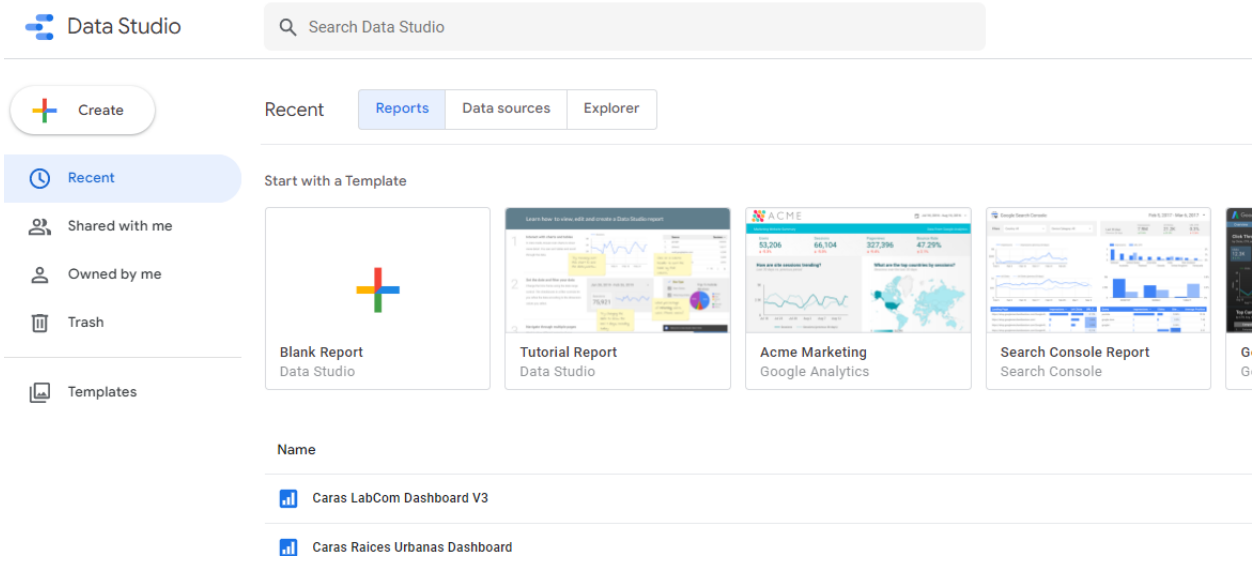
Help Documentation

Table of Contents

Dashboard Feature Documentation

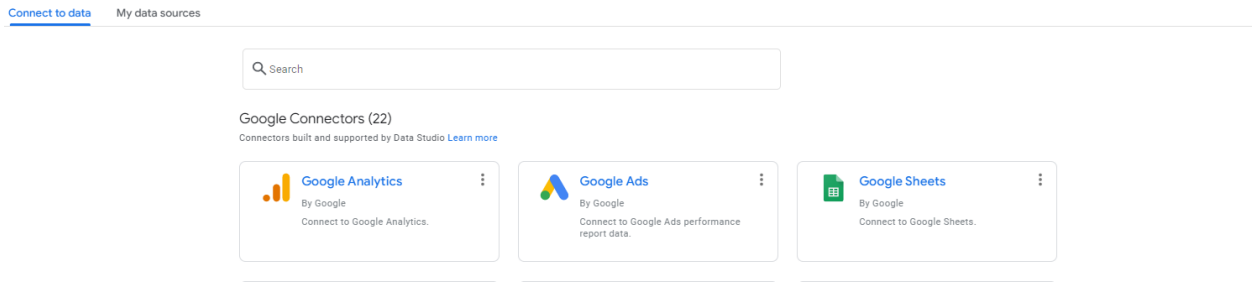
How to Access Data Studio

Login to your Google account and visit the Data Studio website. There you can find “reports” and “Data sources” tabs. The “reports” tab contains the dashboard and the “Data sources” tab contains all the Sheets that the dashboard pulls data from. Double click on the report to access the desired dashboard.

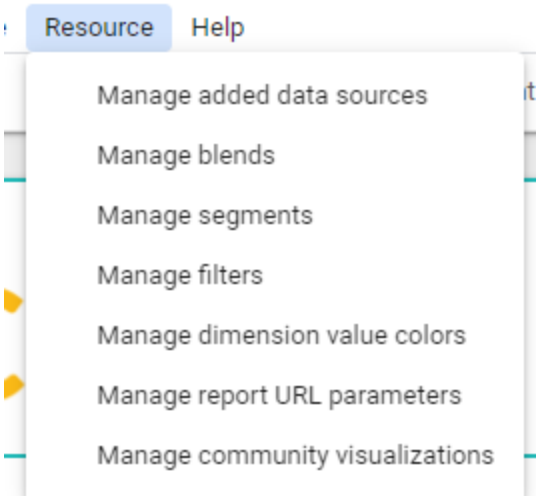


Linking Data Studio to Sheets

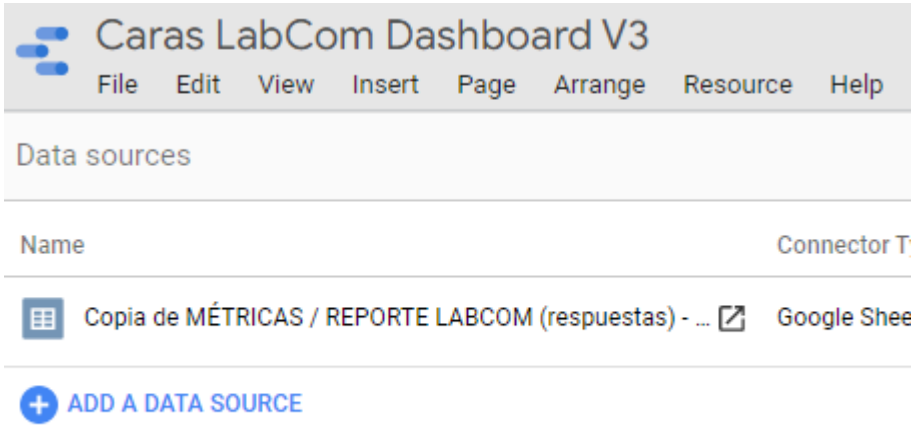
When creating a new report, you are prompted to use a Google connector to access your data source. When we created the LabCom dashboard we selected the “Google Sheets” option, which allowed us to link directly to the LabCom Google Sheet.



After this initial setup, you have the ability to link additional data sources to your dashboard. To accomplish this, navigate to the top of the page and click the “Resource” tab.



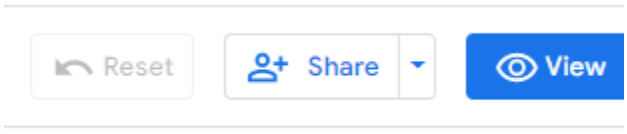
Then click “Manage added data sources” and then click “Add a Data Source”. This will bring up the same options as described above when initially creating a new report.



.

Sharing Access to Data Studio Dashboard

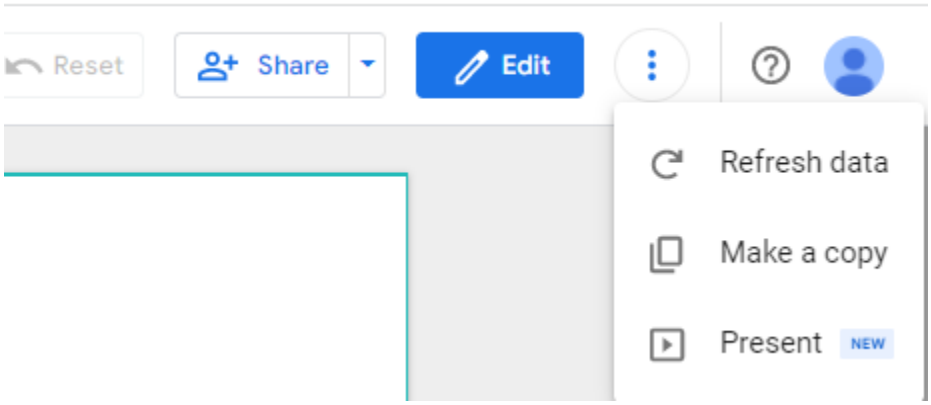
To share the dashboard with another Google account, navigate to the top right of the screen and click the share button. This will allow you to type in other accounts and select what permissions they should have.



