R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
# Load necessary libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.1     ✓ stringr   1.5.2
## ✓ ggplot2   4.0.0     ✓ tibble    3.3.0
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.1.0
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

Task 1 — Load and inspect the dataset

1. Load the Online Retail dataset.
2. Display:

• the first few rows,

• number of rows and columns,

• column names,

• basic data types.

```
ds <- read.csv("online_retail.csv") # 1.Loading the given dataset.
# 2. Getting the First few rows, number of rows and number of columns, column
names and basic data types.
head(ds)
```

```
##   Invoice StockCode                          Description Quantity
## 1  489434     85048 15CM CHRISTMAS GLASS BALL 20 LIGHTS       12
## 2  489434    79323P                   PINK CHERRY LIGHTS       12
## 3  489434    79323W                  WHITE CHERRY LIGHTS       12
## 4  489434     22041         RECORD FRAME 7" SINGLE SIZE        48
```

```
## 5  489434     21232      STRAWBERRY CERAMIC TRINKET BOX       24
## 6  489434     22064          PINK DOUGHNUT TRINKET POT        24
##         InvoiceDate Price Customer.ID        Country
## 1 01-12-2009 07:45  6.95       13085 United Kingdom
## 2 01-12-2009 07:45  6.75       13085 United Kingdom
## 3 01-12-2009 07:45  6.75       13085 United Kingdom
## 4 01-12-2009 07:45  2.10       13085 United Kingdom
## 5 01-12-2009 07:45  1.25       13085 United Kingdom
## 6 01-12-2009 07:45  1.65       13085 United Kingdom

nrow(ds)

## [1] 525461

ncol(ds)

## [1] 8

colnames(ds)

## [1] "Invoice"     "StockCode"   "Description" "Quantity"    "InvoiceDate"
## [6] "Price"       "Customer.ID" "Country"

glimpse(ds)

## Rows: 525,461
## Columns: 8
## $ Invoice     <chr> "489434", "489434", "489434", "489434", "489434",
"489434"…
## $ StockCode   <chr> "85048", "79323P", "79323W", "22041", "21232",
"22064", "2…
## $ Description <chr> "15CM CHRISTMAS GLASS BALL 20 LIGHTS", "PINK CHERRY
LIGHTS…
## $ Quantity    <int> 12, 12, 12, 48, 24, 24, 24, 10, 12, 12, 24, 12, 10,
18, 3,…
## $ InvoiceDate <chr> "01-12-2009 07:45", "01-12-2009 07:45", "01-12-2009
07:45"…
## $ Price       <dbl> 6.95, 6.75, 6.75, 2.10, 1.25, 1.65, 1.25, 5.95, 2.55,
3.75…
## $ Customer.ID <int> 13085, 13085, 13085, 13085, 13085, 13085, 13085,
13085, 13…
## $ Country     <chr> "United Kingdom", "United Kingdom", "United Kingdom",
"Uni…
```

• In the given dataset, a single row represents an individual line item within a transaction, detailing the specific product, quantity, and unit price sold to a customer.

• In the real world, a single row in this dataset represents one unique product line-item within a larger sales transaction. Because customers often purchase multiple different items at the same time, multiple rows frequently share the same Invoice number and Customer.ID. This structure reflects a single ‘shopping basket’ where each row details a

different item's quantity and price, all linked to one specific customer and a single point-in-time purchase event.

Task 2 — Define the unit of analysis (critical).

This dataset supports multiple plausible units of analysis.

