Question 1

> Suppose that the height of men has mean 68 inches and standard deviation 4 inches. We draw 100 men at random. Find (approximately) the probability that the average height of men in our sample will be at least 68.5 inches.

The sample mean $\bar{X}$ will be approximately normally distributed when $n = 100$, with $\mu = 68$ and standard deviation $\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = \frac{4}{\sqrt{100}} = 0.4$.

Since X is normally distributed, we can standardize the random variable $\bar{X}$ to get $Z = \frac{\bar{X}-\mu}{\sigma}$. We can then find $P(\bar{X} \geq 68.5)$ by finding $P(Z \geq \frac{68.5-68}{0.4})$.

The PDF of the standard normal distribution is given by:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

To find the cumulative density, we can integrate the PDF over the provided range:

$$P(Z \geq z) = 1 - P(Z \leq z) = 1 - \int_{-\infty}^{z} f(z) dz$$

For $z = \frac{68.5-68}{0.4}$, we have:

$$P(Z \geq \frac{68.5-68}{0.4}) = 1 - \int_{-\infty}^{\frac{68.5-68}{0.4}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz =$$

$$1 - \int_{-\infty}^{1.25} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \approx$$

$$1 - 0.8944 = \boxed{0.1056}$$

Question 2

Suppose we have a book consisting of n = 100 pages. The number of misprints at each page is independent and has a Poisson distribution with mean 1. Find the probability that the total number of misprints is at least 80 and at most 90 using central limit theorem

---

For a single page, the number of misprints is Poisson distributed with $\lambda = 1$. The sum of n Poisson random variables is also Poisson distributed with $\lambda = n\lambda = 100 \cdot 1 = 100$.

Let S_n be the sum of the number of misprints on each page. Since we have a sufficiently large sample and iid random variables, we can approximate S_n with a normal distribution with $\mu = n\lambda = 100$ and $\sigma = \sqrt{n\lambda} = \sqrt{100} = 10$.

$$S_n \sim N(100, 10^2)$$

We can standardize the random variable S_n to get Z:

$$Z = \frac{S_n - \mu}{\sigma} = \frac{S_n - 100}{10}$$

We can then find $P(80 \leq S_n \leq 90)$ by finding $P(80 \leq S_n \leq 90)$.

First find $P(S_n \leq 90)$:

$$P(S_n \leq 90) = P(Z \leq \frac{90 - 100}{10}) = P(Z \leq -1) \approx 0.1587$$

Then find $P(S_n \leq 80)$:

$$P(S_n \leq 80) = P(Z \leq \frac{80 - 100}{10}) = P(Z \leq -2) \approx 0.0228$$

$P(80 \leq S_n \leq 90)$ will be $P(S_n \leq 90) - P(S_n \leq 80)$:

$$P(80 \leq S_n \leq 90) = 0.1587 - 0.0228 = \boxed{0.1359}$$

Question 3

Let $X_1, \ldots, X_n \sim Poisson(\lambda)$ and let $\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} X_i$
Find the mean squared error of this estimator.

The mean squared error of an estimator is given by:

$$MSE(\hat{\lambda}) = \text{Var}(\hat{\lambda}) + \text{Bias}(\hat{\lambda})^2$$

First, find the expected value of $\hat{\lambda}$:

$$E(\hat{\lambda}) = E(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \frac{1}{n}\sum_{i=1}^{n} \lambda = \lambda$$

Use to find the bias:

$$\text{Bias}(\hat{\lambda}) = E(\hat{\lambda}) - \lambda = \lambda - \lambda = 0$$

Next, find the variance of $\hat{\lambda}$:

$$\text{Var}(\hat{\lambda}) = \text{Var}(\frac{1}{n}\sum_{i=1}^{n} X_i)$$

From the properties of variance, we know that if you have a random variable X and multiply it by a constant a, the variance of the product is a^2 times the variance of X. Therefore:

$$\text{Var}(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(X_i)$$

Since X_i is Poisson distributed with λ, we know that $E(X_i) = \lambda$ and $\text{Var}(X_i) = \lambda$. Therefore:

$$\frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(X_i) = \frac{1}{n^2} \cdot n \cdot \lambda = \frac{\lambda}{n} = \text{Var}(\hat{\lambda})$$