Presenting the Dashboard

Similar to Powerpoint or Google Slides, you can present the dashboard. To do this, click the three dots in the top right of the page and click the present button. While in presentation mode,

your entire screen is filled by the dashboard, and you can still hover over visualizations to view their data.

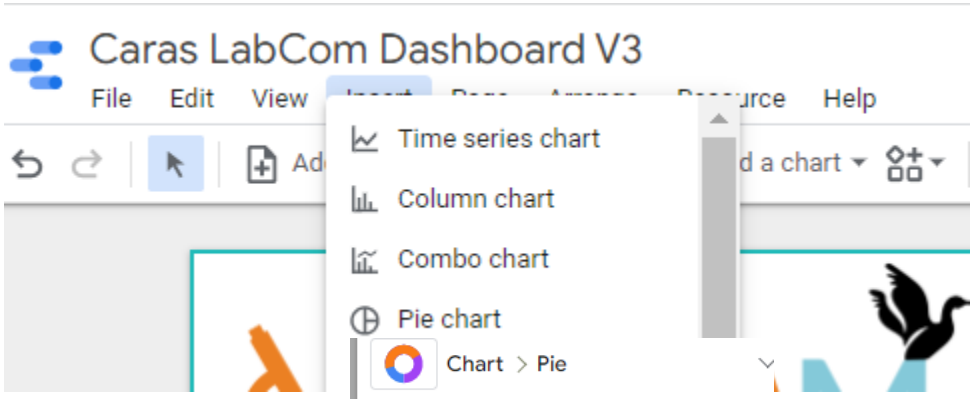


Adding Visualizations

Visualizations are defined as any representation of data that is found in the dashboard. Commonly, these visualizations will be in the form of charts.

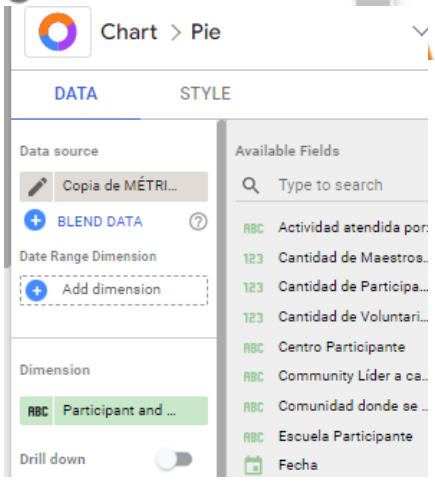
Inserting Charts

Within Data studio there are multiple tabs at the top of the page. Click the insert tab and it will present a dropdown list of a variety of charts. Once you have chosen an option, you can place the chart wherever you want on the page.



Editing Charts

After placing the chart, you can double click on it to open an editor menu on the right of the page. There are two sections, the “Data” and the “Style” sections. The “Data” section allows for you to edit metrics, dimensions, date ranges, and essentially all aspects of the data you are trying to visualize. The “Style” section allows for you to add chart labels, axis labels, edit the style of the visualization, change the colors, etc.



Useful Vocabulary

Dimensions

A dimension is equivalent to a column in Google Sheets with a few minor differences. The difference being that a dimension can be a “column” that only exists within Google Data Studio. This means a dimension can be a calculated field, parameter or a raw data column from the linked spreadsheet.

Metrics

Metrics are similar to a dimension in that they can be used to display data on a chart in Google Data Studio. However, the difference between the two is that metrics can be used to aggregate data in a dimension. For example, a metric could be used to calculate the sum of hours found in a dimension. Furthermore, metrics work by grouping aggregations based on different keys of a dimension.

How Metrics and Dimensions interact

The first table in the example below, there are two dimensions: **Name** and **Hours Donated**.

Name	**Hours Donated**
StudentA	2
StudentB	3
StudentA	5
StudentA	7

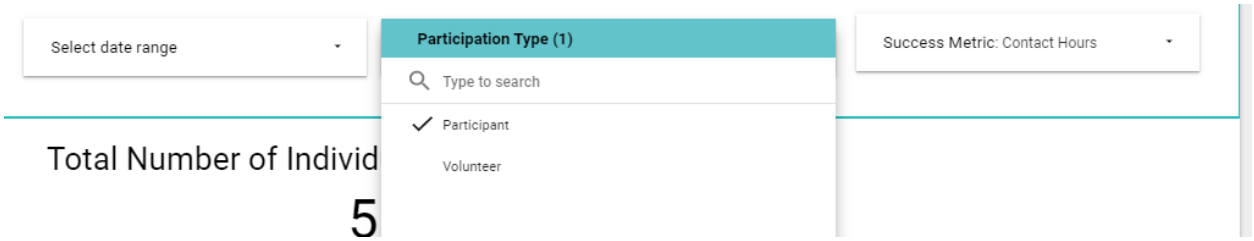
After creating a new metric called **Sum Hours** that would be the sum of **Hours Donated** (In a Google Data Studio formula, this would look like ***Sum(Hours Donated)***), the data would transform by aggregating values in the **Hours Donated** column and grouping by the **Name** column.

Name	**Sum Hours**
StudentA	14
StudentB	3

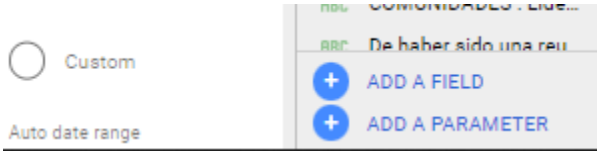
Ultimately, metrics prove to be useful by allowing a user to aggregate data in one dimension and group that aggregation based on another dimension.

Parameters

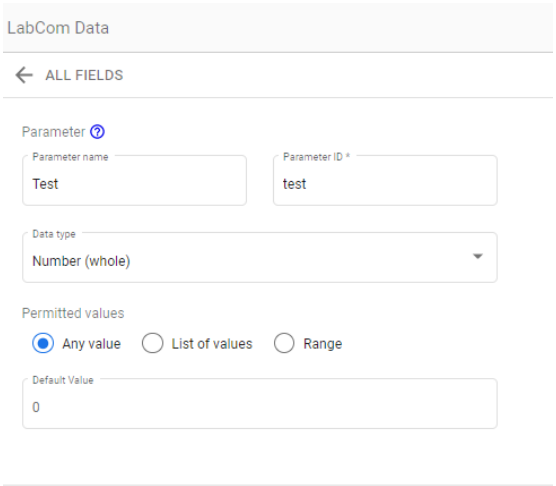
Parameters are key to implementing interactivity within Google Data Studio dashboards. This feature is the first step to allow for users to click drop down boxes that contain multiple options about what specific data should be displayed.



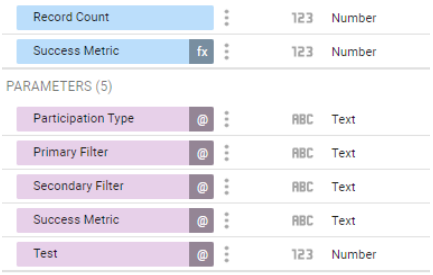
For example, in the image above the “Participation Type” dropdown allows us to choose between “Participant” or “Volunteer” data. Parameters can vary from datatypes, such as success metrics, participation types, or data ranges. Essentially, they let us determine what specific breakdowns we want to display on the dashboards visualizations.



To create a new parameter, navigate to the Google Data Studio dashboard and click the blue edit button in the top right corner of the screen. Next, click the “Add a Parameter” option at the bottom right of the screen. Now, you can type in a “Parameter name” and define what “Data type” it should be.



To view a list of paraments, follow the same process to add a parameter, but hit the cancel button. This will display all dimensions, calculated fields, and at the bottom of the list parameters. Now that you have created a parameter, you will need to understand Calculated Fields to create a drop-down selection to implement interactivity in your dashboard.



