Production Data Preparation Script

Description
The code in Prod_Data_Prep.py file processes simulation data from a text file (Prod_all_cases.txt) exported from PEREL SOFTWARE using export feature of production data (Cumulative Production) for selected runs and generates Excel files containing yearly production of oil, condensate, and gas production for each simulation run.

The script applies specific business rules:

- **Gas production (**Gas **column)** is set to **zero** for all years if the simulation run name **does not contain** "BDPRODUCERS".
- For runs **with** "BDPRODUCERS", the Gas column is:
    - **Zero for years 2024–2028**
    - **Calculated normally (yearly difference) for 2029 and later**.

Functions
read_simulation_data(file_path)

Purpose: Reads and parses raw simulation data from a text file.

Input:

- file_path (str): Path to the input text file (e.g., "Prod_all_cases.txt").

Output:

- data (dict): A dictionary where keys are simulation run names (formatted) and values are DataFrames containing:
    - Date (str): Date in DD-MMM-YYYY format.
    - Year (int): Extracted year.
    - GPT (float): Gas production total.
    - OPT (float): Oil production total.

Logic:

1. Identifies simulation runs using the "SUMMARY OF RUN:" marker.
2. Extracts production data for each run, skipping malformed entries.

---

calculate_yearly_changes_and_save(data)

Purpose: Computes yearly production changes and exports results to Excel.

Input:

- data (dict): The dictionary returned by read_simulation_data. **Output**:
- Excel files (one per simulation run) with columns:
    - Year (int): Adjusted to represent the year **before** the production change.
    - Oil (float): Yearly change in oil production (OPT.diff()).
    - Condensate (float): Placeholder column (all zeros).
    - Gas (float): Yearly gas production change (follows business rules).

Business Rules:

1. **Gas Production Handling**:
   - If the run name **contains** "BDPRODUCERS" **(case-insensitive)**:
     - Gas = 0 for years **2024–2028**.
     - Gas = GPT.diff() for **2029+**.
   - **Otherwise**, the entire Gas column is set to 0.
2. **Year Adjustment**:
   - The Year column is decremented by 1 to align production changes with the **prior year**.

Usage Example

python

```
file_path = "Prod_all_cases.txt"  # Raw simulation data
simulation_data = read_simulation_data(file_path)
calculate_yearly_changes_and_save(simulation_data)
```

Output Files:

- RunName_BDPRODUCERS.xlsx (Gas calculated for 2029+).
- RunName_Other.xlsx (Gas column entirely zero).

Dependencies

- Python 3.x
- Libraries:
  - pandas (for data manipulation).
  - re (for regex parsing).

Notes

- **Input File Format**: Expected to contain blocks of data starting with "SUMMARY OF RUN: RunName :" followed by production values.
- **Error Handling**: Skips rows with invalid data (non-numeric GPT/OPT).
- **Output Structure**: Each Excel file contains filtered columns (Year, Oil, Condensate, Gas) with no index.

This documentation ensures clarity for future maintenance and collaboration.