Worksheet#5_group(Berja,Bibit,Buenvenida)

Berja,Bibit,Buenvenida

2024-11-06

Extracting Amazon Product Reviews

```
library(polite)
library(rvest)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
library(ggplot2)

# Function to scrape products from a given category URL
scrape_amazon_category <- function(category_url, category_name) {
  # Create a polite session
  session <- bow(category_url, user_agent = "Educational")

  # Scrape the page
  page <- tryCatch({
    scrape(session)
  }, error = function(e) {
    message("Error scraping the page: ", e)
    return(NULL)  # Return NULL if there's an error
  })

  # Check if page is NULL
  if (is.null(page)) {
    return(tibble())  # Return empty tibble if page is NULL
  }

  # Extract product details
  products <- page %>%
    html_nodes(".s-main-slot .s-result-item") %>%
    map_df(~ {
      # Safely extract each element and handle potential NULLs
      title <- .x %>% html_node("h2") %>% html_text(trim = TRUE)
      price <- .x %>% html_node(".a-price .a-offscreen") %>% html_text(trim = TRUE)
      description <- .x %>% html_node(".a-text-normal") %>% html_text(trim = TRUE)
      rating <- .x %>% html_node(".a-icon-alt") %>% html_text(trim = TRUE)
      reviews <- .x %>% html_node(".a-size-small .a-link-normal") %>% html_text(trim = TRUE)

      tibble(
        Title = title,
        Price = price,
        Description = description,
        Rating = rating,
        Reviews = reviews,
        Category = category_name  # Ensure the category name is included
      )
    }) %>%
    filter(!is.na(Title))  # Filter out rows with missing titles

  # Check if products were found
  if (nrow(products) == 0) {
    message("No products found for category: ", category_name)
  } else {
    message("Scraped ", nrow(products), " products from category: ", category_name)
  }

  return(products)
}

# Example category URLs (you need to adjust these)
categories <- list(
  fishing = 'https://www.amazon.com/s?k=fishing',
  electronics = 'https://www.amazon.com/s?k=electronics',
  books = 'https://www.amazon.com/s?k=books',
  home_kitchen = 'https://www.amazon.com/s?k=home+kitchen',
  clothing = 'https://www.amazon.com/s?k=clothing'
)

# Initialize an empty data frame to store all products
all_products <- tibble()

# Loop through categories and scrape products
for (category_name in names(categories)) {
  category_url <- categories[[category_name]]
  category_products <- scrape_amazon_category(category_url, category_name)
  all_products <- bind_rows(all_products, category_products)
}
```

```
## Warning: Server error: (503) Service Unavailable
## https://www.amazon.com/s?k=fishing
```

```
## Warning: Server error: (503) Service Unavailable
## https://www.amazon.com/s?k=electronics
```

```
## Warning: Server error: (503) Service Unavailable
## https://www.amazon.com/s?k=books
```

```
## Warning: Server error: (503) Service Unavailable
## https://www.amazon.com/s?k=home+kitchen
```

```
## Warning: Server error: (503) Service Unavailable
## https://www.amazon.com/s?k=clothing
```

```r
# Check if 'Category' exists before grouping
if ("Category" %in% colnames(all_products)) {
  # Convert Price and Rating to numeric for analysis
  all_products$Price <- as.numeric(gsub("\\$", "", gsub(",", "", all_products$Price)))
  all_products$Rating <- as.numeric(gsub(" out of 5 stars", "", all_products$Rating))

  # Prepare data for bar plots
  avg_price <- all_products %>%
    group_by(Category) %>%
    summarize(Average_Price = mean(Price, na.rm = TRUE), .groups = 'drop')

  avg_rating <- all_products %>%
    group_by(Category) %>%
    summarize(Average_Rating = mean(Rating, na.rm = TRUE), .groups = 'drop')

  # Create a bar plot for Average Price by Category using ggplot2
  ggplot(avg_price, aes(x = reorder(Category, Average_Price), y = Average_Price)) +
    geom_bar(stat = "identity", fill = "lightblue") +
    labs(title = "Average Price of Products by Category",
         x = "Category",
         y = "Average Price ($)") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))

  # Create a bar plot for Average Rating by Category using ggplot2
  ggplot(avg_rating, aes(x = reorder(Category, Average_Rating), y = Average_Rating)) +
    geom_bar(stat = "identity", fill = "pink") +
    labs(title = "Average Ratings of Products by Category",
         x = "Category",
         y = "Average Rating (out of 5)") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
} else {
  message("Column 'Category' not found in all_products.")
}
```

```
## Column 'Category' not found in all_products.
```