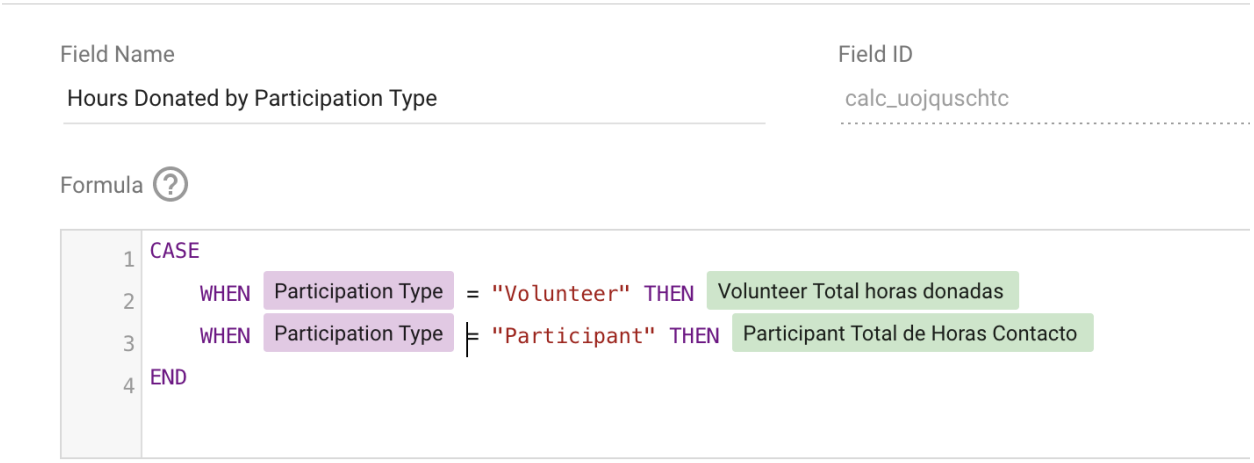
Calculated Fields

Calculated Fields allow a user to manipulate data by creating a new dimension based on other dimensions as well as function operations. Calculated Fields are similar to an Excel Formula in that you can create new data based on your existing columns as well as different functions.

Relation to Parameters

Calculated Fields also become helpful by allowing dimensions to be dynamic based on human input. As explained previously, Parameters provide an interface for human input to Google Data Studio. As a result, Parameter values can be used directly in Calculated Fields.

In our working dashboard prototype, this interaction was used several times. One specific example would be showing hours donated data based on whether the selected **Participation Type** was *Volunteer* or *Participant*. Using the Parameter value that stored **Participation Type**, we were able to set the value of a Calculate Field to be a dimension.



Useful Documentation

Adding Calculated Fields
Tutorial: Creating Calculated Field

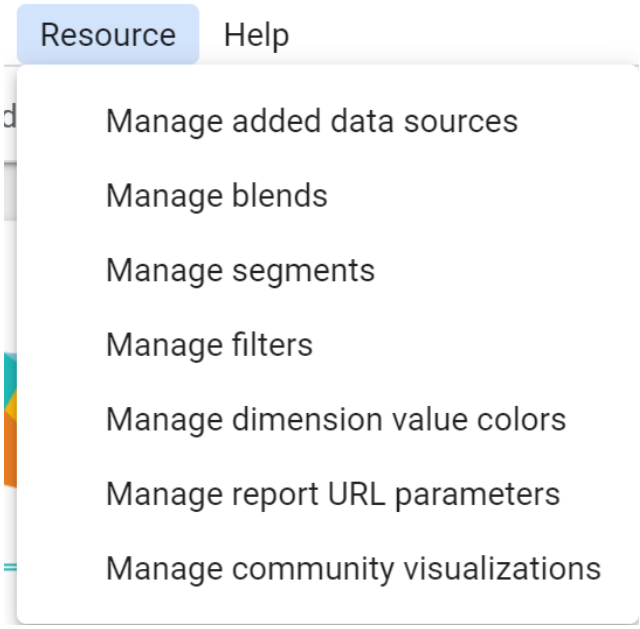
Filters

Filters allow you to filter the information shown in a chart. This has a number of uses, most notably preventing null values from being displayed.

Managing filters (resource-> manage filters)
Adding a filter
Editing a filter
Explain options in filter

Managing Filters

To manage the filters in your Google Data Studio project go to resource -> manage filters

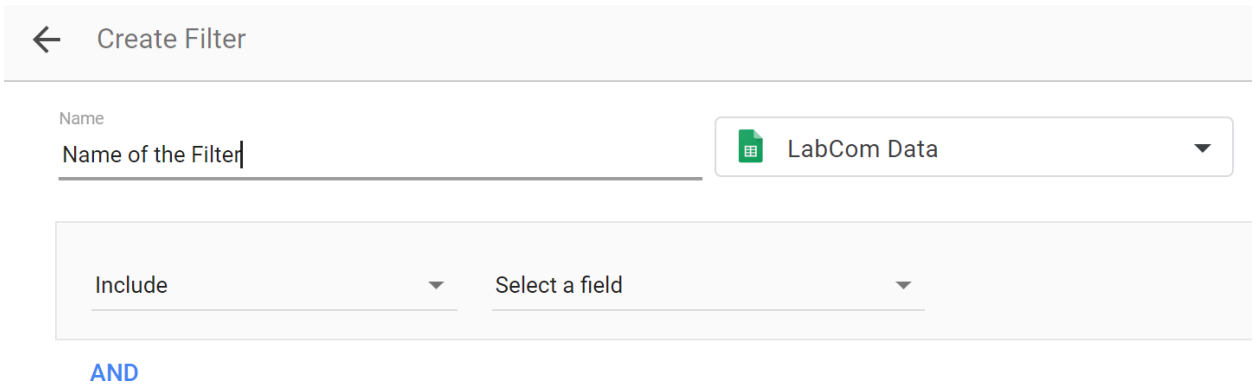


This will allow you to view and edit any of your existing filters, as well as add new filters.

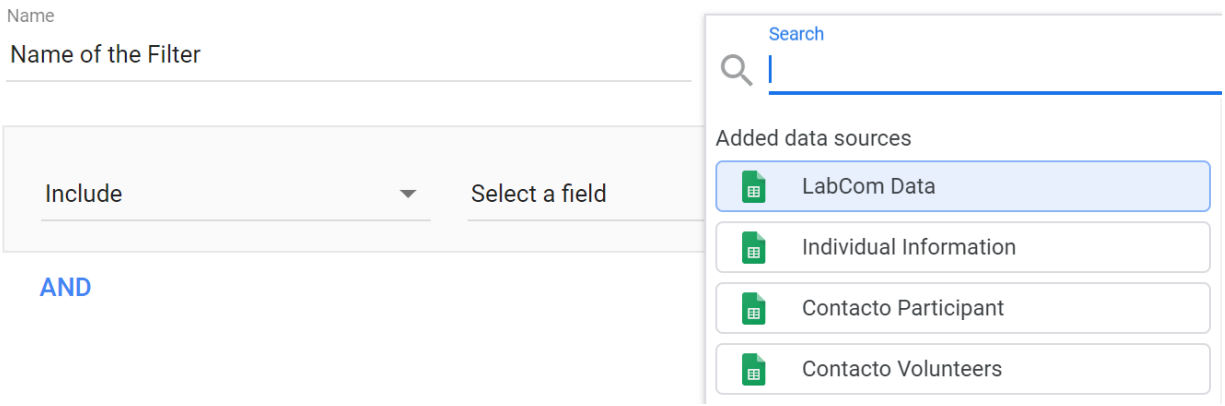
Adding New Filters

To add a new filter go to the “Manage Filters” section and click “ADD A FILTER”

This will open a menu to create a new filter



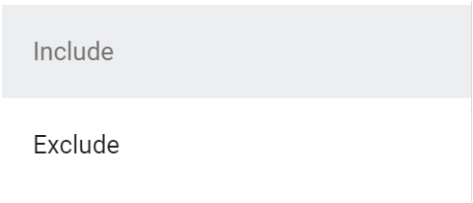
In order to select which data source the filter will apply to, click on the green spreadsheet icon. In the image above, this would be “Labcom Data”



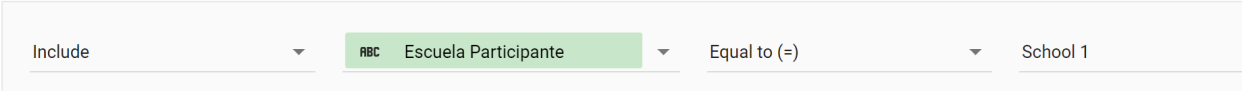
This will bring up a dropdown menu where you can select from your added data sources.

Include vs Exclude

When creating a filter there are two options when you click on the “Include” dropdown. Include and Exclude.