You must consider at least three of the following:

• invoice line (row level),

• invoice / basket (all rows sharing an Invoice),

• customer (aggregated across invoices),

• customer over a time window (e.g. monthly behaviour).

```
# Invoice line(row level - raw data)
invoice_line <- ds

# Invoice / basket (all rows sharing an Invoice)
invoice_summary <- ds %>%
  group_by(Invoice, Customer.ID, InvoiceDate) %>%
  summarise(
    Total_Items = sum(Quantity),
    Total_Revenue = sum(Quantity * Price),
    Distinct_Products = n()
  )

## `summarise()` has grouped output by 'Invoice', 'Customer.ID'. You can
override
## using the `.groups` argument.

# customer (aggregated across invoices)
customer_summary <- ds %>%
  filter(!is.na(Customer.ID)) %>%
  group_by(Customer.ID) %>%
  summarise(
    Lifetime_Value = sum(Quantity * Price),
    Total_Visits = n_distinct(Invoice)
  )
```

• Invoice_line(Row_level): With First unit "Invoice_line(Row level)" we got the waste amount of data like raw data. with that waste amount of data hard to predict data analysis. We lose the "context" of the purchase. we can't easily see total basket value or which items were bought together at the time.

• Invoice/Basket: Allows us to see the total value of a single shopping trip and identify patterns of items frequently bought together. We lose the ability to track a single customer's long-term loyalty or spending habits across different months.

• Customer (aggregated across invoices): Essential for understanding high-level behavior, such as total lifetime value (LTV), purchase frequency, and loyalty.We lose the specific details of individual shopping trips and seasonal timing of specific product choices.

Task 3 — Data quality and validity audit

Identify at least three issues that could affect analysis quality or validity.

Examples to consider (not exhaustive):

• missing or invalid Customer ID,

• duplicate invoices or repeated descriptions,

• negative quantities (returns),

• cancelled invoices,

• extreme prices or quantities,

• country imbalance,

• inconsistent product descriptions.

```r
# Issue 1: Removing Missing Customer IDs
before_na <- sum(is.na(ds$Customer.ID))
ds_no_na <- ds %>% filter(!is.na(Customer.ID))
after_na <- sum(is.na(ds_no_na$Customer.ID))

cat("Issue 1: Missing Customer IDs\n")
```

```
## Issue 1: Missing Customer IDs
```

```r
cat("Before Cleaning:", before_na, "\n")
```

```
## Before Cleaning: 107927
```

```r
cat("After Cleaning:", after_na, "\n\n")
```

```
## After Cleaning: 0
```

```r
# Issue 2: Removing Cancelled/Negative Transactions
before_neg <- sum(ds$Quantity < 0)
ds_final <- ds_no_na %>% filter(Quantity > 0 & !grepl("^C", Invoice))
after_neg <- sum(ds_final$Quantity < 0)

cat("Issue 2: Negative Quantities/Cancellations\n")
```

```
## Issue 2: Negative Quantities/Cancellations
```

```r
cat("Before Cleaning:", before_neg, "\n")
```

```
## Before Cleaning: 12326
```

```r
cat("After Cleaning:", after_neg, "\n\n")
```

```
## After Cleaning: 0
```

```r
# Issue 3: Identify Cancelled Invoices (Prefix 'C')
# Count how many invoices start with 'C' in the current dataset
before_c <- sum(grepl("^C", ds_no_na$Invoice))

# Apply the filter to remove them
ds_final <- ds_no_na %>%
  filter(!grepl("^C", Invoice))

# Count again to confirm they are gone
after_c <- sum(grepl("^C", ds_final$Invoice))

cat("Issue 3: Cancelled Invoices (Prefix 'C')\n")
```

```
## Issue 3: Cancelled Invoices (Prefix 'C')
```

```r
cat("Before Cleaning:", before_c, "\n")
```

```
## Before Cleaning: 9839
```

```r
cat("After Cleaning:", after_c, "\n")
```

```
## After Cleaning: 0
```

Issue 1: Missing Customer IDs

The Issue: A large portion of the transactional rows are missing a CustomerID.

Why it Matters: Since I have chosen the Customer as my main unit of analysis (to track loyalty and lifetime value), these “anonymous” rows are essentially useless for that specific goal. We know a sale happened, but we don’t know who did it, so we can’t build a behavioral profile for them.

