



JavaScript

```
/**
 * Created by Francesco Cifardi
 * For issues or questions reach out on LinkedIn
 (https://www.linkedin.com/in/francescocifardi/)
 * Free to use and share, a mention on LinkedIn would be appreciated :- )
 * Free Google Ads Scripts Newsletter - No boring content, no regular emails,
 only scripts and other automation tools whenever they are ready.
 * https://medialauncher.beehiiv.com/subscribe
 *
 * Budget Adjustments Report v1.4 (Single Account) ©
 *
 * What the Script Does:
 * This script generates a comprehensive report of all your active Google Ads
 campaigns, focusing on key performance metrics and budget allocation. Here's
 what it provides:
 * 1. A complete list of all active campaigns across all types (Search, Display,
 Video, Shopping, Performance Max, etc.)
 * 2. Current daily budgets and actual spend
 * 3. Conversion data, including number of conversions and conversion value
 * 4. Performance ratios, such as percentage of total budget, cost, conversions,
 and conversion value for each campaign
 * 5. Proposed new budgets based on both conversion volume and conversion value
 *
 * How It Works:
 * 1. The script connects to your Google Ads account and retrieves data for all
 active campaigns from the past 30 days.
 * 2. It calculates total budget, spend, conversions, and conversion value
 across all campaigns.
 * 3. For each campaign, it computes what percentage of these totals the
 campaign represents.
 * 4. Based on these percentages, it calculates two proposed budgets:
 * a) One based on the campaign's share of total conversions
 * b) Another based on the campaign's share of total conversion value
 * 5. All this data is then populated into a Google Sheets report for easy
 viewing and analysis.
 *
```

* Visual Analysis Features:

* The report includes conditional formatting in columns G to J (percentage columns) using a yellow color scale. This visual aid helps you quickly identify outlier campaigns:

- * - Darker yellow indicates a higher percentage
- * - Lighter yellow or white indicates a lower percentage

*

* How to Use It:

* 1. First, make a copy of this template spreadsheet:

https://docs.google.com/spreadsheets/d/1sjQRP7yjp39QoXLR_8TRxJ-S4HggmbFM4NhhKmGqspQ/copy

* 2. In your Google Ads account, create a new script and paste the provided code.

* 3. Replace the spreadsheet URL in the script with the URL of your copy of the template.

* 4. If you want to filter campaigns by a label, specify the label name in the `labelName` variable. Leave it empty if you do not want to filter by label.

* (credit to Rares Zlavog for this addition

[<https://www.linkedin.com/in/rares-zlavog-ppc-b66b85113/>])

* 5. Run the script. It will populate the 'Report' sheet in your spreadsheet with the latest data.

* 6. Review the report, paying special attention to:

- * - Campaigns with high conversion value but low budget allocation
- * - Campaigns with high spend but low conversions or conversion value
- * - The proposed budgets in columns K and L

* 7. Use this data to inform your budget adjustment decisions. You might choose to:

- * - Increase budgets for high-performing campaigns that are limited by budget
- * - Decrease budgets for underperforming campaigns
- * - Investigate campaigns with unusual metrics to identify opportunities or issues

*/

```
function main() {  
    var scriptStartTime = new Date();  
    var today = new Date();  
    var last30Days = new Date(today.getTime() - 30 * 24 * 60 * 60 * 1000);  
    var spreadsheetUrl = 'YOUR_SHEET_URL'; // Replace with your Google Sheet  
    URL  
  
    var spreadsheet = SpreadsheetApp.openByUrl(spreadsheetUrl);  
    var sheet = spreadsheet.getSheetByName('Report');
```

```

// Specify the label name if you want to filter campaigns by label
var labelName = ""; // Example: "Your Label Name"

// Check if the sheet exists, if not, create it
if (sheet == null) {
    sheet = spreadsheet.insertSheet('Report');
    Logger.log('Created new sheet named "Report"');
} else {
    // Clear only the content of the sheet, preserving formatting
    var lastRow = Math.max(sheet.getLastRow(), 1);
    var lastColumn = Math.max(sheet.getLastColumn(), 1);
    if (lastRow > 1) {
        sheet.getRange(2, 1, lastRow - 1, lastColumn).clearContent();
    }
    Logger.log('Cleared existing content in "Report" sheet, preserving
formatting');
}

// Define headers
var headers = [
    'Campaign Name', 'Campaign Type', 'Current Daily Budget', 'Cost',
    'Conversions', 'Conversion Value',
    '% of Total Daily Budget', '% of Total Cost', '% of Total Conversions',
    '% of Total Conv Value',
    'Proposed New Budget (Conversions)', 'Proposed New Budget (Conv
value)',
    'Start Date', 'End Date', 'Script Run Time'
];

// Check if headers already exist, if not, add them
if (sheet.getRange("A1").getValue() !== headers[0]) {
    sheet.getRange(1, 1, 1, headers.length).setValues([headers]);
    Logger.log('Added headers to the sheet');
}

// Define query elements
var metrics = 'metrics.cost_micros, metrics.conversions_value,
metrics.conversions, metrics.impressions, metrics.clicks';
var campaignFields = 'campaign.name, campaign.id,
campaign_budget.amount_micros, campaign.advertising_channel_type';
var dateCondition = 'segments.date DURING LAST_30_DAYS';
var enabledCondition = 'campaign.status = "ENABLED"';
var labelResourceName = labelName ? getLabelResourceByName(labelName) :
null;