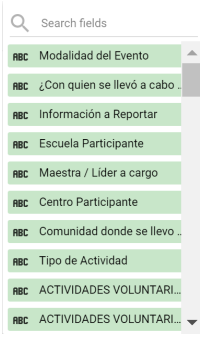
Include means that ONLY the data you select will be shown on graphs with this filter. For example this filter ignores all data points that are not from School 1:



Exclude filters out everything with the specified condition. In the example above changing “Include” to “Exclude” would edit the filter to show data from every school except “School 1”.

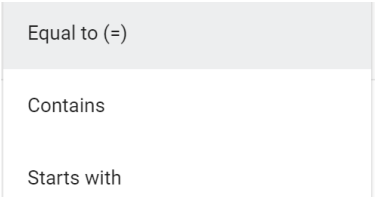
Selecting a Field

To select the field of data that you want a filter to look at, click on the “Select a Field” dropdown menu.



Selecting Conditions

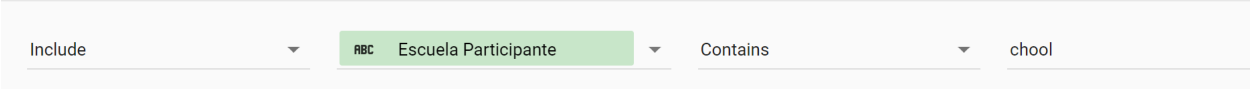
There are multiple conditions that can be applied to a field. Once you have selected a field and whether to include or exclude a new drop down menu will appear asking you to select a condition. Clicking on this menu will provide you with a number of options:



The most useful ones in the context of the dashboard are Equal to, Is Null, Contains, and Starts with.

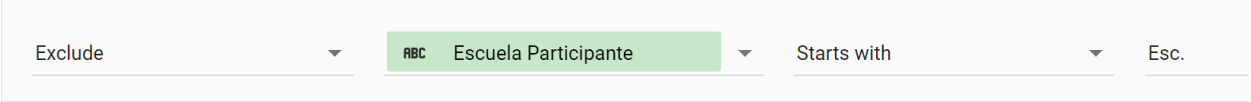
Equal to, checks whether each data point in your chosen field is equal to the value you choose. In the School 1 example above, the filter will only include data points where Escuela Participante equals “School 1”. This is case and whitespace sensitive.

The “Contains” option checks whether each data point in the chosen field contains a certain value.

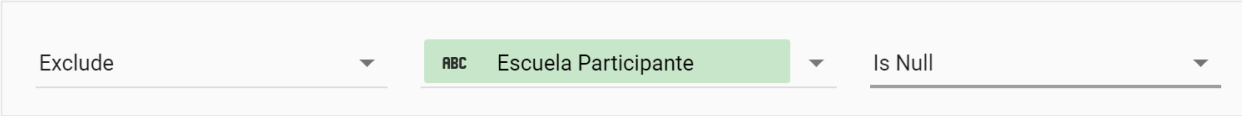


In the above example, this filter will include only schools that have “chool” somewhere in their name.

Starts with is very similar to Contains except it checks whether each data point begins with the chosen value



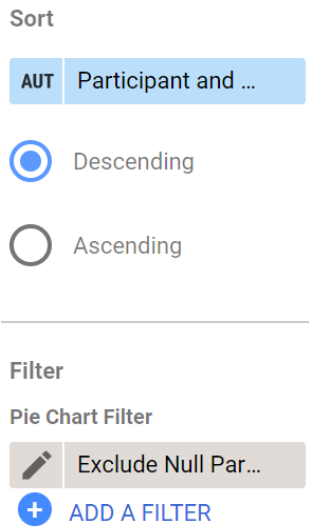
In this example all schools that start with Esc. will be excluded. For example, “Esc. Rosalina” would be excluded but “Rosalina Esc.” or “Esc Rosalina” without the period would not be excluded.
Finally, “Is Null” checks whether the value is null, as a reminder a null value represents an empty entry in the field’s column. This is the most commonly used filter in the dashboard and should be applied any time a chart has “null” as one of the breakdowns.



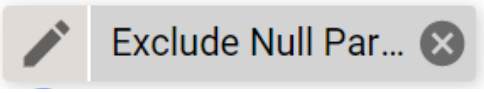
The above example is a filter that will exclude all data points where Escuela Participante is Null. This means that any data points where there is no associated school will be removed from the chart. This is especially important because without the null filter lots of data with invalid entries for Escuela Participante will also be shown on the chart.

Applying Filters to Charts

Once you have created a filter, in order for it to do anything you need to add it to a chart. To do so click on any chart and scroll down on the DATA menu until you see the Filters section

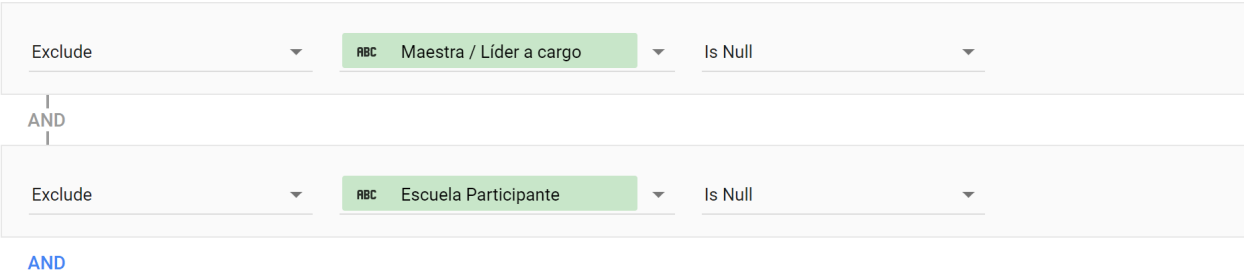


To add a filter to the chart click the add filter button and select the filter you want to apply. You can also directly edit your applied filters by clicking on the pencil icon next to the filters that have been applied. To remove a filter from the graph hover over the filter you want to remove and click the X that appears



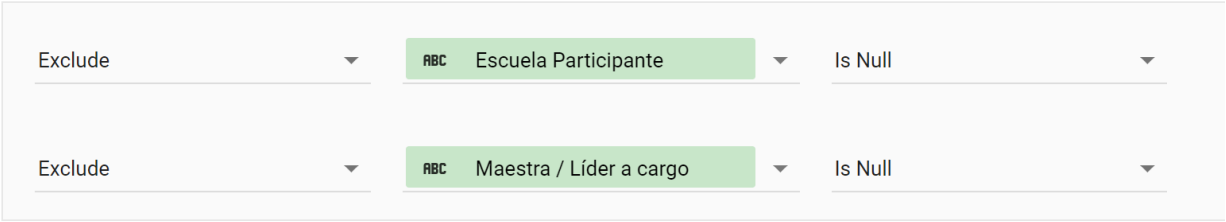
Adding Multiple Conditions to a Filter

If you want to have a filter that checks multiple conditions you need to use the AND functionality in the edit filters menu.



To add a condition, click the blue “AND” button. In the above example the filter will exclude all data points with a null value for Maestra AND exclude all data points with a null value Escuela.

There is also an OR option, which filters out data based on whether both conditions are true. For example the following filter would remove all data that was missing both Escuela Participante and Maestra, but if a data point has at least one of these fields it will still be included.



Escuela and Maestra are Null	Excluded
Escuela Null, Maestra has data	Included
Escuela has data, Maestra Null	Included

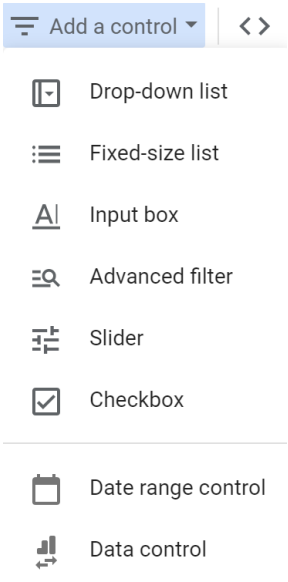
Useful Documentation

About filters - Data Studio Help
Create, edit, and manage filters - Data Studio Help

Controls

Controls are an additional method of controlling what data is displayed on a Google Data Studio Report. They can do a variety of things including filtering data by date and field.