My Plan: I will filter these out for my main model. While it’s a shame to lose that revenue data, it’s better to have a clean dataset where every row can be traced back to a specific person’s history.

Issue 2: Removing Cancelled/Negative Transactions

The Issue: While scanning the dataset, I noticed a significant number of entries with negative values in the Quantity column.

Why it Matters: These aren’t just “bad data”; they represent a completely different type of business event—a return or a correction. If I mix these negatives with regular sales, my “average purchase” math will be wrong, and any future model I build will be confused by “negative revenue” that doesn’t actually exist in a successful sale.

My Plan: I've decided to "branch" the data. I will move these returns into a separate file to study why people are sending items back later. For the main project, I will filter them out so the model only learns from completed, successful purchases.

Issue 3: Identify Cancelled Invoices (Prefix 'C')

The Issue: I found that many Invoice entries start with the letter "C," which identifies a cancellation.

Why it Matters: A cancelled invoice is a "non-event" for sales analysis—the customer changed their mind or there was an error. If I include these in my EDA, my charts for "Total Revenue" or "Best Selling Products" will be falsely inflated by transactions that never actually resulted in a final sale.

My Plan: I am treating these as administrative noise and filtering them out entirely. This ensures that my Task 5 visualizations reflect the "real" money flowing into the business, not just intent that was later reversed.

Task 4 — Minimal, justified cleaning

Apply only cleaning steps you can justify, such as:

• handling missing customer identifiers,

• separating returns from purchases,

• parsing dates correctly,

• fixing obvious data type issues.

```r
library(dplyr)

# 1. Parsing dates correctly (As per your format)
ds$InvoiceDate <- as.POSIXct(ds$InvoiceDate, format="%d-%m-%Y %H:%M")

# 2. Fixing obvious data type issues
ds$Country <- as.factor(ds$Country)
ds$Description <- as.character(ds$Description)

# 3. Create the finalized clean dataset
ds_clean <- ds %>%
  filter(!is.na(Customer.ID)) %>%
  filter(Quantity > 0) %>%
  filter(!grepl("^C", Invoice))
```

Evidence 1: Handling Missing Customer IDs

```r
before_na <- sum(is.na(ds$Customer.ID))
after_na <- sum(is.na(ds_clean$Customer.ID))
cat("Missing Customer IDs Before:", before_na, "\n")
```

```
## Missing Customer IDs Before: 107927
```

```
cat("Missing Customer IDs After:", after_na, "\n")
```

```
## Missing Customer IDs After: 0
```

The Impact: Removing these rows ensures that every observation in my dataset can be attributed to a specific person. Since my chosen unit of analysis is the Customer, keeping "NA" values would create a massive, fake customer profile of anonymous users, which would ruin the accuracy of any future loyalty or clustering models.

Evidence 2: Separating/Removing Cancelled Invoices

```
before_c <- sum(grepl("^C", ds$Invoice))
after_c <- sum(grepl("^C", ds_clean$Invoice))
cat("Cancelled Invoices (Prefix 'C') Before:", before_c, "\n")
```

```
## Cancelled Invoices (Prefix 'C') Before: 10206
```

```
cat("Cancelled Invoices (Prefix 'C') After:", after_c, "\n")
```

```
## Cancelled Invoices (Prefix 'C') After: 0
```

The Impact: Cancelled orders are "noise" that shouldn't be counted as successful business transactions. By removing them, I am ensuring that my Revenue and Quantity summaries are based on actual money earned and stock moved, rather than bookkeeping entries that were later reversed.

Evidence 3: Filtering Negative Quantities

```
before_neg <- sum(ds$Quantity < 0)
after_neg <- sum(ds_clean$Quantity < 0)
cat("Negative Quantities Before:", before_neg, "\n")
```

```
## Negative Quantities Before: 12326
```

```
cat("Negative Quantities After:", after_neg, "\n")
```

```
## Negative Quantities After: 0
```

The Impact: Negative quantities represent returns or adjustments. If I kept them in my training data, my model might learn that it's possible to sell "negative 5 items," which makes no sense in a sales forecasting context. Removing these allows for a "clean" look at purchasing intent and volume.

