

Βασικές εντολές, κατανομές και προεπεξεργασία δεδομένων στην R

Πώς τρέχουμε τον κώδικα:

1. Ανοίξτε το RStudio.
2. Δημιουργήστε ένα νέο script (File > New File > R Script).
3. Επικολλήστε τον παραπάνω κώδικα μέσα στο script.
4. Αποθηκεύστε το με ένα ενδεικτικό όνομα, π.χ., r_course_2.R.
5. Εκτελέστε γραμμή προς γραμμή (επιλέξτε και πατήστε Ctrl+Enter ή χρησιμοποιήστε το κουμπί "Run") ώστε να παρακολουθείτε τη διαδικασία. Διαφορετικά, μπορείτε να επιλέξετε και να τρέξετε ολόκληρο τον κώδικα απευθείας.

Αντιγράψτε το παρακάτω κείμενο στο RStudio

Θα το τρέξουμε γραμμή-γραμμή

```
# =====  
# Basic Statistical Distributions and Data Preprocessing  
# =====  
# -----  
# 1. Basic Statistical Distributions  
# -----  
# Probability Distributions - Probabilities  
# Binomial Distribution  
# For example, 10 trials, 3 successes, success probability=0.5  
dbinom(x=0:10, size=10, prob=0.5)  
# Calculating probability of exactly 3 successes  
dbinom(3, 10, 0.5)  
  
# Probabilities for all possible outcomes  
plot(0:10, dbinom(0:10, 10, 0.5), type='h',  
     main="Binomial Distribution (n=10, p=0.5)", xlab="Number of Successes",  
     ylab="Probability")
```

```

# Normal Distribution (Gaussian)
# Typically models continuous data
x <- seq(-4, 4, length=100)
y <- dnorm(x, mean=0, sd=1)
plot(x, y, type='l', main="Normal Distribution (mean=0, sigma=1)",
      ylab="Density", xlab="Values")

# Geometric Distribution
# For example, number of trials until the first success with p=0.3
x <- 1:20
plot(x, dgeom(x-1, prob=0.3), type='h',
      main="Geometric Distribution", xlab="Trials until first success",
      ylab="Probability")

# Estimating probabilities with the normal distribution
# Probability that a standard normally distributed variable is within ±1 std dev
pnorm(1, mean=0, sd=1) - pnorm(-1, mean=0, sd=1)

# -----
# 2. Plotting with Base R
# -----

# Sampling from a normal distribution
sample_data <- rnorm(1000, mean=0, sd=1)

# Histogram of the sample
hist(sample_data, main="Histogram of Normal Distribution", col="orange",
      xlab="Values")

# Bar plot of positive vs. negative counts
positive_counts <- table(sample_data > 0)
barplot(positive_counts, main="Proportion of Positive Data", col="skyblue",
      ylab="Count")

# Overlay normal density on sample data plot
plot(sample_data, main="Sample Data and Normal Density", col="purple")
hist(sample_data, probability = TRUE, main = "Histogram with Normal Density", col
     = "orange", xlab = "Values")

```

```

curve(dnorm(x, mean=mean(sample_data), sd= sd(sample_data)), add=TRUE, col =
"red", lwd=2) # Add the normal density scaled appropriately

# -----
# 3. Data Preprocessing: Handling Outliers & Missing Values
# -----
# Create sample data with missing values and outliers
set.seed(123)
data <- rnorm(100) # Generate 100 random normal values
data[sample(1:100, 10)] <- NA # Inject missing values (NA)
data[which(data > 3)] <- 10 # Add some outliers (values outside typical range)

# Check for missing data
sum(is.na(data)) # Count of missing (NA) values

# Removing missing values
data_no_na <- na.omit(data)

# Find outliers based on a common rule (e.g., beyond 3 standard deviations)
mean_data <- mean(data, na.rm=TRUE)
sd_data <- sd(data, na.rm=TRUE)
lower_bound <- mean_data - 3*sd_data
upper_bound <- mean_data + 3*sd_data

# Identify outliers
outliers <- data[which(data < lower_bound | data > upper_bound)]
print("Detected outliers:")
print(outliers)

# Visualize data with outliers
hist(data, main="Histogram with Outliers and Missing Values", col="lightgreen",
xlab="Values")
abline(v=c(lower_bound, upper_bound), col="red", lty=2)
legend("topright", legend=c("Outlier thresholds"), col="red", lty=2)

# Summary of data excluding missing values
summary(data, na.rm=TRUE)

```