Adding Controls



To add a control to a report click “Add a control” on the top bar of Data Studio.
This will bring up a dropdown menu of the different types of controls. This tutorial will cover date controls and drop down lists.

To add a control to a report click the type of control that you want and then click again where you want to put the control. You can cancel placement of a control by pressing the ESCAPE (esc) button on your keyboard.

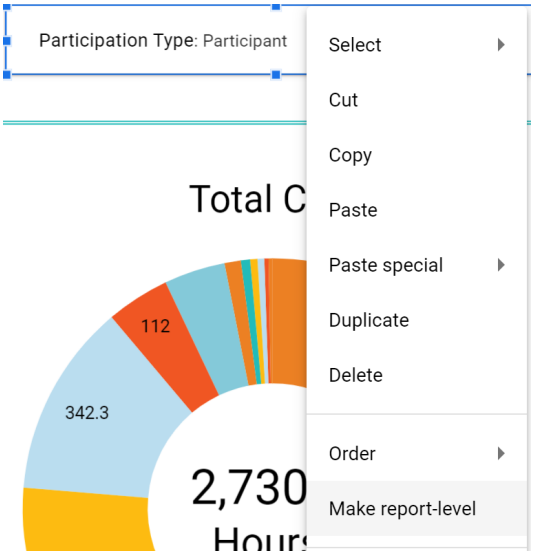
Date range control

Date range controls allow a user to filter the data on a report by the date associated with the data.

When you click on “Select date range”, a calendar will drop down allowing you to choose the start and end dates of the data you wish to display.

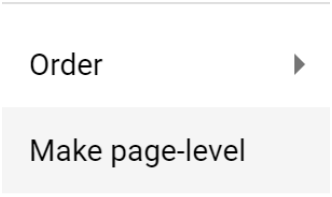
Date range controls work by filtering data based upon the “date range” field of each chart. If a chart is not responding to changes in the date range, double check that it has a valid date range field. To do so click on the chart you want to select and under D

Applying Controls Across Multiple Pages



You may have a Data Studio Report that you want to apply parameters or controls to across multiple pages. To do so, right click on the control/parameter you wish to modify and click “Make report-level”. This will make the control/parameter appear in the same place on every page. Furthermore, selecting a dropdown from the control will cause it to apply to every page on the report.

If you wish to undo this, right click on a report-level control/parameter and select “Make page-level”



Reset Controls and Parameters

If you have modified various controls and parameters and wish to set them to their defaults To reset all parameters and controls on the page to their default values, hit the blue “Reset button at the top of the page”

Useful Documentation

[Google Data Studio: Filters and Controls]
[About controls - Data Studio Help]

Blending

To learn about data blending lets use a relevant example! For this example let's use the two spreadsheets: a contact spreadsheet that contains information about individuals that are involve with Caras con Causa and a spreadsheet that contains individual LabCom project attendance.

	A	B	C	D	E
1	**Name**	**Municipality**	**Gender**		
2	a	Cabo Rojo	M		
3	b	Guayama	F		
4	c	Dorado	F		
5	d	Hatillo	F		
6	e	Hatillo	M		
7	f	Guayama	Prefer not to say		
8	g	San Juan	F		

Table 1: Contact Data

	A	B	C	D
1	**Name**	**Participation Type**	**Project**	**Hours Donated**
2	a	Participant	General LabCom	2
3	b	Participant	General LabCom	1
4	q	Participant	General LabCom	1
5	a	Volunteer	Monitoring Solid Waste	3

Table 2: LabCom Data

What is a blend?

Blends are a feature of Google Data Studio that lets us combine two spreadsheets into a single table. Many of the types of blends work based on a “join condition” that links the two spreadsheets together. This is useful if two spreadsheets provide information **about** the same thing but each provides **different** information. In our example these spreadsheets are perfect to join because they are both providing information for a person, but each provides unique information about the person.

Creating a blend

To create a blend navigate to the *Resource* tab and select manage blends. Then select add a blend. Now select “Join Another Table” to add another spreadsheet. At this point select the two spreadsheets you would like to combine

Now select the oujoin type you want to perform. In this example we selected an inner join. Now drag the data columns from each table that you want to be in the newly created table. In this example we want all of the data from each spreadsheet in our new table.

Inner Join

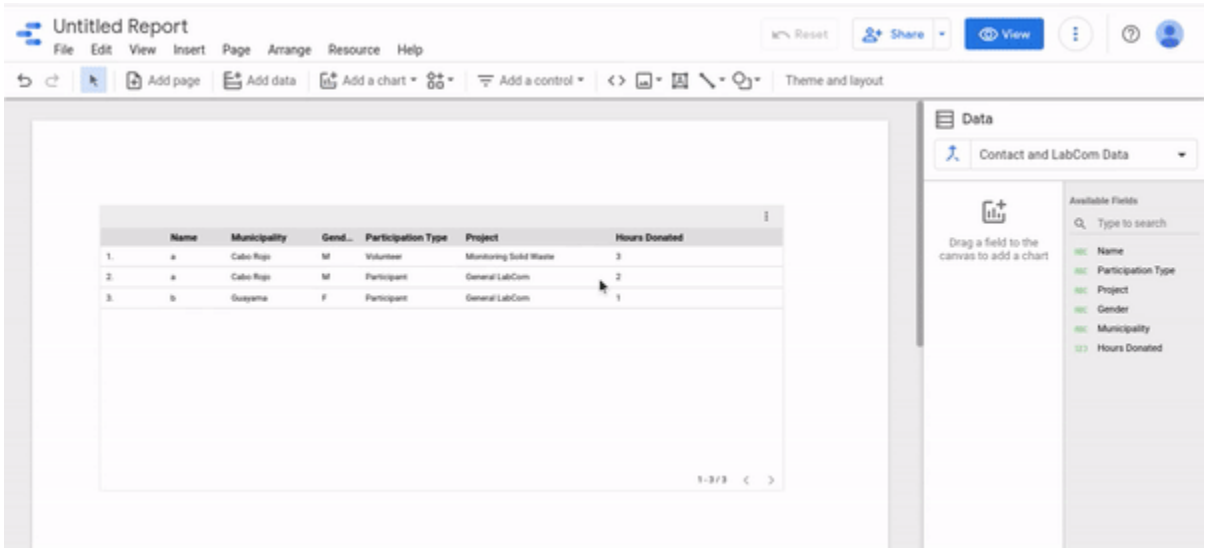
An inner join combines two data sources based on one or more columns where data is shared between the two tables. In our example the column we will join the two sets with is name. The resulting table will only contain names that are shared inside the two tables being blended. The resulting data for the inner join is provided in the following image

	Name	Municipality	Gend...	Participation Type	Project	Hours Donated
1.	a	Cabo Rojo	M	Volunteer	Monitoring Solid Waste	3
2.	a	Cabo Rojo	M	Participant	General LabCom	2
3.	b	Guayama	F	Participant	General LabCom	1

Left Outer Join and Right Outer Join

Like an inner join both right and left outer joins combine two data sources based on one or more columns where data is shared between the two tables. Again to continue our example we will also use the name column. The difference between left/right outer joins and an inner join is that it also takes all members from the left/right table.

Continuing our example, we change the blend we made previously for an inner join to be a left join. The resulting set still contains the individuals with names a and b, however it now also includes q. As you may notice q does not exist in the Contact Data table, it only exists within the LabCom Data table. The individual with the name q is included because a left outer join includes all names from the LabCom table. Since q does not exist in the contact data form there are “missing values” for all the information contained in the Contact Data Table displayed by null.



	Name	Municipality	Gend...	Participation Type	Project	Hours Donated
1.	a	Cabo Rojo	M	Volunteer	Monitoring Solid Waste	3
2.	a	Cabo Rojo	M	Participant	General LabCom	2
3.	b	Guayama	F	Participant	General LabCom	1
4.	q	null	null	Participant	General LabCom	1

Figure : Result from Left Outer Join

Now to perform a right outer join we will do the same steps as we did above to change the join condition to be a right outer join. The result of the right outer join will contain all names that are in both spreadsheets and all names in the right table (Contact Data).