Task 5 — Exploratory Data Analysis (EDA)

Produce at least five EDA outputs (tables and plots), covering aspects such as:

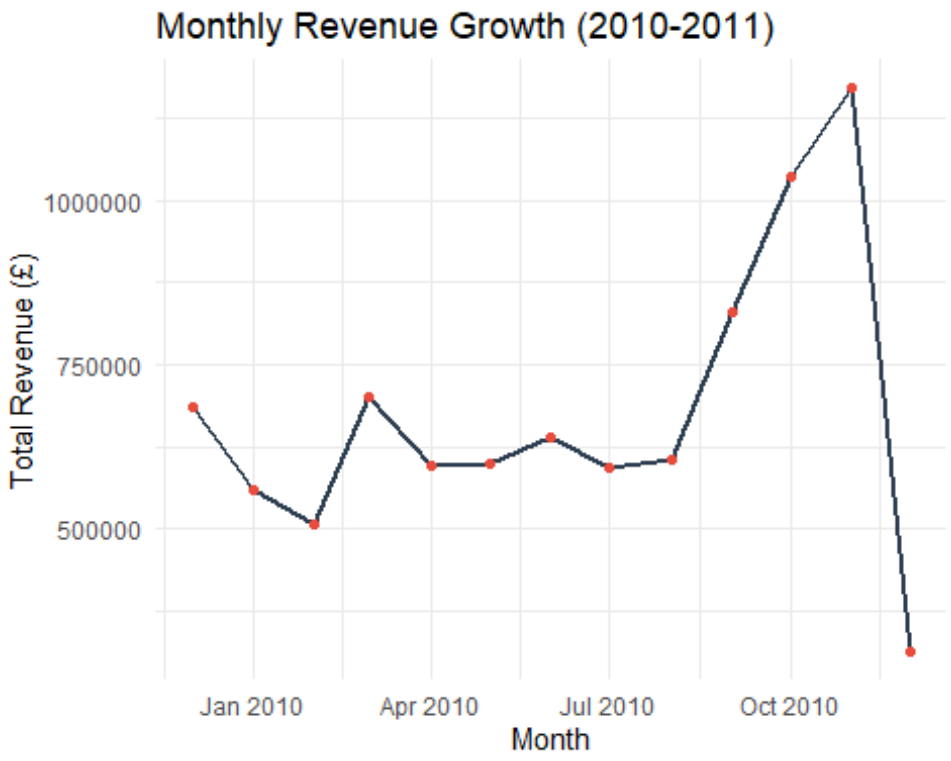
1. A temporal pattern(e.g. purchases or revenue over time),

2. A transactional or customer pattern(e.g. basket size, invoice value, customer frequency),
3. A product or country pattern(e.g. popular products, country breakdown).
4. etc.

