# The Question

> A scientist is carrying out a series of experiments. Each experiment can end with either a success or a failure. The probability of success is **p = 0.820**, and the probability of failure is **q = 0.180**. Experiments in a series are independent of one another.
>
> If an experiment ends with a success, the detector registers its results correctly with a probability of **pr = 0.960**. If the experiment ends with a failure, nothing is registered.
>
> The scientist is going to run a series of **20 experiments**. Calculate the probability of getting **exactly 16** experiment results registered correctly on the detector. Round your answer to the nearest thousandth (three decimal places).

# Solution

This problem is a classic application of the **Binomial Distribution**, but with an initial step to determine the true probability of our event of interest.

1. **Identify the Event of Interest:** The question asks for the probability of a result being "registered". A result is registered *only if* the experiment is a success AND the detector registers it.
2. **Calculate the Probability of the Event:** We need to find the probability of both of these things happening in a single experiment. Since they are sequential, we multiply their probabilities:
   - P(Success) = `p = 0.820`
   - P(Registered | Success) = `pr = 0.960`
   - P(Registered Success) = P(Success) * P(Registered | Success)
   - P(Registered Success) = `0.820 * 0.960 = 0.7872`

   So, for any single experiment, the probability of its result being registered is **0.7872**. Let's call this `p_final`.
3. **Apply the Binomial Distribution Formula:** We are now looking for the probability of getting exactly `k` successes in `n` trials.
   - The formula is: `P(X=k) = C(n, k) * (p_final^k) * ((1 - p_final)^(n-k))`
   - Where:
     - `n` = total number of experiments = **20**
     - `k` = number of registered results we want = **16**

    - `p_final` = the probability of a registered result = **0.7872**
    - `C(n, k)` is the combination "n choose k", calculated as `n! / (k! * (n-k)!)`
4. **Plug in the numbers:**
    - `C(20, 16)` = `20! / (16! * 4!)` = `(20 * 19 * 18 * 17) / (4 * 3 * 2 * 1)` = **4845**
    - `(p_final^k)` = `0.7872^16`
    - `((1 - p_final)^(n-k))` = `(1 - 0.7872)^(20-16)` = `0.2128^4`
5. **Calculate the final probability:**
    - `P(X=16) = 4845 * (0.7872^16) * (0.2128^4)`
    - `P(X=16) ≈ 4845 * (0.043334) * (0.002052)`
    - `P(X=16) ≈ 0.215835`
6. **Round to the nearest thousandth:**
    - The final answer is **0.216**.

---

This problem is a perfect gateway to mastering the **Binomial Distribution**. Here is your Knowledge Skeleton for this topic.

# Part 1: The Core Concept (Theoretical Foundations)

The **Binomial Distribution** is a fundamental discrete probability distribution. It models the number of "successes" in a fixed number of independent trials, where each trial has the same probability of success.

- **What is it?** Think of it as a mathematical formula that answers the question: "If I flip a weighted coin `n` times, what's the probability I get exactly `k` heads?" It's used for scenarios with two possible outcomes (success/failure, yes/no, click/no-click).
- **Why does it matter?** It's the foundation for modeling binary outcomes in data science. It's used in A/B testing (e.g., comparing conversion rates), quality control (e.g., number of defective items), and understanding any process that consists of repeated, independent binary trials.
- **Underlying Assumptions (Bernoulli Trials):** For the Binomial Distribution to be applicable, the process must satisfy three conditions known as Bernoulli trials:

i. There are only two possible outcomes for each trial (e.g., success or failure).
ii. The number of trials, `n`, is fixed.
iii. Each trial is independent, and the probability of success, `p`, is the same for every trial.