	Name	Municipality	Gend...	Participation Type	Project	Hours Donated
1.	a	Cabo Rojo	M	Volunteer	Monitoring Solid Waste	3
2.	a	Cabo Rojo	M	Participant	General LabCom	2
3.	b	Guayama	F	Participant	General LabCom	1
4.	c	Dorado	F	null	null	null
5.	d	Hatillo	F	null	null	null
6.	e	Hatillo	M	null	null	null
7.	f	Guayama	Prefer ...	null	null	null
8.	g	San Juan	F	null	null	null

Figure: Result from Right Outer Join

Fun Fact: Performing a Left Outer Join on table 1 and table 2 where table 1 is the left table is the same as performing a Right Outer Join on table 2 and 1 where table 1 is the right table!

Cross Join

A cross join is similar to doing a left and a right outer join at the same time. However the column that the join was performed on (in our example this was name) does not appear as a single column like we saw in all previous joins. Instead the left outer join is computed and placed into its own name field, similarly the right outer join is computed and placed into its own name field.

	Name (Table 1)	Name (Table 2)	Municipality	Gender	Participation Type	Project	Hours Donated
1.	a	a	Cabo Rojo	M	Volunteer	Monitoring Solid Waste	3
2.	a	a	Cabo Rojo	M	Participant	General LabCom	2
3.	b	b	Guayama	F	Participant	General LabCom	1
4.	q	null	null	null	Participant	General LabCom	1
5.	null	c	Dorado	F	null	null	null
6.	null	d	Hatillo	F	null	null	null
7.	null	e	Hatillo	M	null	null	null
8.	null	f	Guayama	Prefer not to say	null	null	null
9.	null	g	San Juan	F	null	null	null

Figure : Result from Cross Join

Full Outer Join

Performing a cross join will take each row of the left table and combine its data with all the rows in the right table. This means that all possible combinations of individuals are shown as a result of a full outer join. Please note this is the most expensive of the joins.

	Name (Table 1)	Name (Table 2)	Municipality	Gender	Participation Type	Project	Hours Donated
1.	a	a	Cabo Rojo	M	Volunteer	Monitoring Solid Waste	3
2.	a	b	Guayama	F	Volunteer	Monitoring Solid Waste	3
3.	a	c	Dorado	F	Volunteer	Monitoring Solid Waste	3
4.	a	d	Hatillo	F	Volunteer	Monitoring Solid Waste	3
5.	a	e	Hatillo	M	Volunteer	Monitoring Solid Waste	3
6.	a	f	Guayama	Prefer not to say	Volunteer	Monitoring Solid Waste	3
7.	a	g	San Juan	F	Volunteer	Monitoring Solid Waste	3
8.	a	a	Cabo Rojo	M	Participant	General LabCom	2
9.	a	b	Guayama	F	Participant	General LabCom	2
1...	a	c	Dorado	F	Participant	General LabCom	2
1...	a	d	Hatillo	F	Participant	General LabCom	2
1...	a	e	Hatillo	M	Participant	General LabCom	2
1...	a	f	Guayama	Prefer not to say	Participant	General LabCom	2
1...	a	g	San Juan	F	Participant	General LabCom	2
1...	b	a	Cabo Rojo	M	Participant	General LabCom	1
1...	b	b	Guayama	F	Participant	General LabCom	1
1...	b	c	Dorado	F	Participant	General LabCom	1

1 - 28 / 28

Figure : Result from Full Outer Join

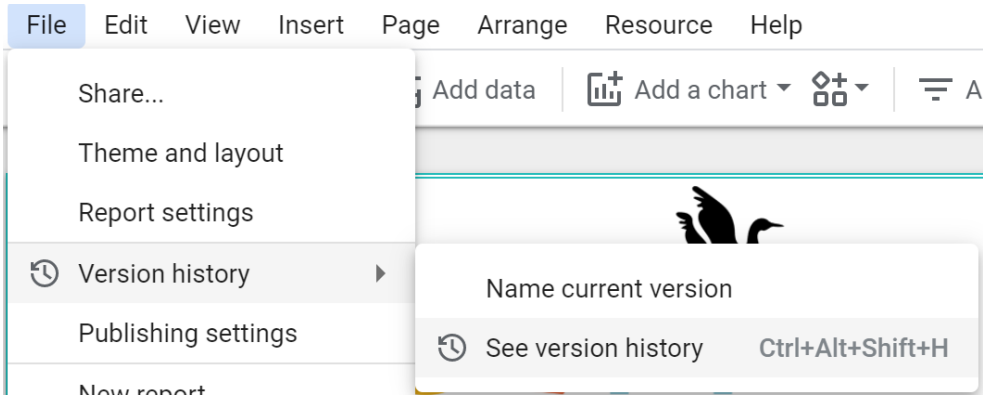
Useful Links

https://support.google.com/datastudio/answer/9061420?hl=en

Restoring from a Backup

Just in case you change something on the dashboard that you can't remember how to change back, there are options available to undo what you have done.
These buttons up top, along with Control/Command Z can undo the last thing that was done

In some cases though, this may not be enough. When you have made a large mistake that you cannot undo through control z, Google Data Studio keeps frequent backups of the dashboard through version history.



This will allow you to click on all of the backups Data Studio has on the right side menu. If you wish to restore a particular backup, click

