT exam 2020.

1. Calculate the probability that a student will score less than 800 in SAT exam
2. Calculate the probability that a student will score more than 1300 in SAT exam
3. Calculate the minimum marks a student must score in order to secure 90th percentile
4. Calculate the minimum marks a student must score in order to be in the top 5%

```
sat_score= pd.read_csv("sat_score.csv")
sat_score.head()
```

	student_id	score
0	1	1018
1	2	1218
2	3	611
3	4	723
4	5	541

```
score= sat_score['score']
prob_less_than_800=(score<800).mean()
print(prob_less_than_800)
```

```
0.157
```

```
prob_more_than_1300=(score>1300).mean()
print(prob_more_than_1300)
```

```
0.068
```

```
score_90th_percentile=score.quantile(0.90)
print(score_90th_percentile)
```

```
1269.0
```

```
score_top_5=score.quantile(0.95)
print(score_top_5)
```

```
1338.1
```