Plug into the MSE formula:

$$MSE(\hat{\lambda}) = \mathsf{Var}(\hat{\lambda}) + \mathsf{Bias}(\hat{\lambda})^2 = \frac{\lambda}{n} + 0 = \boxed{\frac{\lambda}{n}}$$

Question 4

We would like to build a simple model to predict the number of traffic accidents at a junction. The number of accidents is modeled as Poisson distributed. Recall that the Poisson is a discrete distribution over the number of arrivals (accidents) in a fixed time-frame. It has the probability function:

$$\text{Poisson}(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

The parameter λ is the rate parameter that represents the expected number of traffic accidents $E(x) = \lambda$ in a month. To fit the model we need to estimate the rate parameter using some data $X_1, \dots, X_n$, representing the number of accidents in a sample of n months. For this purpose first write the logarithm of the joint probability distribution $\log p(X_1, \dots, X_n; \lambda)$ using summations.

If the random variables $X_1, \dots, X_n$ are iid Poisson distributed with parameter λ, the joint probability distribution is just the product of the individual Poisson probabilities:

$$p(X_1, \dots, X_n; \lambda) = p(X_1; \lambda) \cdot p(X_2; \lambda) \cdot \dots \cdot p(X_n; \lambda) =$$

$$\frac{\lambda^{X_1} e^{-\lambda}}{X_1!} \cdot \frac{\lambda^{X_2} e^{-\lambda}}{X_2!} \cdot \dots \cdot \frac{\lambda^{X_n} e^{-\lambda}}{X_n!} =$$

$$\prod_{i=1}^{n} \frac{\lambda^{X_i} e^{-\lambda}}{X_i!}$$

The logarthim of the joint probability distribution is known as the log-likelihood function and found by taking the natural logarithm of the joint probability distribution:

$$\log p(X_1, \dots, X_n; \lambda) = \log \left(\prod_{i=1}^{n} \frac{\lambda^{X_i} e^{-\lambda}}{X_i!} \right) =$$

$$\sum_{i=1}^{n} \log \left(\frac{\lambda^{X_i} e^{-\lambda}}{X_i!} \right) =$$

$$\sum_{i=1}^{n} \left(\log\left(\lambda^{X_i} e^{-\lambda}\right) - \log(X_i!)\right) =$$

$$\sum_{i=1}^{n} \left(X_i \log(\lambda) - \lambda - \log(X_i!)\right)$$

Question 5

> Compute the maximum likelihood estimate of the rate parameter which maximizes the joint probability found in Question 4 by finding the zero-derivative solution

---

To find the maximum likelihood estimate of the rate parameter λ, we need to find the value of λ that maximizes the log-likelihood function found in Question 4. We can do this by taking the derivative of the log-likelihood function with respect to λ and setting it equal to zero.

The log-likelihood function found in the previous problem is:

$$\sum_{i=1}^{n} (X_i \,\mathsf{log}(\lambda) - \lambda - \mathsf{log}(X_i!))$$

Taking the derivative with respect to λ:

$$\frac{d}{d\lambda} \sum_{i=1}^{n} (X_i \,\mathsf{log}(\lambda) - \lambda - \mathsf{log}(X_i!)) =$$

Since derivative is distributive:

$$\sum_{i=1}^{n} \frac{d}{d\lambda} (X_i \,\mathsf{log}(\lambda) - \lambda - \mathsf{log}(X_i!)) =$$

$$\sum_{i=1}^{n} \frac{d}{d\lambda} X_i \,\mathsf{log}(\lambda) - \frac{d}{d\lambda}\lambda - \frac{d}{d\lambda}\,\mathsf{log}(X_i!) =$$

Since X_i is a constant with respect to λ:

$$\sum_{i=1}^{n} \frac{X_i}{\lambda} - 1$$

Distribute the summation and factor out the λ:

$$\frac{1}{\lambda} \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} 1 =$$

$$\frac{1}{\lambda}\sum_{i=1}^{n} X_i - n$$

Set the derivative equal to zero and solve for λ to find the maximum likelihood estimate:

$$\frac{1}{\lambda}\sum_{i=1}^{n} X_i - n = 0$$

$$\frac{1}{\lambda}\sum_{i=1}^{n} X_i = n$$

$$\lambda = \boxed{\frac{1}{n}\sum_{i=1}^{n} X_i}$$

Question 6

How many accidents are expected in the next month under this model, if in the last three months $X_1 = 2$, $X_2 = 5$, $X_3 = 3$ accidents were observed?

