R

Basics

Variables

```r
x <- 6
y = 7
```

String Interpolation

```r
name <- "John"
sprintf("Hello, %s!", name)
```

Data Types

```r
x <- 6 # Numeric
y <- "Hello" # Character
z <- TRUE # Logical
```

Data Structures

```r
# Vector
x <- c(1, 2, 3, 4, 5)
print(x[1]) # 1, Array indices start from 1

# Matrix
y <- matrix(1:6, nrow = 2, ncol = 3) # 2x3 matrix

# List
z <- list(1, "Hello", TRUE)
```

Difference between Vector and List

- Vector: All elements must be of the same type.
- List: Elements can be of different types.

Functions

```
add <- function(x, y) {
  return(x + y)
}
```

Control Structures

```
if (x > 5) {
  print("x is greater than 5")
} else {
  print("x is less than or equal to 5")
}
```

Loops

```
for (i in 1:5) {
  print(i)
}
```

Importing Datasets

```
attach(iris)

# Print the first n rows
head(iris, n=5)

# Print the last n rows
tail(iris, n=5)

# Print the structure of the dataset
str(iris)

# Print the summary of the dataset
summary(iris)

# Print the dimensions of the dataset
dim(iris)

# Print the column names of the dataset
names(iris)

# Print the first n rows of the dataset
iris[1:5,]
```

```r
# Print the first n rows of the dataset for a specific
column
iris[1:5, "Sepal.Length"]

# Print the first n rows of the dataset for multiple columns
iris[1:5, c("Sepal.Length", "Sepal.Width")]
```

Statistical Functions

```r
x <- c(1, 2, 3, 4, 5)

# Mean
mean(x)

# Median
median(x)

# Standard Deviation
sd(x)

# Mode
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

Mode(x)
```

Probability Distributions

Normal Distribution

- `dbinom(x, size, prob)`: **Probability Mass Function:** Returns the probability of getting exactly x successes in n trials.

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

- `pbinom(x, size, prob)`: **Cumulative Distribution Function:** Returns the probability of getting x or fewer successes in n trials.

$$P(X \leq x) = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}$$

- `qbinom(p, size, prob)`: **Quantile Function:** Returns the number of successes such that the probability of getting x or fewer successes is p.
- `rbinom(n, size, prob)`: Random Sampling

Poisson Distribution

- `dpois(x, lambda)`: **Probability Mass Function:** Returns the probability of getting exactly x successes in a Poisson distribution with mean lambda.

$$P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}$$

- `ppois(x, lambda)`: **Cumulative Distribution Function:** Returns the probability of getting x or fewer successes in a Poisson distribution with mean lambda.

$$P(X \leq x) = \sum_{i=0}^{x} \frac{e^{-\lambda}\lambda^i}{i!}$$

- `qpois(p, lambda)`: **Quantile Function:** Returns the number of successes such that the probability of getting x or fewer successes is p.
- `rpois(n, lambda)`: Random Sampling

Expected Value

Discrete Random Variable

```r
x <- c(1, 2, 3, 4, 5)
p <- c(0.1, 0.2, 0.3, 0.2, 0.2)

# Expected Value
sum(x * p)
weighted.mean(x, p)
```

Continuous Random Variable

```r
f <- function(x) {
  return(2*x)
}

# Expected Value
integrate(f, 0, 1)
```

Miscellaneous

Plotting Graphs

```r
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)

# Scatter Plot
plot(x, y)

# Line Plot
plot(x, y, type = "l")

# Bar Plot
barplot(y)
```

Birthday Paradox

```r
birthday_paradox <- function(n) {
  p <- 1
  for (i in 1:(n-1)) {
    p <- p * (365 - i) / 365
  }
  return(1 - p)
}

birthday_paradox(23)

# OR
pbirthday(23)
```

Sampling

- Repeating a value in a vector:

```r
x <- c(rep('H', 5), rep('T', 10))
```

Random Sampling

```r
x <- c(rep("Heads",10), rep("Tails", 10))

# Sample 5 elements from x
sample(x, 5)

# Sample 5 elements with replacement
sample(x, 5, replace = TRUE)

# Print all possible combinations of 2 elements from x
combn(x, 2)

# Print all possible permutations of 2 elements from x
permn(x, 2)
```