/\*\*

\* =============================================================================

\* @file tools\_data\_analysis.gs – v12.9

\* @description

\* وحدة متقدمة لتحليل وفهم بنية البيانات في Google Sheets.

\* توفر للمساعد "عيونًا" ليرى ويفهم الجداول التي يعمل عليها.

\* =============================================================================

\*/

/\*\*

\* 📄 [أداة] تُنشئ ملخصًا نصيًا ذكيًا للورقة النشطة أو المحددة.

\* تقوم بتحليل الأعمدة وأنواع البيانات فيها لتقديم رؤية شاملة.

\* @param {string} [sheetName] - اسم الورقة المراد تلخيصها (اختياري).

\* @returns {string} - ملخص نصي منسق بصيغة Markdown.

\*/

function getPageSummary(sheetName) {

try {

const ss = SpreadsheetApp.getActiveSpreadsheet();

const sheet = sheetName ? ss.getSheetByName(sheetName) : ss.getActiveSheet();

if (!sheet) return showError(`لم يتم العثور على الورقة "${sheetName}".`);

const range = sheet.getDataRange();

const values = range.getValues();

if (values.length === 0) return showMessage("info", `الورقة "${sheet.getName()}" فارغة.`);

const headers = values[0];

const rowCount = values.length - 1;

const colCount = headers.length;

// تحليل أنواع البيانات في الأعمدة

const columnTypes = headers.map((header, i) => {

const columnValues = values.slice(1).map(row => row[i]).filter(v => v !== null && v !== '');

if (columnValues.length === 0) return 'فارغ';

const isAllNumbers = columnValues.every(v => typeof v === 'number');

if (isAllNumbers) return 'رقمي';

const isAllDates = columnValues.every(v => v instanceof Date);

if (isAllDates) return 'تاريخ';

return 'نصي/مختلط';

});

let summary = `🧾 ملخص ورقة: \*${sheet.getName()}\*\n`;

summary += `\* \*\*حجم البيانات:\*\* ${rowCount} صف بيانات | ${colCount} عمود.\n\n`;

summary += `\*\*📌 الأعمدة وأنواع بياناتها:\*\*\n`;

headers.forEach((h, i) => {

summary += ` - \`${h}\` (عمود ${columnIndexToLetter(i+1)}): \*${columnTypes[i]}\*\n`;

});

return summary;

} catch (e) {

return showError(`فشل في إنشاء ملخص الورقة: ${e.message}`);

}

}

/\*\*

\* 🧾 [أداة] تُرجع وصفًا منظمًا للورقة الحالية بصيغة JSON.

\* مفيدة لتمرير بنية البيانات إلى أدوات أخرى أو للتحليل البرمجي.

\* @param {string} [sheetName] - اسم الورقة (اختياري).

\* @returns {string} - سلسلة JSON تحتوي على بنية الورقة وعينة بيانات.

\*/

function getSheetInsightAsJson(sheetName) {

try {

const ss = SpreadsheetApp.getActiveSpreadsheet();

const sheet = sheetName ? ss.getSheetByName(sheetName) : ss.getActiveSheet();

if (!sheet) throw new Error(`لم يتم العثور على الورقة "${sheetName}".`);

const name = sheet.getName();

const data = sheet.getDataRange().getValues();

if (data.length < 1) return JSON.stringify({ sheetName: name, message: "الورقة فارغة" }, null, 2);

const headers = data[0];

// استخدام المتغير من env.gs لتحديد عدد صفوف العينة

const sampleRows = data.slice(1, SAMPLE\_DATA\_ROW\_COUNT + 1);

const structure = {

sheetName: name,

rowCount: data.length - 1,

columnCount: headers.length,

columns: headers,

sampleData: sampleRows

};

return JSON.stringify(structure, null, 2);

} catch (e) {

return JSON.stringify({ error: "فشل توليد JSON", message: e.message }, null, 2);

}

}