Using the maximum likelihood estimate of the rate parameter found in the previous problem:

$$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

We can find the expected number of accidents in the next month by plugging in the observed values of X_1, X_2, X_3:

$$\hat{\lambda} = \frac{1}{3} \sum_{i=1}^{3} X_i = \frac{1}{3} \cdot (2 + 5 + 3) = \frac{10}{3}$$

Since accidents are modeled as Poisson distributed, the expected number of accidents in the next month is equal to the rate parameter λ:

$$E(X) = \hat{\lambda} = \boxed{\frac{10}{3}}$$

Question 7

Let us numerically verify the law of large numbers. We will simulate $m = 100$ sample mean trajectories of $X_1, \ldots, X_N \sim Bernoulli(\mu = 0.2)$ and plot them altogether in one plot. Here, a sample mean trajectory means a sequence of $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_N$ where $\hat{X}_i$ is the sample mean using samples $X_1, \ldots, X_i$. We will plot $\hat{X}_n$ as a function of n, but do this multiple times. Take n from 1 to $N = 1000$. An ideal plot would look like Figure 1-Left. You must use the 'alpha' option to `pyplot.plot()` to give some transparency (you should obtain a similar look visualization as the one in figure). You may want to use the 'color' option to specify the color.

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

np.random.seed(0)

m = 100 # Number of sample mean trajectories
N = 1000 # Number of samples
mu = 0.2 # Probability of success

# Shape: [sample, Bernoulli trials]
X = np.random.binomial(1, mu, (m, N))

# Shape: [sample, cumulative number of successes up to each trial]
X_cumsum = np.cumsum(X, axis=1)

# Shape: [sample, sample mean up to each trial]
X_mean = X_cumsum / np.arange(1, N + 1)

plt.figure(figsize=(10, 5))
for i in range(m):
    plt.plot(np.arange(1, N + 1), X_mean[i], alpha=0.1, color='blue')

plt.plot(np.arange(1, N + 1), np.full(N, mu), color='red')

plt.xlabel('n')
plt.ylabel('Sample Mean')
plt.title('Sample Mean Trajectories of Bernoulli Random Variables')
plt.show()
```

Question 8

Let us verify the central limit theorem (CLT) by simulation. For N ∈{10,100,1000,10000}, perform:

- Take N samples from $Bernoulli(\mu = 0.05)$ and compute the sample mean. Repeat this 1000 times.
- Plot those 1000 numbers as a histogram (`pyplot.hist`) with a proper number of bins. Use density=True.
- With a red line, overlay the pdf of a Gaussian distribution with the parameters suggested by the CLT (figure this out!).

An ideal answer would look like Figure 1-Right. To receive full credit, you must use py- plot.subplot to have four plots in one figure.

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

np.random.seed(0)

N = [10, 100, 1000, 10000] # Number of samples
mu = 0.05 # Probability of success
m = 1000 # Number of samples

fig, axs = plt.subplots(2, 2, figsize=(10, 10))

for i, n in enumerate(N):
    # Shape: [sample, Bernoulli trials]
    X = np.random.binomial(1, mu, (m, n))

    # Shape: [sample, sample mean]
    X_mean = np.mean(X, axis=1)

    # Plot histogram
    axs[i // 2, i % 2].hist(X_mean, bins=30, density=True, color='blue', alpha=0.7)

    # CLT parameters
    mu_clt = mu

    # Since X is Bernoulli distributed, the variance is mu * (1 - mu)
    sigma_clt = np.sqrt(mu * (1 - mu) / n)

    # Plot Gaussian PDF using CLT parameters (since the sample mean is normally distributed
```

```
    x = np.linspace(mu_clt - 4 * sigma_clt, mu_clt + 4 * sigma_clt, 100)

    # Plot the PDF of the normal distribution
    axs[i // 2, i % 2].plot(x, stats.norm.pdf(x, mu_clt, sigma_clt), color='red')

    axs[i // 2, i % 2].set_title(f'N = {n}')
    axs[i // 2, i % 2].set_xlabel('Sample Mean')
    axs[i // 2, i % 2].set_ylabel('Density')

plt.tight_layout()
plt.show()
```

