

SWIFT Assignment - Transit Performance Analysis

You are working for a logistics analytics company that helps courier partners optimize their transit networks and improve operational efficiency. The dataset provided contains detailed tracking information about shipments moving through a courier logistics network. Your task is to analyze the transit performance by examining how shipments move through various facilities, calculate transit efficiency metrics, and generate a summary of the dataset given to you.

Assignment Tasks

Part 1: Load and Explore Data

1. Download the data from the provided source
2. Upload this into a workspace/project directory (Jupyter/vanilla Python)
3. Read the JSON data into your environment
4. Perform initial data exploration to understand the structure

Part 2: Flatten and Extract Transit Data

Extract and flatten the following information for each shipment:

Shipment Identifiers:

- Tracking number (from `trackingNumber` field)
- Service type (from `service.type` field)
- Service description (from `service.description` field)
- Carrier code (from `carrierCode` field)

Weight and Package Information:

- Package weight (value and units)
- Packaging type

Location Information:

- Origin city, state, postal code
- Destination city, state, postal code

Transit Events:

For each shipment, extract ALL events from the `events` array:

- Event type (from `eventType` field - can be any value like IT, AR, DP, PU, OD, DL, etc.)
- Event timestamp (from `timestamp` field)
- Event description (from `eventDescription` field)
- Event location city (from `address.city` field)
- Event location state (from `address.stateOrProvinceCode` field)
- Event location postal code (from `address.postalCode` field)
- Arrival location type (from `arrivalLocation` field - can be any value)

Part 3: Compute Transit Performance Metrics

For each shipment, calculate the following derived metrics:

1. Facility Touchpoints:

- Total number of unique facilities visited (count distinct facilities from events where `arrivalLocation` field contains the substring "FACILITY")
- Number of events with specific event types (count events by `eventType` - you should identify which event types represent "in transit" vs "arrival" based on the data)
- List all unique event types found in the dataset

2. Transit Time Analysis:

- Total transit time in hours (from first pickup-type event to final delivery-type event - identify these by analyzing `eventType` and `eventDescription` fields)
- Time in inter-facility transit in hours (calculate based on event timestamps and facility touchpoints)

3. Transit Velocity:

- Average hours per facility (total transit time / number of facilities)
- Service category classification (classify based on `service.type` field - you should identify patterns like express vs standard services)

4. Delivery Characteristics:

- Final delivery location type (from `deliveryLocationType` field)
- Number of out-for-delivery attempts (count events where `eventType` or `eventDescription` indicates out-for-delivery status)
- Was delivered on first attempt (TRUE if only 1 out-for-delivery event, FALSE otherwise)

Part 4: Handle Edge Cases

Your solution should handle:

- Shipments with missing or null values in any field
- Timestamps in different formats (MongoDB `$numberLong` format vs ISO string format)
- Shipments with incomplete event sequences
- Missing address information (city, state, postal code)
- Duplicate events at the same timestamp
- Events array being empty or missing
- Nested fields that may not exist in all records

Part 5: Output Detailed Transit CSV

Create a CSV file named `transit_performance_detailed.csv` with the following columns:

- `tracking_number`
- `service_type`
- `carrier_code`
- `package_weight_kg`
- `packaging_type`
- `origin_city`, `origin_state`, `origin_pincode`
- `destination_city`, `destination_state`, `destination_pincode`
- `pickup_datetime_ist`, `delivery_datetime_ist`
- `total_transit_hours`
- `num_facilities_visited`
- `num_in_transit_events`

- time_in_inter_facility_transit_hours
- avg_hours_per_facility
- is_express_service
- delivery_location_type
- num_out_for_delivery_attempts
- first_attempt_delivery
- total_events_count

Part 6: Output Network Performance Summary CSV

Create a summary CSV file named `transit_performance_summary.csv` with the following statistics:

Overall Metrics:

- total_shipments_analyzed
- avg_transit_hours, median_transit_hours, std_dev_transit_hours
- min_transit_hours, max_transit_hours

Facility Metrics:

- avg_facilities_per_shipment, median_facilities_per_shipment
- mode_facilities_per_shipment
- avg_hours_per_facility, median_hours_per_facility

Service Type Comparison:

(Group by unique values found in `service.type` field)

- avg_transit_hours_by_service_type (for each unique service type)
- avg_facilities_by_service_type (for each unique service type)
- count_shipments_by_service_type (for each unique service type)

Delivery Performance:

- pct_first_attempt_delivery
- avg_out_for_delivery_attempts

We prefer you send the solution within **10 hours** of receiving the assignment regardless of the official deadline, unless of course you want to take more time and surprise us with

analytical wizardry.

Our judgement is based not only on the accuracy of the result but also how you have gone about getting to the result (i.e. the approach in the code) as well as quality of code, commenting practices and readability of code.

Your submission should include your code project (or ipynb) and the two resulting CSV files.

