PRACTICAL NO 11

AIM Reshaping data using pivot_longer()/pivot_wider() (R).

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

```
> library(dplyr)
> library(tidyr)
> # Use readr::read_csv (explicit) and correct 'na' argument
> df <- readr::read_csv("R/flipkart_com-ecommerce_sample.csv", na = c("", "NA")) %>%
+   mutate(InvoiceIndex = row_number())
Rows: 20000 Columns: 15
-- Column specification ----------------------------------------------------------------------
Delimiter: ","
chr (12): uniq_id, crawl_timestamp, product_url, product_name, product_category_tree, pid, image, description, product_rating, overall_rating, brand, produ...
dbl  (2): retail_price, discounted_price
lgl  (1): is_FK_Advantage_product

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> # Print column names so you can confirm what's actually in the file
> cat("--- Columns in the imported CSV ---\n")
--- Columns in the imported CSV ---
> print(colnames(df))
 [1] "uniq_id"               "crawl_timestamp"       "product_url"           "product_name"          "product_category_tree"
 [6] "pid"                   "retail_price"          "discounted_price"      "image"                 "is_FK_Advantage_product"
[11] "description"           "product_rating"        "overall_rating"        "brand"                 "product_specifications"
[16] "InvoiceIndex"
> # Try to keep only the columns of interest if they exist (avoids select() errors)
> df <- df %>% select(InvoiceIndex, any_of(c("Product.line", "Unit.price", "Tax.5.")))
> cat("--- 1. Original Wide Data ---\n")
--- 1. Original Wide Data ---
> print(head(df))
# A tibble: 6 x 1
  InvoiceIndex
         <int>
1            1
2            2
3            3
4            4
5            5
```
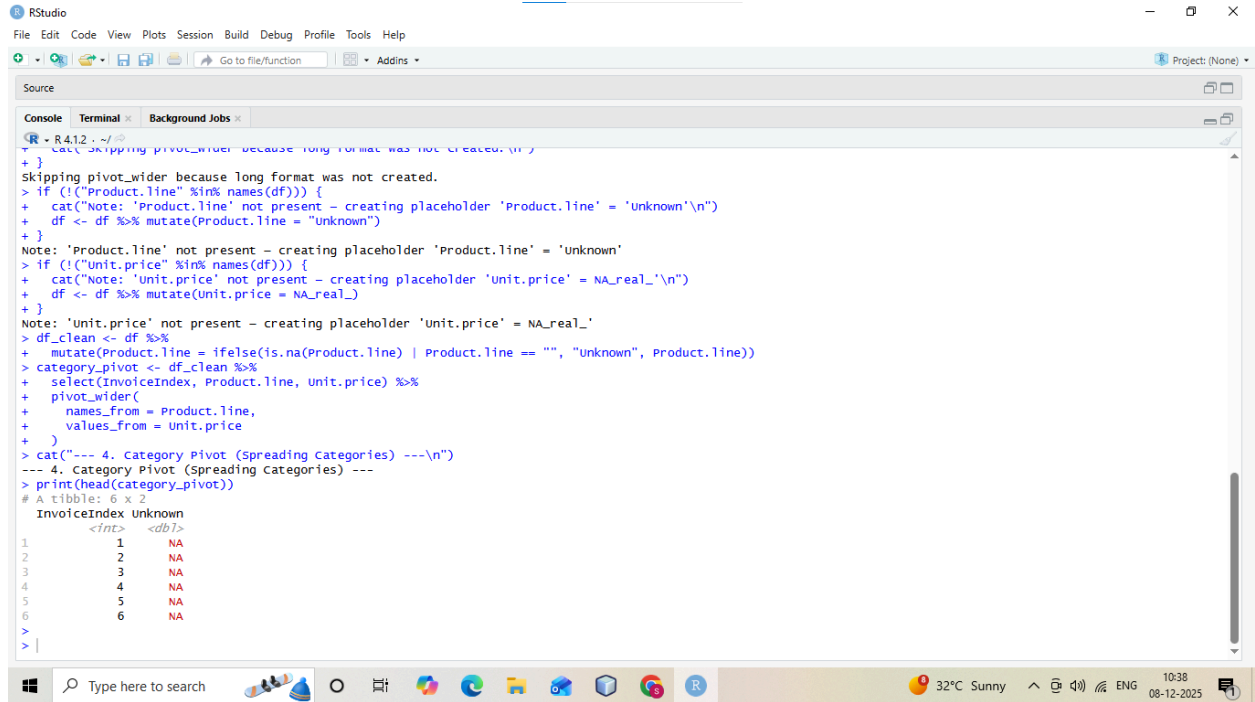
32°C Sunny   ENG   10:38 08-12-2025

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

```
2            2
3            3
4            4
5            5
6            6
> pivot_cols <- intersect(names(df), c("Unit.price", "Tax.5."))
> if (length(pivot_cols) == 0) {
+   cat("Warning: No pivot columns (Unit.price or Tax.5.) found - skipping pivot_longer\n")
+   long_df <- df   # fallback: keep original
+ } else {
+   long_df <- df %>%
+     pivot_longer(
+       cols = any_of(pivot_cols),
+       names_to = "Metric",
+       values_to = "Value"
+     )
+   cat("--- 2. Long Format (pivot_longer) ---\n")
+   print(head(long_df, 6))
+ }
Warning: No pivot columns (Unit.price or Tax.5.) found - skipping pivot_longer

> if ("Metric" %in% names(long_df)) {
+   wide_df <- long_df %>%
+     pivot_wider(
+       names_from = Metric,
+       values_from = Value
+     )
+   cat("--- 3. Wide Format (Back to Original) ---\n")
+   print(head(wide_df))
+ } else {
+   cat("Skipping pivot_wider because long format was not created.\n")
+ }
Skipping pivot_wider because long format was not created.
> if (!("Product.line" %in% names(df))) {
```

32°C Sunny   ENG   10:38 08-12-2025

```
+   cat("Skipping pivot_wider because long format was not created.\n")
+ }
Skipping pivot_wider because long format was not created.
> if (!("Product.line" %in% names(df))) {
+   cat("Note: 'Product.line' not present — creating placeholder 'Product.line' = 'Unknown'\n")
+   df <- df %>% mutate(Product.line = "Unknown")
+ }
Note: 'Product.line' not present — creating placeholder 'Product.line' = 'Unknown'
> if (!("Unit.price" %in% names(df))) {
+   cat("Note: 'Unit.price' not present — creating placeholder 'Unit.price' = NA_real_'\n")
+   df <- df %>% mutate(Unit.price = NA_real_)
+ }
Note: 'Unit.price' not present — creating placeholder 'Unit.price' = NA_real_'
> df_clean <- df %>%
+   mutate(Product.line = ifelse(is.na(Product.line) | Product.line == "", "Unknown", Product.line))
> category_pivot <- df_clean %>%
+   select(InvoiceIndex, Product.line, Unit.price) %>%
+   pivot_wider(
+     names_from = Product.line,
+     values_from = Unit.price
+   )
> cat("--- 4. Category Pivot (Spreading Categories) ---\n")
--- 4. Category Pivot (Spreading Categories) ---
> print(head(category_pivot))
# A tibble: 6 x 2
  InvoiceIndex Unknown
         <int>   <dbl>
1            1      NA
2            2      NA
3            3      NA
4            4      NA
5            5      NA
6            6      NA
>
> |
```