Data Analytics Project: FoodExpress Platform

Questions developed for beginner to intermediate level data analytics practice

Case Study: FoodExpress

FoodExpress is a rapidly growing food delivery platform that connects customers with local restaurants and delivery partners. The platform operates across multiple cities and handles thousands of orders daily. The company uses data analytics to optimize delivery times, improve customer satisfaction, and maximize restaurant partnerships.

The provided CSV dataset represents one month of delivery operations, containing information such as:

- Order details (order ID, date, time)
- Customer information (customer segment, location)
- Restaurant details (restaurant type, cuisine)
- Delivery metrics (delivery time, distance, fees)
- Financial data (order value, delivery fee, commission)
- Performance indicators (customer ratings, delivery partner ratings)

Section 1: Python Fundamentals (3 Questions)

1. Python Loops: Total Revenue Calculation

Write a Python program using a loop to calculate the total revenue (sum of all `OrderValue` amounts) generated across all cities in the dataset.

Expected Output:

```
Total Revenue: $XXXXX.XX
```

2. Python Conditionals: Premium Delivery Identifier

Premium Delivery Classification

Write a Python function that determines whether a delivery is classified as "Premium Delivery". A delivery is considered "Premium" if:

- The delivery fee is greater than $5.00, AND
- The delivery time is less than 25 minutes

The function should take an `OrderID` as input and return `True` if it meets the criteria, or `False` otherwise.

Function Signature:

```
def is_premium_delivery(order_id):
    # Your code here
pass
```

3. Python Functions: Average Order Value by Cuisine

Write a Python function that takes a `CuisineType` (e.g., "Italian", "Chinese", "Fast Food") as input and returns the average order value for that cuisine type.

Function Signature:

```
def get_average_order_value(cuisine_type):
    # Your code here
pass
```

Example Usage:

```
avg_value = get_average_order_value("Italian") print(f"Average order
value for Italian cuisine: ${avg_value:.2f}")
```

Section 2: Data Cleaning and Reshaping Questions (7 Questions)

Use the uncleaned data provided for the following questions.

1. Handle Missing Values

Identify columns with missing values (e.g., `DeliveryTime`, `RestaurantRating`, `CustomerRating`, `TipAmount`).

- Fill missing **numerical** values with the column **mean**
- Replace missing **categorical** values with the **mode** (most frequent value)

Tasks:

- Print the number of missing values in each column before cleaning
- Apply the appropriate imputation method
- Verify that all missing values have been handled

2. Replace Unclean Data

Replace invalid values such as `99999` in the `DeliveryTime` column with the column **median**.

Rationale: Delivery times of 99999 minutes are clearly data entry errors and should be treated as outliers.

Tasks:

- Identify how many rows contain the value 99999
- Replace these with the median delivery time
- Verify the replacement

3. Clean String Values

Replace invalid strings like `"INVALID"` or `"N/A"` in the `CustomerRating` column with `NaN`, then handle these missing values by imputing the **median rating**.

Tasks:

- Identify all non-numeric values in the CustomerRating column
- Replace them with NaN
- Impute missing values with the median
- Verify the data type is numeric

4. Fix Data Types

Convert the following columns to appropriate data types:

- `OrderDate` → datetime
- `CustomerRating` → float (after cleaning)
- `RestaurantRating` → float (after cleaning)
- `OrderValue` → float
- `DeliveryFee` → float

Task: Ensure all columns have correct data types for analysis and calculations.

5. Remove Outliers

Detect and remove rows where the `DeliveryTime` is unusually high (e.g., greater than the 99th percentile) OR the `OrderValue` is unusually low (e.g., less than $5).

Tasks:

- Calculate the 99th percentile for DeliveryTime
- Identify orders with DeliveryTime > 99th percentile OR OrderValue < $5
- Remove these outlier rows
- Report how many rows were removed

6. Reshape Data Using Melt

Use the `melt()` function to reshape the payment data from wide to long format.

Scenario: The dataset has separate columns for different payment methods:

- `CashPayment`
- `CardPayment`
- `WalletPayment`
- `UPIPayment`

For each order, only one payment method column contains the OrderValue, while others are 0.

Task: Combine these columns into:

- A single `PaymentMethod` column
- A single `PaymentAmount` column
- Filter out rows where PaymentAmount = 0 (since each order uses only one payment method)

Example Before (from your dataset):

OrderID	CashPayment	CardPayment	WalletPayment	UPIPayment
5001	0	45.50	0	0
5002	32.80	0	0	0
5003	0	0	0	28.90

Example After:

OrderID	PaymentMethod	PaymentAmount
5001	CardPayment	45.50
5002	CashPayment	32.80
5003	UPIPayment	28.90

Hint: Use `pd.melt()` and filter for PaymentAmount > 0

7. Feature Engineering: Net Revenue

Add a new column called `NetRevenue` calculated as:

```
NetRevenue = OrderValue - (DiscountAmount + PromoDiscount) - RestaurantCommission
```

Where:

- `DiscountAmount` : Customer discount applied
- `PromoDiscount` : Promotional discount
- `RestaurantCommission` : Commission paid to the platform (typically 15-25% of OrderValue)

Task: Create this new feature and analyze which city generates the highest net revenue.