```

```

var query;
if (labelResourceName) {
    var campaignIds = getCampaignIdsByLabel(labelResourceName);
    if (campaignIds.length > 0) {
        query = `
            SELECT ${campaignFields}, ${metrics}
            FROM campaign
            WHERE ${dateCondition} AND ${enabledCondition}
            AND campaign.id IN (${campaignIds.join(',')})
            ORDER BY campaign.name
        `;
    } else {
        Logger.log('No campaigns found with the specified label.');
```

return;

```
    }
} else {
    query = `
        SELECT ${campaignFields}, ${metrics}
        FROM campaign
        WHERE ${dateCondition} AND ${enabledCondition}
        ORDER BY campaign.name
    `;
}

// Run the report and get the data
var report = AdsApp.report(query);
var rows = report.rows();

var totalDailyBudget = 0;
var totalConvValue = 0;
var totalCost = 0;
var totalConversions = 0;
var campaignData = [];

// Process the report data
while (rows.hasNext()) {
    var row = rows.next();
    var campaignName = row['campaign.name'];
    var campaignType = row['campaign.advertising_channel_type'];
    var dailyBudget = row['campaign_budget.amount_micros'] / 1000000; //
Convert micros to actual currency
    var cost = row['metrics.cost_micros'] / 1000000; // Convert micros to
actual currency

```

```

var conversions = parseFloat(row['metrics.conversions']);
var convValue = parseFloat(row['metrics.conversions_value']);

totalDailyBudget += dailyBudget;
totalConvValue += convValue;
totalCost += cost;
totalConversions += conversions;

campaignData.push({
    name: campaignName,
    type: campaignType,
    dailyBudget: dailyBudget,
    cost: cost,
    conversions: conversions,
    convValue: convValue
});
}

// Calculate proposed budgets and percentages
var dataToUpdate = [];
for (var i = 0; i < campaignData.length; i++) {
    var data = campaignData[i];
    var percentOfTotalBudget = data.dailyBudget / totalDailyBudget;
    var percentOfTotalCost = data.cost / totalCost;
    var percentOfTotalConversions = data.conversions / totalConversions;
    var percentOfTotalConvValue = data.convValue / totalConvValue;
    var proposedBudgetConversions = totalDailyBudget * (data.conversions /
totalConversions);
    var proposedBudgetConvValue = totalDailyBudget * (data.convValue /
totalConvValue);

    dataToUpdate.push([
        data.name,
        data.type,
        data.dailyBudget,
        data.cost,
        data.conversions,
        data.convValue,
        percentOfTotalBudget,
        percentOfTotalCost,
        percentOfTotalConversions,
        percentOfTotalConvValue,
        proposedBudgetConversions,
        proposedBudgetConvValue,

```

```

        formatDate(last30Days), // Start Date
        formatDate(today), // End Date
        formatDateTime(scriptStartTime)
    ]);

    Logger.log('Campaign: ' + data.name + ' (' + data.type + ')');
}

// Update the Google Sheet
if (dataToUpdate.length > 0) {
    sheet.getRange(2, 1, dataToUpdate.length,
dataToUpdate[0].length).setValues(dataToUpdate);
}

    Logger.log('Report spreadsheet updated: ' + spreadsheetUrl);
}

// Helper functions
function formatPercentage(value) {
    return value; // Return raw value, formatting applied in sheet
}

function formatDate(date) {
    return Utilities.formatDate(date, AdsApp.currentAccount().getTimeZone(),
'yyyyMMdd');
}

function formatDateTime(date) {
    return Utilities.formatDate(date, AdsApp.currentAccount().getTimeZone(),
'yyyy-MM-dd HH:mm:ss');
}

function getLabelResourceByName(labelName) {
    var query = `SELECT label.resource_name FROM label WHERE label.name =
'${labelName}'`;
    var report = AdsApp.report(query);
    var rows = report.rows();
    if (rows.hasNext()) {
        return rows.next()['label.resource_name'];
    }
    return null;
}

function getCampaignIdsByLabel(labelResourceName) {

```

```
var query = `
SELECT campaign.id
FROM campaign
WHERE campaign.labels CONTAINS ANY('${labelResourceName}')
`;
var report = AdsApp.report(query);
var rows = report.rows();
var campaignIds = [];
while (rows.hasNext()) {
    campaignIds.push(rows.next()['campaign.id']);
}
return campaignIds;
}
```