Data Generation Process

The code simulates a realistic retail transactional dataset for the fashion industry. It incorporates multiple dimensions such as product details, pricing, customer behavior, fulfillment, and even anomaly injections. The process is modular and broken down into several steps, each of which contributes to creating a rich and realistic dataset.

1. Data Setup

Product Lines and Hierarchies

- **Categories & Brands:**
  The process begins by defining a dictionary of product categories (e.g., *Apparel*, *Footwear*, *Accessories*, *Beauty & Personal Care*). Each category includes:
  - A list of associated brands (e.g., *Zara*, *H&M* for Apparel).
  - A list of detailed merchandise hierarchies (e.g., Apparel.Men.Shirts.Casual).
- **Product Code Mapping:**
  A unique product code is assigned to every combination of brand and hierarchy. The code follows a format like P-0001, P-0002, and so on.

Sales Channels and Promotional Codes

- **Sales Channels:**
  A list of realistic sales channels is provided, including physical retail stores (e.g., *Mall Store*, *Outlet Store*), online platforms (e.g., *Website*, *Mobile App*), and B2B channels.
- **Promo Codes:**
  A set of promo codes (e.g., FREE10, PARTY10) is defined to simulate different discount scenarios.

Pricing, Costs, and Demand Modeling

- **Price Ranges and Unit Cost Ratios:**
  Each category has specified price ranges and corresponding unit cost ratios to ensure that the generated prices and costs are realistic.

- **Quantity Generation:**
  Order quantities are modeled using a Poisson distribution with category-specific lambda values (e.g., 1.5 for Apparel) to simulate customer purchasing behavior.
- **Product Weights:**
  For fulfillment cost calculations, product weights are generated using random uniform distributions tailored by category.
- **Return Rates:**
  Fixed return rates are set per category, influencing the cost calculations in case of product returns.

Geographic and Shipping Details

- **Location Mapping:**
  A mapping is defined that includes countries (USA, France, Germany, UK, Italy), along with corresponding states and cities.
- **Shipping Parameters:**
  Shipping rules include a free shipping threshold and a fee per unit, influencing the overall sales and profitability.

2. Core Functions

Discount Calculation

- **Purpose:**
  The calculate_discount function computes the discount amount for an order.
- **Method:**
  It uses:
  - A base discount determined by the promo code.
  - Multipliers based on the sales channel, product category, and customer loyalty.
- **Constraint:**
  The discount is capped so it does not exceed the total sales amount.

Fulfillment Cost Calculation

- **Purpose:**
  The calculate_fulfillment_cost function estimates the cost of fulfilling an order.
- **Method:**
  It considers:
  - Quantity ordered.
  - Sales channel-specific cost factors.
  - Territory multipliers (e.g., NA vs. EMEA).
  - A weight factor based on the product's weight.

Data Generation Function: generate_fashion_data_with_brand

- **Date Range Iteration:**
  A date range is generated between the specified start and end dates. A base demand is set for each day (with higher demand on specific days, like weekends).
- **Order Generation:**
  For every date and for each product (brand and hierarchy):
  - The function simulates a number of orders equal to the base demand.
  - It randomly determines various order attributes such as:
    - **Price:**
      Calculated using a lognormal distribution and bounded within pre-defined price ranges.
    - **Quantity:**
      Generated using a Poisson distribution.
    - **Sales:**
      Derived as the product of price and quantity.

        - **Discount & Net Sales:**
          Discount is calculated using the dedicated function and net sales are computed after discount.
        - **Costs:**
          Unit cost, fulfillment cost, marketing cost, return cost, and cost of goods sold are calculated.
        - **Profit Calculation:**
          Profit and profit margins are computed after accounting for all expenses and shipping revenue.
        - **Customer and Order Metadata:**
          Additional details such as order number, order status, geographic information, customer names, and deal sizes are generated.
- **Output:**
  All simulated records are compiled into a Pandas DataFrame.

Anomaly Injection: inject_anomalies_by_date

- **Purpose:**
  This function is designed to introduce anomalies into the dataset to simulate irregular or abnormal events for analysis and testing of anomaly detection systems.
- **How It Works:**
    - **Anomaly Schedule:**
      A schedule (mapping dates to anomaly details) specifies the type, severity, root cause, and scope (sales channel or merchandise hierarchy) of the anomaly.
    - **Types of Anomalies:**
      The function can simulate:
        - **ExcessiveDiscount:**
          Increases discounts and recalculates net sales and return cost.

        - **COGSOverstatement:**
          Inflates unit cost and, therefore, the cost of goods sold.
        - **FulfillmentSpike:**
          Increases fulfillment costs.
        - **ShippingDisruption:**
          Alters shipping revenue.
        - **ReturnSurge:**
          Elevates return costs.
    - **Recalculation:**
      After applying the anomaly, it recalculates profit and profit margins to ensure consistency across the affected records.

Summary

The data generation process is a comprehensive simulation that combines:

- **Static mappings** (product hierarchies, geographic locations, promo codes) with
- **Randomized elements** (pricing, quantities, costs) and
- **Dynamic adjustments** (discounts, anomalies) to create a realistic retail sales dataset.

This synthetic dataset is ideal for testing various analytical models, such as those used in sales forecasting, anomaly detection, and profitability analysis.