Question 9

This question shows a way of estimating the correlation ρ of two random variables X, Y. For our chosen model, we will use a bivariate Gaussian distribution $(X, Y)^T$. Note that such a distribution denoted with $N(\mu, \Sigma)$, has two parameters, where μ denotes the 2-dimensional mean vector consisting of $\mu x, \mu y$ and Σ denotes the covariance matrix. The entry $(1, 1)$ of Σ is $Cov(X, X)$, the entry $(2, 2)$ is $Cov(Y, Y)$, and the entries $(1, 2), (2, 1)$ are $Cov(X, Y)$. For this example the means are $\mu_x = \mu_x = 0$, the standard deviations are $\sigma_x = \sigma_y = 1$, and the true (unknown) correlation is $\rho = 0.6$. Therefore the covariance matrix is

$$\Sigma = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

Using `numpy.random.seed` set your random number generator seed to 0 and answer the following:

Create a dataset by drawing $N = 500$ samples from our model using `numpy.random` function multivariate normal. Compute and report the plug-in estimator of correlation, given by:

$$\hat{\rho} = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2 \sum_{i=1}^{N}(Y_i - \bar{Y})^2}}$$

Where $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$ is the sample mean (and similarly for Y).

```python
import numpy as np

np.random.seed(0)

N = 500 # Number of samples
mu = [0, 0] # Mean vector
cov = [[1, 0.6], [0.6, 1]] # Covariance matrix

# Generate samples from bivariate Gaussian distribution
```

```python
X, Y = np.random.multivariate_normal(mu, cov, N).T # Shape: [N]

# Compute sample means
X_mean = np.mean(X)
Y_mean = np.mean(Y)

# Compute plug-in estimator of correlation
numerator = np.sum((X - X_mean) * (Y - Y_mean))
denominator = np.sqrt(np.sum((X - X_mean)**2) * np.sum((Y - Y_mean)**2))

# Plug-in estimator of correlation to find the correlation
rho_hat = numerator / denominator

print(f'Plug-in estimator of correlation: {rho_hat}')
# >>> Plug-in estimator of correlation: 0.5826300529635126
```

```
0.5826300529635126
```

Question 10

Repeat the above process $m = 5,000$ times to generate $\hat{\rho}_1, \ldots, \hat{\rho}_m$, each one based on a fresh set of $N = 500$ samples. Display a histogram of your m estimates using `matplotlib.pyplot.hist` with 30 bins. Label the axes.

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

np.random.seed(0)

N = 500 # Number of samples
m = 5000 # Number of iterations
mu = [0, 0] # Mean vector
cov = [[1, 0.6], [0.6, 1]] # Covariance matrix

rhos = [] # Store correlation estimates

for i in range(m):
    # Generate samples from bivariate Gaussian distribution
    X, Y = np.random.multivariate_normal(mu, cov, N).T # Shape: [N]

    # Compute sample means
    X_mean = np.mean(X)
    Y_mean = np.mean(Y)

    # Compute plug-in estimator of correlation
    numerator = np.sum((X - X_mean) * (Y - Y_mean))
    denominator = np.sqrt(np.sum((X - X_mean)**2) * np.sum((Y - Y_mean)**2))

    # Plug-in estimator of correlation to find the correlation
    rho_hat = numerator / denominator
    rhos.append(rho_hat)

plt.hist(rhos, bins=30, color='blue', alpha=0.7)
plt.xlabel('Correlation Estimate')
plt.ylabel('Frequency')
plt.title('Histogram of Correlation Estimates')
plt.show()
```

Question 11

Use m estimates obtained in the above question to estimate $E[(\widehat{\rho} - \rho)^2]$, the mean square error (MSE) of plug-in estimator $\widehat{\rho}$. What is the value of your MSE estimate?

```
# MSE = E[(rho_hat - rho)^2] (we found rho = 0.6 in the problem statement)
mse = np.mean((np.array(rhos) - 0.6)**2)

print(f'MSE of plug-in estimator: {mse}')
# >>> MSE of plug-in estimator: 0.0008393677723569983
```

0.0008393677723569983