# Part 2: The Interview Gauntlet (Theoretical Questions)

## Conceptual Understanding:

1. What is the Binomial Distribution and what does it describe?
2. What are the two key parameters that define a Binomial Distribution? (*Answer:* `n` *- the number of trials, and* `p` *- the probability of success on a single trial.*)
3. What is a Bernoulli trial, and how does it relate to the Binomial Distribution? (*Answer: A Bernoulli trial is a single experiment with two outcomes. The Binomial Distribution models the outcome of* `n` *independent Bernoulli trials.*)

## Intuition & Trade-offs:

4. Can you give a business example where you might use the Binomial Distribution? (*Example: A company sends a marketing email to 10,000 customers. Based on a historical open rate of 20%, what is the probability that exactly 2,050 people open the email?*)
5. How does the shape of the Binomial Distribution change as the probability `p` approaches 0.5? What about when `n` (the number of trials) gets very large? (*Answer: As* `p` *approaches 0.5, the distribution becomes more symmetric. As* `n` *gets very large, the shape of the Binomial Distribution can be approximated by the Normal Distribution.*)
6. What is the difference between the Binomial Distribution and the Poisson Distribution? When would you use one over the other? (*Answer: Binomial models the number of successes in a fixed number of trials. Poisson models the number of events occurring in a fixed interval of time or space. You use Poisson when* `n` *is very large and* `p` *is very small, or when you know the average rate of events but not the number of trials.*)

### Troubleshooting & Edge Cases:

7. In the original problem, what if the success of one experiment made the detector more likely to register the next success? Which assumption of the Binomial Distribution would be violated? (*Answer: The assumption of independent trials.*)
8. You are modeling website conversions. You find that the conversion probability is higher in the evening than in the morning. Why can't you directly apply a single Binomial Distribution to the entire day's traffic? (*Answer: The probability of success `p` is not constant across all trials, violating a core assumption. You might need to segment the data and model each time block separately.*)

## Part 3: The Practical Application (Code & Implementation)

In Python, the `scipy.stats` library is the standard tool for working with statistical distributions like the Binomial. You don't need to manually calculate combinations or powers.

The key object is `scipy.stats.binom`.

- **`binom.pmf(k, n, p)`**: Calculates the **Probability Mass Function (PMF)**. This answers the question: "What is the probability of getting *exactly* `k` successes?" This is what we used to solve the problem above.
- **`binom.cdf(k, n, p)`**: Calculates the **Cumulative Distribution Function (CDF)**. This answers the question: "What is the probability of getting `k` successes *or fewer*?"
- **`binom.sf(k, n, p)`**: Calculates the **Survival Function** (1 - CDF). This answers: "What is the probability of getting *more than* `k` successes?"
- **`binom.rvs(n, p, size=N)`**: Generates **Random Variates**. This simulates running the experiment `N` times. For example, `size=100` would give you an array of 100 numbers, where each number represents the count of successes from a set of `n` trials.

## Part 4: The Code Challenge (Practical Questions)

**Scenario:** A company runs an advertising campaign. On any given day, a person who sees the ad has a **5% chance (`p = 0.05`)** of clicking it. The ad is shown to **100 people (`n = 100`)** today.

**Your Tasks:**

1. Write Python code to calculate the probability that **exactly 7 people** click the ad.
2. Write Python code to calculate the probability that **10 or fewer people** click the ad.
3. Write Python code to calculate the probability that **more than 4 people** click the ad.

**Answer & Explanation:**

```
# Import the necessary library
from scipy.stats import binom

# --- Define the parameters of our distribution ---
n = 100  # Number of trials (people shown the ad)
p = 0.05 # Probability of success (a single person clicking)

# --- Task 1: Probability of EXACTLY 7 clicks ---
# We use the Probability Mass Function (pmf) for this.
k_exact = 7
prob_exact_7 = binom.pmf(k=k_exact, n=n, p=p)
print(f"The probability of exactly {k_exact} clicks is: {prob_exact_7:.4f}")
# Expected output: The probability of exactly 7 clicks is: 0.1060

# --- Task 2: Probability of 10 OR FEWER clicks ---
# We use the Cumulative Distribution Function (cdf) for this.
k_le_10 = 10
prob_le_10 = binom.cdf(k=k_le_10, n=n, p=p)
print(f"The probability of {k_le_10} or fewer clicks is: {prob_le_10:.4f}")
# Expected output: The probability of 10 or fewer clicks is: 0.9885

# --- Task 3: Probability of MORE THAN 4 clicks ---
# We use the Survival Function (sf), which is 1 - cdf.
# sf(k) calculates P(X > k).
k_gt_4 = 4
prob_gt_4 = binom.sf(k=k_gt_4, n=n, p=p)
print(f"The probability of more than {k_gt_4} clicks is: {prob_gt_4:.4f}")
# Expected output: The probability of more than 4 clicks is: 0.5832
```