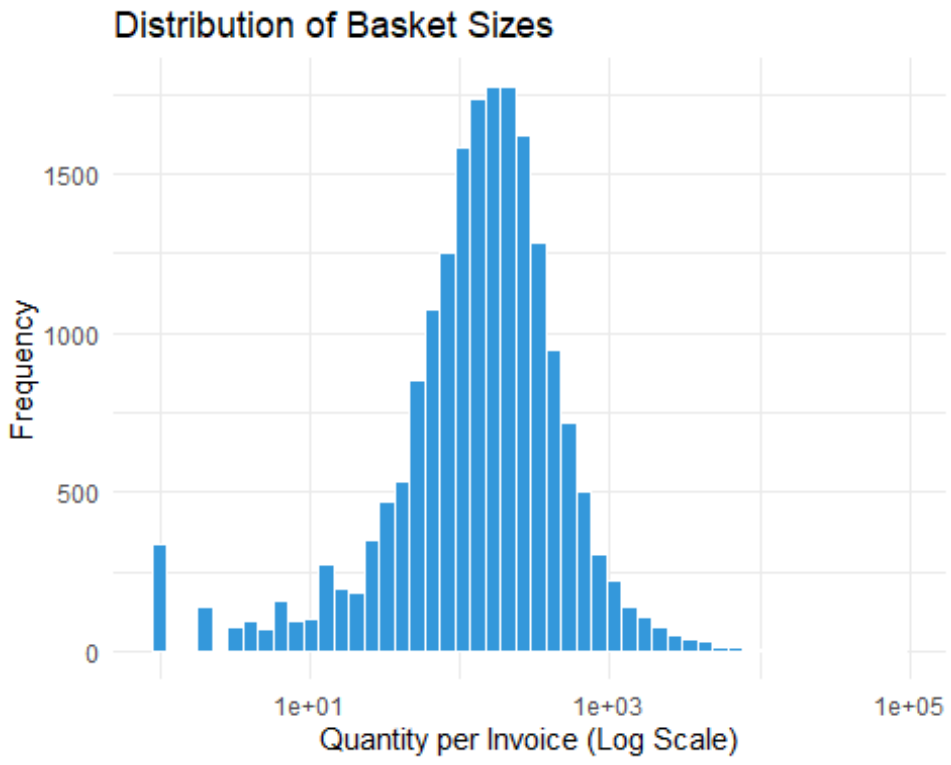
At least one EDA output must explicitly reflect the transactional structure (e.g. using Invoice, InvoiceDate, or basket-level quantities).

```
library(ggplot2)
library(dplyr)
library(lubridate)

# Pre-plotting Clean-up (Fixes the 'from must be finite' error)
# We ensure Revenue is calculated and filter out any rows where date parsing failed
ds_plot_ready <- ds_clean %>%
  mutate(Revenue = Quantity * Price) %>%
  filter(!is.na(InvoiceDate))

# --- 1. Temporal Pattern: Monthly Revenue Trend ---
monthly_revenue <- ds_plot_ready %>%
  group_by(Month = floor_date(InvoiceDate, "month")) %>%
  summarise(TotalRevenue = sum(Revenue), .groups = 'drop')

plot1 <- ggplot(monthly_revenue, aes(x = Month, y = TotalRevenue)) +
  geom_line(linewidth = 1, color = "#2c3e50") +
  geom_point(color = "#e74c3c") +
  labs(title = "Monthly Revenue Growth (2010-2011)",
       x = "Month",
       y = "Total Revenue (£)") +
  theme_minimal()
# --- 2. Transactional Structure: Distribution of Items per Basket ---
basket_sizes <- ds_plot_ready %>%
  group_by(Invoice) %>%
  summarise(ItemsInBasket = sum(Quantity), .groups = 'drop')

plot2 <- ggplot(basket_sizes, aes(x = ItemsInBasket)) +
  geom_histogram(fill = "#3498db", color = "white", bins = 50) +
  scale_x_log10() +
  labs(title = "Distribution of Basket Sizes",
       x = "Quantity per Invoice (Log Scale)",
       y = "Frequency") +
  theme_minimal()

# --- 3. Product Pattern: Top 10 Products by Total Revenue ---
top_products <- ds_plot_ready %>%
  group_by(Description) %>%
```