Section 3: Visualization Questions (6 Questions)

After cleaning the data, create the following visualizations:

1. Bar Chart: Total Orders by Cuisine Type

Create a bar chart showing the total number of orders for each cuisine type (`Italian` , `Chinese` , `Indian` , `Fast Food` , `Mexican` , etc.) across all cities.

Requirements:

- X-axis: Cuisine Type
- Y-axis: Total Orders
- Add title and axis labels
- Use different colors for each cuisine

2. Histogram: Delivery Time Distribution

Plot a histogram of the `DeliveryTime` column to analyze the distribution of delivery times after cleaning.

Requirements:

- Use appropriate bins (e.g., 20-30 bins)
- Add title: "Distribution of Delivery Times"
- Mark the mean and median with vertical lines
- Add a legend

3. Pie Chart: Customer Segment Proportion

Create a pie chart displaying the proportion of orders across customer segments:

- `New Customer`
- `Regular Customer`
- `VIP Customer`

Requirements:

- Show percentages on each slice
- Use contrasting colors
-

Add a title

4. Scatterplot: Order Value vs. Delivery Time

Generate a scatterplot to analyze the relationship between `OrderValue` and `DeliveryTime` .

Requirements:

- X-axis: Order Value
- Y-axis: Delivery Time
- Color points by `CustomerSegment`
- Add a trend line
- Include correlation coefficient in the title

5. Boxplot: Delivery Time by City

Create a boxplot to visualize the spread and outliers in `DeliveryTime` for each city.

Requirements:

- X-axis: City
- Y-axis: Delivery Time (minutes)
- Identify which city has the most consistent delivery times
- Highlight outliers

6. Line Chart: Daily Order Trends

Create a line chart showing daily order trends throughout the month for each cuisine category.

Requirements:

- X-axis: Date
- Y-axis: Number of Orders
- Multiple lines (one for each major cuisine type)
- Add legend
- Identify peak ordering days

Section 4: Statistical Test Questions (9 Questions)

T-Tests

1. Independent Samples T-Test: Premium vs. Regular Customers

Problem Statement:
FoodExpress wants to understand if customer segment affects ratings. Conduct an independent samples t-test to determine if there is a statistically significant difference in average customer ratings between `VIP Customers` and `Regular Customers` .

Requirements:

- **H_0** (Null Hypothesis): There is no difference in mean ratings between VIP and Regular customers
- **H_1** (Alternative Hypothesis): There is a significant difference in mean ratings
- Use α = 0.05 significance level
- Calculate t-statistic and p-value
- Interpret results in business context
- Create a visualization comparing rating distributions for both groups

2. One-Sample T-Test: Delivery Time Target

Problem Statement:
The operations team has set a target average delivery time of 30 minutes for the `North Zone` . Test whether the North Zone is meeting this target using a one-sample t-test.

Requirements:

- **H_0**: The mean delivery time in North Zone is 30 minutes (μ = 30)
- **H_1**: The mean delivery time in North Zone is NOT 30 minutes (μ ≠ 30)
- Use α = 0.05
- Calculate t-statistic and p-value
- Provide clear interpretation
- Recommend operational improvements if target is not met

Chi-Square Tests

3. Chi-Square Test of Independence: Payment Method and Customer Segment

Problem Statement:

The finance team wants to know if there's a relationship between payment methods (`Cash` , `Card` , `Wallet` , `UPI`) and customer segments (`New` , `Regular` , `VIP`). Since payment data is in wide format
(CashPayment, CardPayment, WalletPayment, UPIPayment columns), you'll first need to create a PaymentMethod column by identifying which payment column has a value > 0 for each order. Then conduct a chi-square test of independence.

Requirements:

- $\mathbf{H_0}$: Payment method and customer segment are independent
- $\mathbf{H_1}$: Payment method and customer segment are associated
- First, create a PaymentMethod column from the wide format payment data
- Create a contingency table
- Calculate expected frequencies
- Compute chi-square statistic and p-value
- Use α = 0.05
- Interpret practical significance

4. Chi-Square Goodness of Fit: Cuisine Distribution

Problem Statement:

The marketing team expects that orders should be evenly distributed across five major cuisines (`Italian` , `Chinese` , `Indian` , `Fast Food` , `Mexican`) - each accounting for 20% of total orders. Test if the actual distribution matches this expectation.