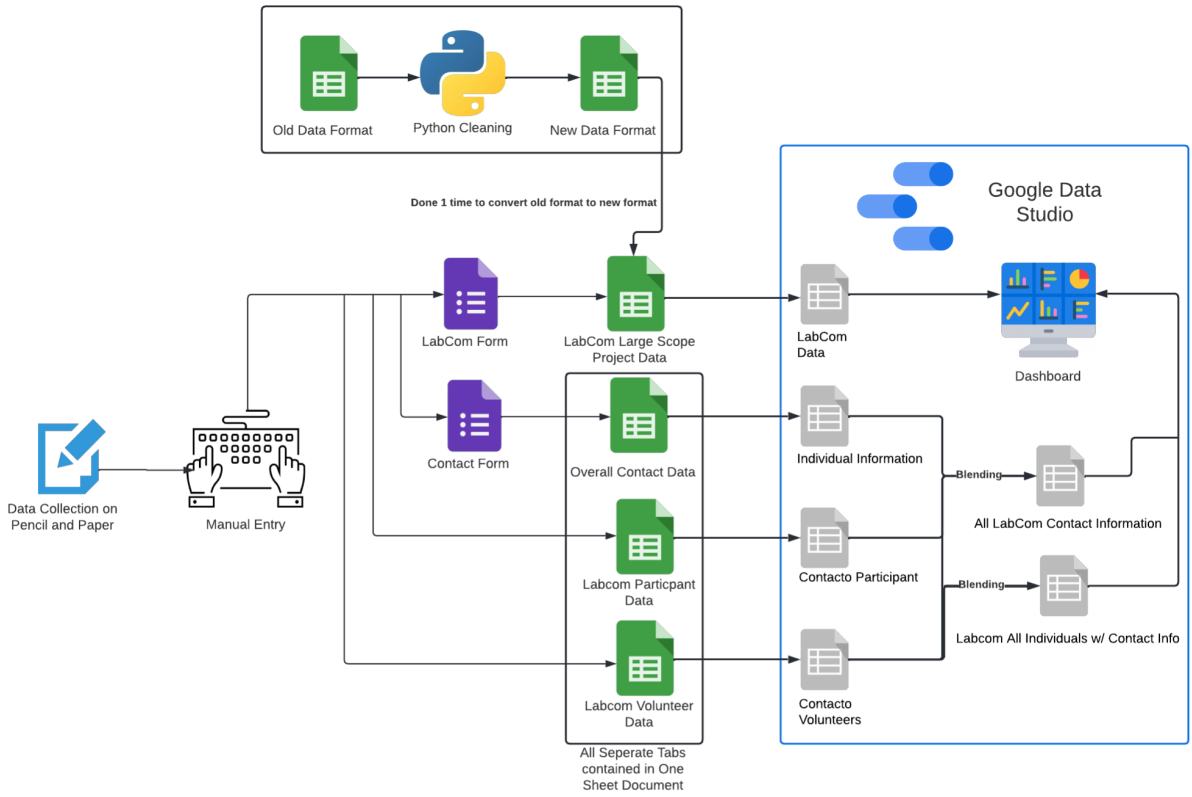
Restore this version

Fundamental Knowledge to Continue Dashboard Development

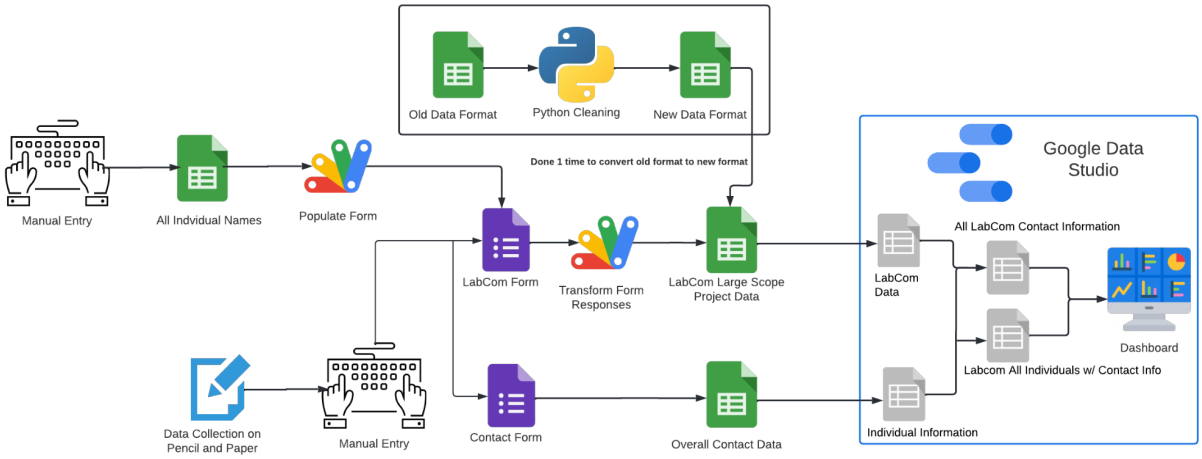
Dashboard

Skills Needed

Overview of Dashboard Architecture



LabCom Dashboard Architecture



LabCom Dashboard Architecture w/ Dynamic Population

Data Cleaning

Caras con Causa's data was not tailored to be visualized in a platform like Google Data Studio initially. Due to this, problems arose that required multiple meetings to sort out. This section will define the iterative process that occurred to resolve these problems that you should be aware of to not repeat.

Process

As we developed each version of our dashboard, we ran into problems with the data that we were using. There were three main problems: duplicated header names, comma separated data, and unstandardized text entries.

Duplicated header names refer to the column titles in a Google Sheets document. If two column names are the same, then Data Studio cannot distinguish between the two columns of data. This is caused by the linked Google Form using a repeated question. To solve this, we changed the question names to be more specific on the Google Forms.

Comma separated data refers to data that is grouped together and separated by a comma. For example, an entry says that 50 people in grades 6,7, and 8 served 50 hours. There is no way to distinguish what contribution grade 6 did specifically. Overall, this undermines the data and does not show individual contributions. We developed python code to separate these values out, and average them across the board. We arrived at this conclusion after numerous talks with Caras con Causa. Additionally, we removed the option in Google Forms to select multiple checkboxes of data so that this problem cannot keep happening.

Finally, the Google Forms have unstandardized text entry options. This would make data appear in Google Sheets as "John", "JOHN", or "john". While we can easily distinguish that these all

mean the same person, Google Data studio cannot. To fix this, we standardized their Google Form answers and removed all text entries for the data that we care about. Additionally, we went through their Sheets to standardize all entries that were supposed to be the same.

Skills Needed

Understanding of data flow between Google Forms, Sheets, and Data Studio

Python Script

During one of our meetings with Caras con Causa, we discussed the importance of differentiating Event Statistics versus Project Statistics. As a result several changes to the data format of the main LabCom spreadsheet changed:

1. Columns were added to track total number of participants, number of volunteers, and number of volunteer leaders on an event level
2. Column added to denote whether a row should be considered as a new row
3. Project names can no longer be comma separated (Project1, Project2, Project3)
    a. Once project names are separated, certain columns such as hours donated should be divided evenly between all projects

As a result of these changes, we needed to transform the old data format into the new data format. In order to do so as efficiently as possible, our group developed a Python script to programmatically transform the old data set into the new data set format. The Python script works with a local XLSX file. Therefore, the first step in order to use the Python script would be to download the Google Sheet as an Excel file. After that, place the Python script and the XLSX file in the same directory. Finally, in the PY file containing the script, there are numerous configuration options at the top:

FILENAME: String value of the local XLSX file
SHEET_NAME: String value of the specified sheet to work with within the spreadsheet
SPLIT_DELIMETER: String value of the delimiter that is used to split project values (Should be ‘, ‘)
COMMA_SEPERATE_COLUMNS: A dictionary where keys indicate an alphanumeric column header and values are an array of alphanumeric columns where values should be split evenly between the elements of the key
NUM_VOLUNTEERS_COLUMN: Alphanumeric column header that contains the number of volunteers
NUM_PARTICIPANTS_COLUMN: Alphanumeric column header that contains the number of participants
NUM_VOLUNTEER_LEADERS_COLUMN: Alphanumeric column header that contains the number of volunteer leaders
IS_EVENT_COLUMN: Alphanumeric column header that contains whether or not a specific row should denote a new event

IS_EVENT_STR_CONSTANTS: Should be a tuple containing two values where the first element is the equivalent of entering “Yes” into the “Is this a new event question”
TOTALS_NUM_VOLUNTEERS_COLUMN: Alphanumeric column header that contains the event total number of volunteers
TOTALS_NUM_PARTICIPANTS_COLUMN: Alphanumeric column header that contains the event total number of participants
TOTALS_NUM_VOLUNTEER_LEADERS_COLUMN: Alphanumeric column header that contains the event total number of volunteer leaders

Once all of these configuration options have been set properly, the Python script can be run and will output a new excel sheet called “new_data.xlsx” containing the transformed data.

Skills Needed

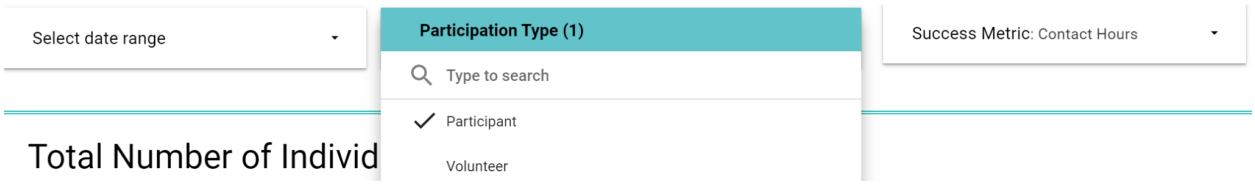
Scripting Programming Language, Understanding of how Excel Files are structured

Final Product Usage Guide

The final product of this project encompasses two dashboards, one for Caras con Causa’s LabCom program and one to track individuals across the whole organization. These dashboards can be accessed on Google Data Studio by using the Google accounts provided by Caras con Causa.

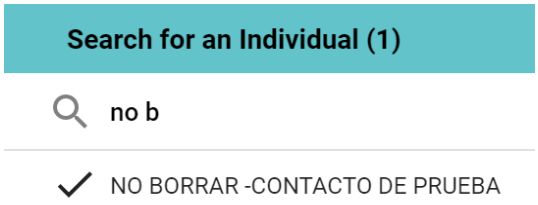
Interactable Components

Each dashboard has a number of components that can be interacted with to change how the data is displayed. These appear in the form of dropdown menus.

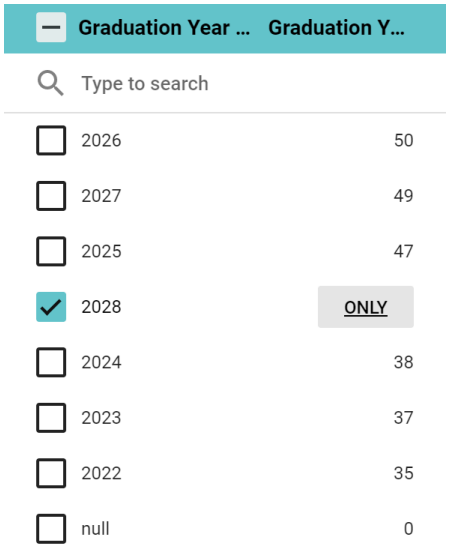


Selecting a component in the dropdown menu will change the corresponding charts. Selections can be reset to their defaults with the reset button on the top menu.