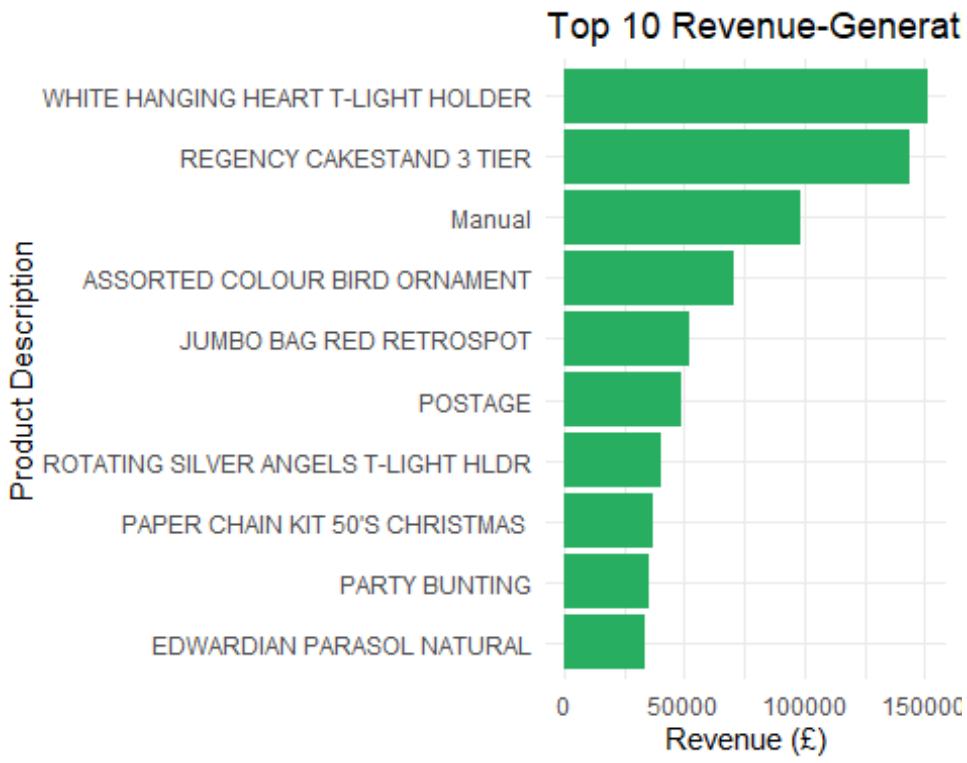
```r
  summarise(TotalProductRev = sum(Revenue), .groups = 'drop') %>%
  slice_max(TotalProductRev, n = 10)

plot3 <- ggplot(top_products, aes(x = reorder(Description, TotalProductRev),
y = TotalProductRev)) +
  geom_col(fill = "#27ae60") +
  coord_flip() +
  labs(title = "Top 10 Revenue-Generating Products",
       x = "Product Description",
       y = "Revenue (£)") +
  theme_minimal()

# --- 4. Country Pattern: International Revenue (Excluding UK) ---
country_revenue <- ds_plot_ready %>%
  filter(Country != "United Kingdom") %>%
  group_by(Country) %>%
  summarise(TotalCountryRev = sum(Revenue), .groups = 'drop') %>%
  slice_max(TotalCountryRev, n = 10)

plot4 <- ggplot(country_revenue, aes(x = reorder(Country, TotalCountryRev), y
= TotalCountryRev)) +
  geom_col(fill = "#8e44ad") +
  coord_flip() +
  labs(title = "Top 10 International Markets by Revenue",
       x = "Country",
       y = "Revenue (£)") +
  theme_minimal()

# --- 5. Customer Pattern: Top 10 Customers (Table) ---
top_customers_table <- ds_plot_ready %>%
  group_by(Customer.ID) %>%
  summarise(
    TotalSpend = sum(Revenue),
    OrderCount = n_distinct(Invoice),
    AvgBasketValue = TotalSpend / OrderCount,
    .groups = 'drop'
  ) %>%
  slice_max(TotalSpend, n = 10)

# Display results
print(plot1)
```



