Pipeline Summary

1. **Data extraction:**

   The two datasets, one from "**restaurant_data.json**" and the other from "**comments.json**," are loaded using **pd.read_json().**

2. **Filtering:**
   - The restaurant data is filtered using a function called **filter_restaurants** from the **filter_functions** module. Restaurants in Barcelona open within the analyzed interval are selected, and their unique identifiers ('_id') are obtained.

   - The comments are filtered using the **filter_comments** function from the **filter_functions** module, using the selected restaurants as a basis.

3. **Auxiliar Database creation:**

   Several auxiliary databases are created to facilitate KPI calculations:
   - **bymonth_df**: Contains information per month with the 'stats' column unnested.
   - **timetables_df**: Each row represents a shift of a restaurant and the days it opens for that shift.
   - **rush_hours_df**: Contains rush hours and the composite time interval in which the restaurant opens each day.
   - **wine_rep_df**: Creates a 'wine_present' column indicating whether the term 'wine' appears in the review.

4. **KPI Calculation:**
   - Various KPIs are calculated using functions from the **kpi_calculation** module. These KPIs include averages (**avg_kp**i), sums (**sum_kpi**), open days (**open_day**s), occupancy rates (**occ_rates**), and repetitions of the concept "wine" in comments (**wine_repetitions**).
   - A results table (**kpi_result_table**) is then created, combining information from the restaurants and the calculated KPIs.

5. **Auxiliar and results database Dump creation:**

   The DataFrames obtained at each step are saved as csv files for further review or use. A function called **save_to_dump** from the **loading_and_dumping** module is utilized for this purpose

6. **Grouped KPI calculations:**

   The KPIs are grouped by different criteria, such as postal code (**postal_groups**), opening hours **(open_groups)**, and relevance of wine (**wine_groups**). These groups are calculated using functions from the **group_by** module.

7. **Loading results to excel:**

   The results of the groups obtained in the previous step are saved to Excel files using the **save_to_xls** function from the **loading_and_dumping** module.