KandruB.CS5200.ComplXPath

Bhavitha Naga Sai Kandru

```r
# Load necessary library
library(xml2)
library(XML)
```

```r
fp <- "CustomersAndOrders.xml"
xmlDocument = xmlParse(fp)
r = xmlRoot(xmlDocument)
```

Question 3

Add a DTD to the XML that validates the XML and save the XML in a new file called orders.xml.

```r
# Define the DTD as a string
dtd_string <- '
<!DOCTYPE Root [
<!ELEMENT Root (Customers, Orders?)>
<!ELEMENT Customers (Customer+)>
<!ELEMENT Customer (CompanyName, ContactName, ContactTitle, Phone, FullAddress)>
<!ATTLIST Customer CustomerID ID #REQUIRED>
<!ELEMENT CompanyName (#PCDATA)>
<!ELEMENT ContactName (#PCDATA)>
<!ELEMENT ContactTitle (#PCDATA)>
<!ELEMENT Phone (#PCDATA)>
<!ELEMENT FullAddress (Address, City, Region?, PostalCode, Country)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Region (#PCDATA)>
<!ELEMENT PostalCode (#PCDATA)>
<!ELEMENT Country (#PCDATA)>
<!ELEMENT Orders (Order+)>
<!ELEMENT Order (Product+, TotalPrice)>
<!ATTLIST Order OrderID ID #REQUIRED>
<!ELEMENT Product (ProductName, Quantity, Price)>
<!ELEMENT ProductName (#PCDATA)>
<!ELEMENT Quantity (#PCDATA)>
<!ELEMENT Price (#PCDATA)>
<!ELEMENT TotalPrice (#PCDATA)>
]>
'

# Read the original XML content, ensuring it starts with the XML declaration
xml_path <- 'CustomersAndOrders.xml'
xml_content <- readLines(xml_path, warn = FALSE)
```

```r
# Extract the XML declaration assuming it's the first line of the XML content
xml_declaration <- xml_content[1]

# Remove the first line if it is the XML declaration to avoid duplicating it
xml_content <- xml_content[-1]

# Directly concatenate the XML declaration (if it exists), DTD with the XML content
# Assuming the XML declaration is necessary and present
xml_content_new <- paste(xml_declaration, dtd_string, paste(xml_content, collapse = "\n"), sep = "\n")

# Ensure no leading empty lines and write to a new file
file_path <- "orders.xml"
writeLines(trimws(xml_content_new), file_path)
cat("XML with DTD saved to:", file_path, "\n")
```

```
## XML with DTD saved to: orders.xml
```

Question 4:

In a code block, load the XML containing the DTD into R with validation.

```r
# Load and validate the XML with DTD
xml_file_with_dtd <- file_path
doc <- tryCatch({
  xmlParse(file = xml_file_with_dtd, useInternalNodes = TRUE)
}, error = function(e) {
  cat("Error in XML parsing/validation:", e$message, "\n")
  NULL
})

if (!is.null(doc)) {
  cat("XML document loaded and validated successfully.\n")
} else {
  cat("Failed to load and/or validate the XML document.\n")
}
```

```
## XML document loaded and validated successfully.
```

Question 5:

In a new code block, execute an XPath expression that returns the names of all customers that are in the USA.

```r
# Execute XPath expression to get customer names
doc <- xmlParse(file_path)
customer_names <- xpathSApply(doc, "//Customer[FullAddress/Country = 'USA']/CompanyName", xmlValue)

# Print the names of customers living in "USA"
print(customer_names)
```

```
## [1] "Great Lakes Food Market"    "Hungry Coyote Import Store"
## [3] "Lazy K Kountry Store"       "Let's Stop N Shop"
```

Question 6:

Using the result returned in (5) and any additional queries required, calculate the percentage of customers who do are not in the USA. Display the result as markup in your notebook.

```
total_customers<- as.numeric(xpathSApply(doc, "count(//Customer)", xmlValue))
count<-length(customer_names)

# Calculate the number of customers not in the USA
non_usa_customers <- total_customers - count

# Calculate the percentage of customers not in the USA
percentage <- (non_usa_customers/total_customers)*100

# Print  the percentage of customers who do are not in the USA
cat("Percentage of customers not living in USA are",percentage,"%")
```

```
## Percentage of customers not living in USA are 20 %
```

Question 7:

Using a combination of R and XPath, calculate the total amount paid for freight for all orders within the USA.

```
freight_values <- xpathSApply(doc, "//Order[ShipInfo/ShipCountry='USA']/ShipInfo/Freight", xmlValue)

# Convert the result to numeric and calculate the Sum
total_freight <- sum(as.numeric(freight_values))

# Print the total amount paid for freight
cat("The total amount paid for freight for all orders within the USA are",total_freight)
```

```
## The total amount paid for freight for all orders within the USA are 1516.2
```

Question 8:

Using a combination of R and XPath, calculate the average amount paid for freight for all orders shipped to the USA.

```
freight <- xpathSApply(doc, "//Order[ShipInfo/ShipCountry='USA']/ShipInfo/Freight", xmlValue)

# Convert the result to numeric and calculate the average
freight_values <- as.numeric(freight)
average_freight <- mean(freight_values)

# Print the average amount paid for freight
cat("The average amount paid for freight for all orders shipped to the USA is",average_freight)
```

```
## The average amount paid for freight for all orders shipped to the USA is 68.91818
```