Requirements:

- $\mathbf{H_0}$: Orders are evenly distributed (20% each)
- $\mathbf{H_1}$: Orders are not evenly distributed
- Calculate observed and expected frequencies
- Compute chi-square statistic and p-value
- Use α = 0.05
- Provide recommendations for marketing focus

Correlation Analysis

5. Correlation Analysis: Order Value, Delivery Time, and Ratings

Problem Statement:

The analytics team wants to understand relationships between:

- Order Value and Customer Rating
- Delivery Time and Customer Rating
- Order Value and Delivery Time

Requirements:

- Calculate Pearson correlation coefficients for each pair
- Test statistical significance
- Create a correlation heatmap
- Generate scatterplots with trend lines
- Interpret findings and suggest operational improvements

6. Multiple Correlation: Restaurant Rating Prediction

Problem Statement:

Identify which factors most strongly correlate with `RestaurantRating`:

- Order Value
- Delivery Time
- Discount Percentage
- Number of Previous Orders (customer history)

Requirements:

- Calculate correlation coefficients for each factor
- Determine the strongest predictor
- Create visualizations
- Suggest actions for restaurant partnership team

ANOVA Tests

7. One-Way ANOVA: Delivery Time Across Cities

Problem Statement:

Test if there are significant differences in mean delivery times across the four cities (`North Zone` , `South Zone` , `East Zone` , `West Zone`).

Requirements:

- **H_0**: Mean delivery times are equal across all cities
- **H_1**: At least one city has a different mean delivery time
- Calculate F-statistic and p-value
- Use α = 0.05
- If significant, conduct post-hoc tests (Tukey HSD) to identify which cities differ
- Create boxplot comparing delivery times
- Provide operational recommendations

Paired T-Tests

8. Paired T-Test: Weekend vs. Weekday Performance

Problem Statement:

Compare delivery performance (average delivery time) on weekends vs. weekdays for the same delivery partners. Conduct a paired t-test to see if delivery partners perform differently on weekends.

Requirements:

- **H_0**: No difference in mean delivery times between weekdays and weekends
- **H_1**: Significant difference exists
- Prepare paired data (same delivery partner on both periods)
- Calculate mean difference and standard error
- Compute t-statistic and p-value
- Use α = 0.05
- Visualize changes for individual delivery partners
- Suggest scheduling optimizations

9. Paired T-Test: Before and After Promotional Campaign

Problem Statement:
FoodExpress ran a promotional campaign in Week 2 of the month. Compare average order values before (Week 1) and after (Week 3) the campaign for the same restaurants to measure campaign effectiveness.

Requirements:

- $\mathbf{H_0}$: No difference in average order values before and after campaign
- $\mathbf{H_1}$: Campaign significantly affected order values
- Prepare paired data (same restaurants in both periods)
- Calculate mean difference and its standard error
- Compute t-statistic and p-value
- Use α = 0.05
- Create visualization comparing order values by restaurant
- Provide ROI analysis and recommendations

Data Dictionary

Column Name	Description	Data Type
OrderID	Unique order identifier	Integer
OrderDate	Date of order	Date
OrderTime	Time of order placement	Time

Column Name	Description	Data Type
City	Delivery city/zone	Categorical
CustomerSegment	Customer classification	Categorical
CuisineType	Type of cuisine ordered	Categorical
RestaurantID	Unique restaurant identifier	Integer
RestaurantRating	Restaurant rating (1-5)	Float

OrderValue	Total order value in $	Float
DeliveryFee	Delivery charge in $	Float
DeliveryTime	Time taken for delivery (minutes)	Integer
DeliveryDistance	Delivery distance in km	Float
CustomerRating	Customer rating for order (1-5)	Float
DeliveryPartnerRating	Rating for delivery partner	Float
CashPayment	Order value if paid by cash, 0 otherwise	Float
CardPayment	Order value if paid by card, 0 otherwise	Float
WalletPayment	Order value if paid by wallet, 0 otherwise	Float
UPIPayment	Order value if paid by UPI, 0 otherwise	Float
DiscountAmount	Discount applied in $	Float
PromoDiscount	Promotional discount in $	Float
TipAmount	Tip given to delivery partner in $	Float
TimePeriod	Time of day category	Categorical
WeatherCondition	Weather during delivery	Categorical
DayOfWeek	Day of the week	Categorical

Submission Guidelines

1. **Code Quality:** Write clean, well-commented code
2. **Visualizations:** All plots should have titles, axis labels, and legends
3. **Interpretations:** Provide business insights for each statistical test
4. **Documentation:** Include a brief report summarizing key findings
5. **Notebook Organization:** Use markdown cells to separate sections clearly **Common Issues and Solutions:**

1. **Missing data in statistical tests:** Always check for and handle missing values before conducting tests
2. **Data type errors:** Ensure columns are in the correct format (datetime, float, categorical)
3. **Outliers affecting results:** Consider removing or transforming outliers before analysis

4. **Assumption violations:** Check normality and homogeneity assumptions for t-tests and ANOVA
5. **Multiple testing:** When conducting multiple tests, consider adjusting significance levels (e.g., Bonferroni correction)

Good Luck!