1. Write a `pois.prob()` function that computes $P(X = x)$, $P(X \neq x)$, $P(X < x)$, $P(X \leq x)$, $P(X > x)$, and $P(X \geq x)$. Enable the user to specify the rate parameter λ.

```r
pois.prob <- function(x, lambda, type){
  if (type == "="){
    return (dpois(x, lambda)) #P(X=x) def of PMF
  } else if (type== "!=") {
    return (1 - dpois(x, lambad)) # complement rule
  } else if (type== "<"){
    return(ppois(x-1, lambda))  # P(X < x) = P(X <= x-1)
  } else if (type == "<=") {
    return(ppois(x, lambda)) # P (X < = x) def of CDF
  } else if (type == ">") {
    return(1 - ppois(x, lambda)) # complement rule
  } else if (type == ">="){
    return(1 - ppois(x-1, lambda)) # P (X >= x) = 1 - P(X<x) = 1 - P(X <= x-1)
  }
}
```

In this step, I implemented `pois.prob()`, a function that computes probabilities from a Poisson distribution given an input value `x`, rate parameter `lambda`, and a probability type. The Poisson distribution is a discrete probability distribution, meaning it assigns probabilities to specific values of the random variable X. We calculate the probability at a given value `x` using the probability mass function (PMF).

When the type is `"="`, the function computes the probability of `X=x` using the PMF. For `"!="`, the probability is computed as the complement of `P(X=x)`, i.e., $P(X \neq x) = 1 - P(X = x)$. To compute $P(X < x)$, we use $P(X \leq x - 1)$, which is given by the CDF at $x - 1$. This works because the Poisson distribution only takes integer values. Therefore, since the probability X is strictly less than x, it is equivalent to the probability that X is less than or equal to $x - 1$, so we can calculate this as the CDF at $x - 1$. For `"<="`, we directly use the CDF, as that is the definition. Finally, we use the complement rule for `">"` and `">="`.

The key distinction here is that Poisson distributions are discrete, and thus we can compute the probability of specific values of X. For probabilities involving inequalities, we utilize the CDF and apply the complement rule and the fact that discrete distributions only take on integer values when needed.

2. Write a `beta.prob()` function that computes $P(X = x)$, $P(X \neq x)$, $P(X < x)$, $P(X \leq x)$, $P(X > x)$, and $P(X \geq x)$ for a beta distribution. Enable the user to specify the shape parameters α and β.

```r
beta.prob <- function(x, alpha, beta, type){
  if (type == "="){
    return(0)                 # prob of exact value of continuous dist always 0
  } else if (type== "!=") {
    return(1)                 # complement rule
  } else if (type== "<"){
    return(pbeta(x, alpha, beta)) # P(X<x) bc continous = P(X<=x)
  } else if (type == "<=") {
    return(pbeta(x, alpha, beta)) # P(X<=x) def of CDF
  } else if (type == ">") {
    return(1 - pbeta(x, alpha, beta)) # complement rule
  } else if (type == ">="){
    return(1 - pbeta(x, alpha, beta)) # same as above bc continuous
  }
}
```

In this step, I implemented `beta.prob()`, a function that computes probabilities from a Beta distribution given an input value `x`, shape parameters `alpha` and `beta`, and a probability type. Unlike the Poisson distribution, which is discrete, the Beta distribution is a continuous probability, meaning it does not assign probabilities to specific values of X. Instead, it assigns probabilities to intervals of X, so the probability of any exact value is always zero. Therefore, when calculating probabilities, we never use the probability mass function (PMF). We only use the cumulative distribution function (CDF) to calculate probabilities.

When the type is `"="`, the probability of `X=x` is zero, as continuous distributions do not assign probabilities to single points. For `"!="`, the probability is computed as the complement, meaning $P(X \neq x) = 1$

because the probability of any specific value is always zero. To compute $P(X < x)$ or $P(X \leq x)$, we use the CDF. In continuous distributions, $P(X < x) = P(X \leq x)$ because the probability of a single point is zero. Similarly, for `">"` and `">="`, these values are equal, and we again apply the complement rule.

In summary, the key difference between continuous and discrete distributions is that for continuous distributions, we cannot compute probabilities of exact values of X. Instead, probabilities are assigned to intervals, and we rely on the CDF for all calculations.