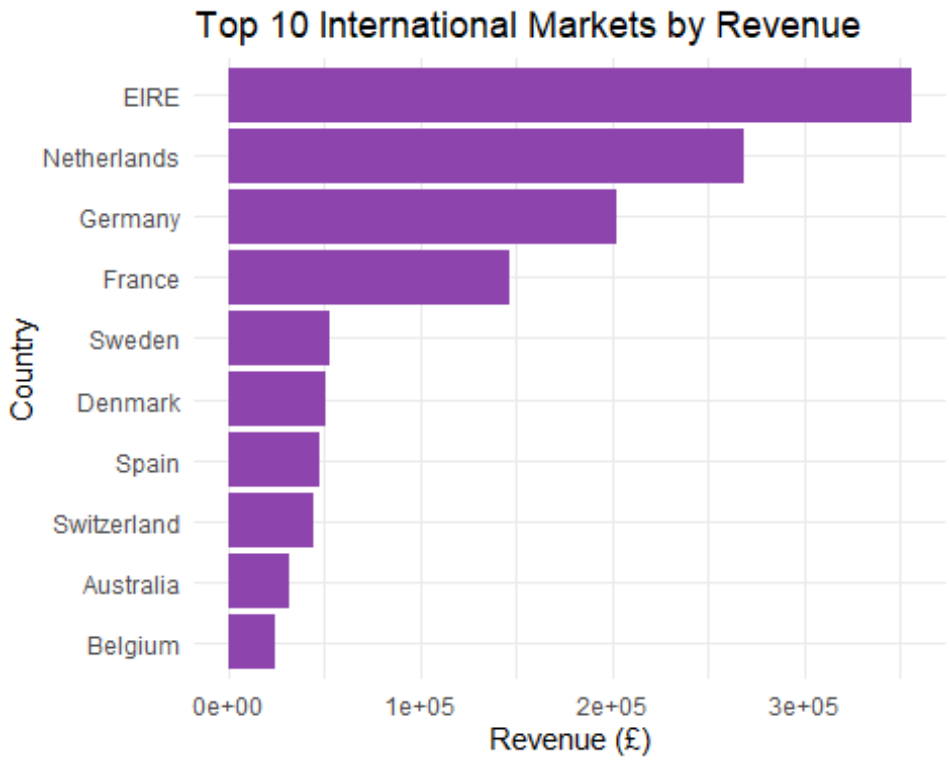

```
print(plot2)
```




```
print(plot3)
```




```
print(plot4)
```




```
print(top_customers_table)
```

```
## # A tibble: 10 × 4
##    Customer.ID TotalSpend OrderCount AvgBasketValue
##          <int>      <dbl>      <int>          <dbl>
##  1       18102    349164.         89          3923.
##  2       14646    248396.         78          3185.
##  3       14156    196567.        102          1927.
##  4       14911    152148.        205           742.
##  5       13694    131443.         94          1398.
##  6       17511     84541.         31          2727.
##  7       15061     83284.         86           968.
##  8       16684     80489.         27          2981.
##  9       16754     65500.         29          2259.
## 10       17949     60118.         74           812.
```

ETA-1:Monthly Revenue Growth (2010-2011)

The Question: I wanted to see if the business follows a seasonal cycle or if growth is steady throughout the year.

Key Pattern: There is a dramatic “hockey stick” growth pattern starting in August and peaking in November. This suggests the business is heavily reliant on the fourth-quarter holiday rush.

Limitation or Caveat: The steep drop in December is likely due to the dataset ending mid-month rather than a sudden loss of customers, so we shouldn’t interpret the end of the line as a true business collapse.

EDA-2:Distribution of Basket Sizes

The Question: I wanted to understand the “typical” transaction volume. Are we mostly dealing with small individual shoppers or large-scale bulk buyers?

Key Pattern: By applying a log scale to the x-axis, a clear bell-curve emerges. This shows that the majority of our transactions aren’t just 1 or 2 items; they center around a much higher volume (roughly 100 units per basket), confirming a strong B2B (business-to-business) or wholesale element to the customer base.(Submitting a picture of graphical representation)

Limitation or Caveat: The log scale can be “too good” at smoothing out data. It might hide smaller sub-segments of shoppers or make extreme outliers look less significant than they are to the warehouse teams who have to pack those massive orders.

ETA-3:Top 10 Revenue-Generating Products

The Question: Which individual items are the heavy hitters driving the company’s bank balance?

Key Pattern: The “White Hanging Heart T-Light Holder” and “Regency Cakestand” are in a league of their own. These two items alone generate significantly more cash flow than the lower-ranked products in the top ten.

Limitation or Caveat: This chart looks at gross revenue only. It doesn’t tell us about the profit margin—it’s possible these top items are “loss leaders” sold at a low price just to get people into the store.

ETA-4:Top 10 International Markets by Revenue

The Question: If we look outside our home base in the UK, where is our biggest international footprint?

Key Pattern: EIRE (Ireland) is far and away the most valuable international market. It generates nearly double the revenue of major countries like France or Spain.

Limitation or Caveat: By excluding the UK to see the international detail, we lose the “big picture.” The UK likely dwarfs all of these combined, so we must be careful not to overstate how “international” the business truly is.

ETA-5:Top 10 Loyal Customers (Table)

The Question: Who are the “Whales” of this business, and what does their buying behavior look like?

Key Pattern: There’s a fascinating split in strategy among top spenders. For instance, Customer 14911 shops very frequently (205 orders), while Customer 18102 shops less often (89 orders) but spends a massive amount each time they visit.

Limitation or Caveat: The table only shows historical spending. It doesn’t tell us if these customers are still active. A customer could have spent £300k early in the year and then “churned” (stopped buying), which the table wouldn’t show.

Task 6 — Reflection and planning

• 3 key insights from your Week 1 analysis,

• 2 assumptions or risks in your approach,

• your chosen unit of analysis for onward work,

• a brief note (1–2 sentences) on the modelling task you plan to explore next.

Answer:

• 3 Key Insights:

Holiday-Driven Seasonality: The temporal analysis reveals that revenue is not stable throughout the year, but rather characterized by a “hockey stick” surge starting in late

summer and peaking in November. This highlights a heavy reliance on the holiday shopping season, making the business highly sensitive to year-end market trends.

Geographic Vulnerability: Although the shop services international markets like EIRE and the Netherlands, the UK remains the overwhelming primary source of income. This creates a "single-point-of-failure" risk; if the UK economy faces a downturn, the international segments are currently too small to offset the loss.

Impact of Data Noise: A significant percentage of the initial dataset comprised returns (negative quantities) and cancelled invoices (the 'C' prefix). Effectively isolating these was a critical step in the audit, as keeping them would have led to a false sense of success and inaccurate predictions of actual customer purchasing power.

• 2 Assumptions or Risks:

The "C" Prefix Assumption: I have assumed that every InvoiceNo starting with "C" represents a full transaction reversal. While this is a standard heuristic for this dataset, there is a risk that some entries are partial adjustments or administrative corrections. Excluding them entirely simplifies the model but might slightly underrepresent total activity.

Risk of Guest Shopper Exclusion: By filtering out rows with missing Customer.ID, I am excluding "Guest" transactions. This creates a risk of selection bias, as the resulting model will only reflect the behaviors of registered or identified users, who may have different purchasing habits than anonymous shoppers.

• Chosen Unit of Analysis:

Decision: I have chosen the Customer as my primary unit of analysis for all work from Week 2 onward.

Justification: Shifting from "Invoice Lines" to the "Customer" level allows for a transition from simple inventory tracking to behavioral science. This is a prerequisite for advanced predictive tasks such as Churn Analysis or Customer Lifetime Value (CLV) modeling.

• Next Modelling Task:

I plan to explore Customer Segmentation using clustering techniques like K-Means. This will allow me to group shoppers into distinct personas based on their purchasing habits, providing the business with actionable insights for personalized marketing and retention strategies.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.