Some dropdown menus have search bars. To use them, click into the dropdown and click Type to search, and then start typing in what you want to search for. This will narrow down the list of options that you can select from by the search terms.

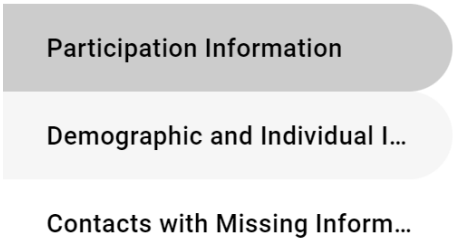


Some dropdowns allow you to select multiple different options. All options that have been checked will appear in the drop down's associated charts and those that are not checked will be excluded. If you wish to only see one of the categories, hover over the selection and click the "ONLY" button that appears.
To check all of the fields at once, click the "-" button at the top of the dropdown menu. This will reset the selection to its default value, where every option is checked.



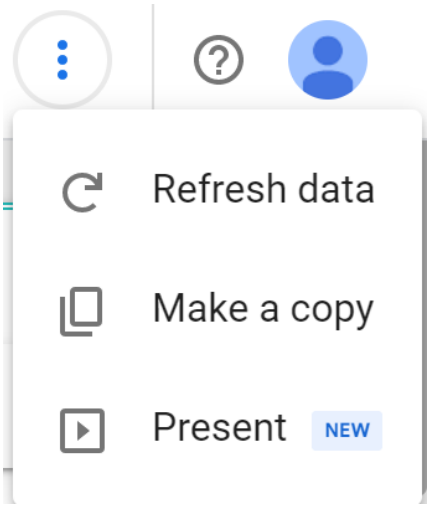
Page Navigation

Each dashboard has multiple pages which contain distinct types of information. To navigate to each of these pages, click on the page you want to look at on the left sidebar.



Refreshing Data

Google Data Studio automatically ensures that the data on the report is up to date every 15 minutes, but you can also do this manually at any point in time. To pull the most recent data from your data source, click the triple dots at the top right of the page and select “Refresh data”



Dynamic Form Overview

Skills Needed

Understanding of JavaScript, time to read through Google API Documentation

Relevant Links

Google Form API: https://developers.google.com/apps-script/reference/forms
Google Spreadsheet API : https://developers.google.com/apps-script/reference/spreadsheet

Why create a dynamic form?

In meetings with Caras con Causa and working with their data we realized we could not extract important information such as the projects an individual has attended, or the number of hours a person has donated to a specific project. This information is important because the organization would like to display the success of their programs by showing specific/all outcomes for individuals that attend their program.

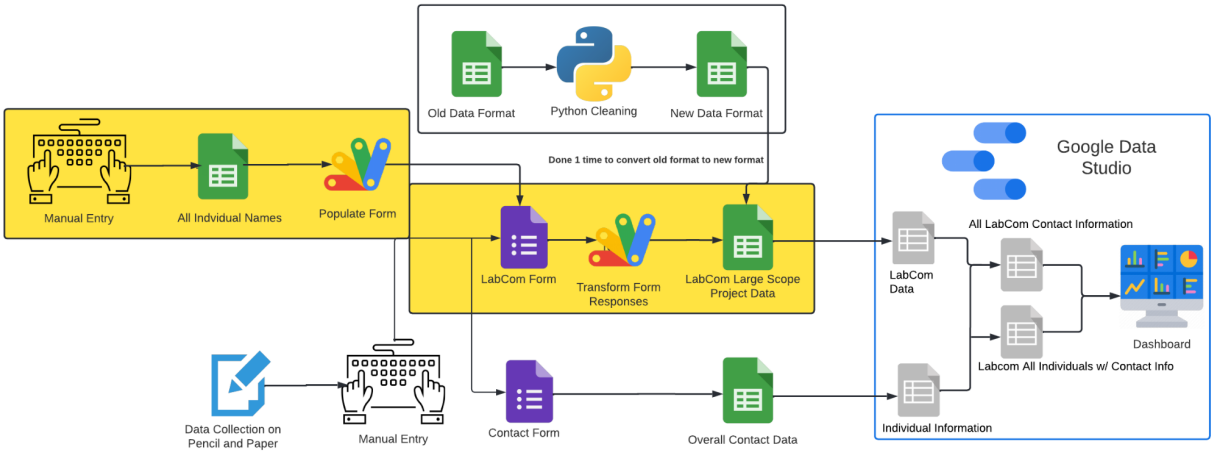
Originally Caras con Causa would manually collect hours for each individual but the format this data was collected in includes the individuals name, their hours, and 365 columns where each column is a day of the year and data in these columns is the project they attended. This format was extremely hard to use within Google Data Studio and using it did not provide all possible information.

We created the dynamic form to solve the issues discussed above. The form needs to be dynamic because the people Caras con Causa serves often changes. We design for this factor by centralizing individuals' names by groupings (such as School and Grade) and allowing the team to add to this information by adding new data and at the click of a button update the form questions.

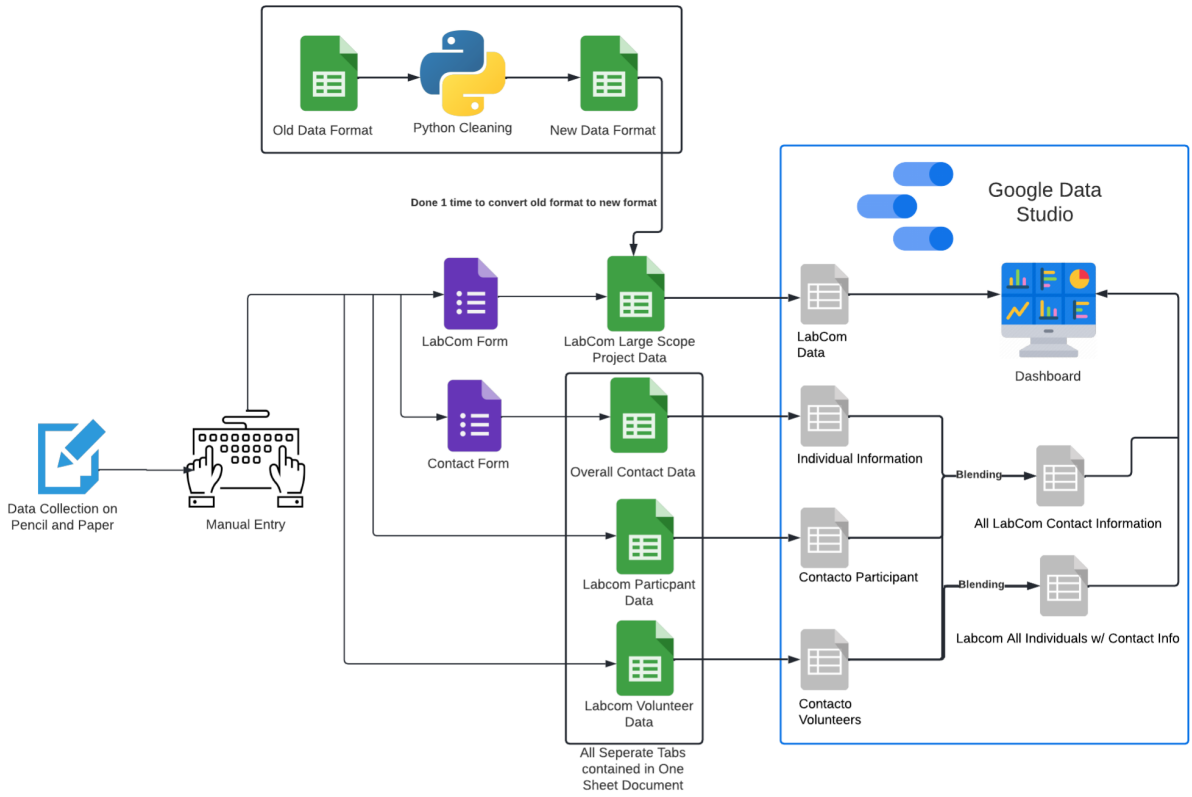
What is AppScript?

Google products can have JavaScript code inserted into them via an extension called AppScript. By using this extension we are able to add functionality that we need.

Proposed Architecture Overview



Including the Dynamic Form into Caras con Causas technology stack would result in the highlighted sections. Previously the structure looked like this

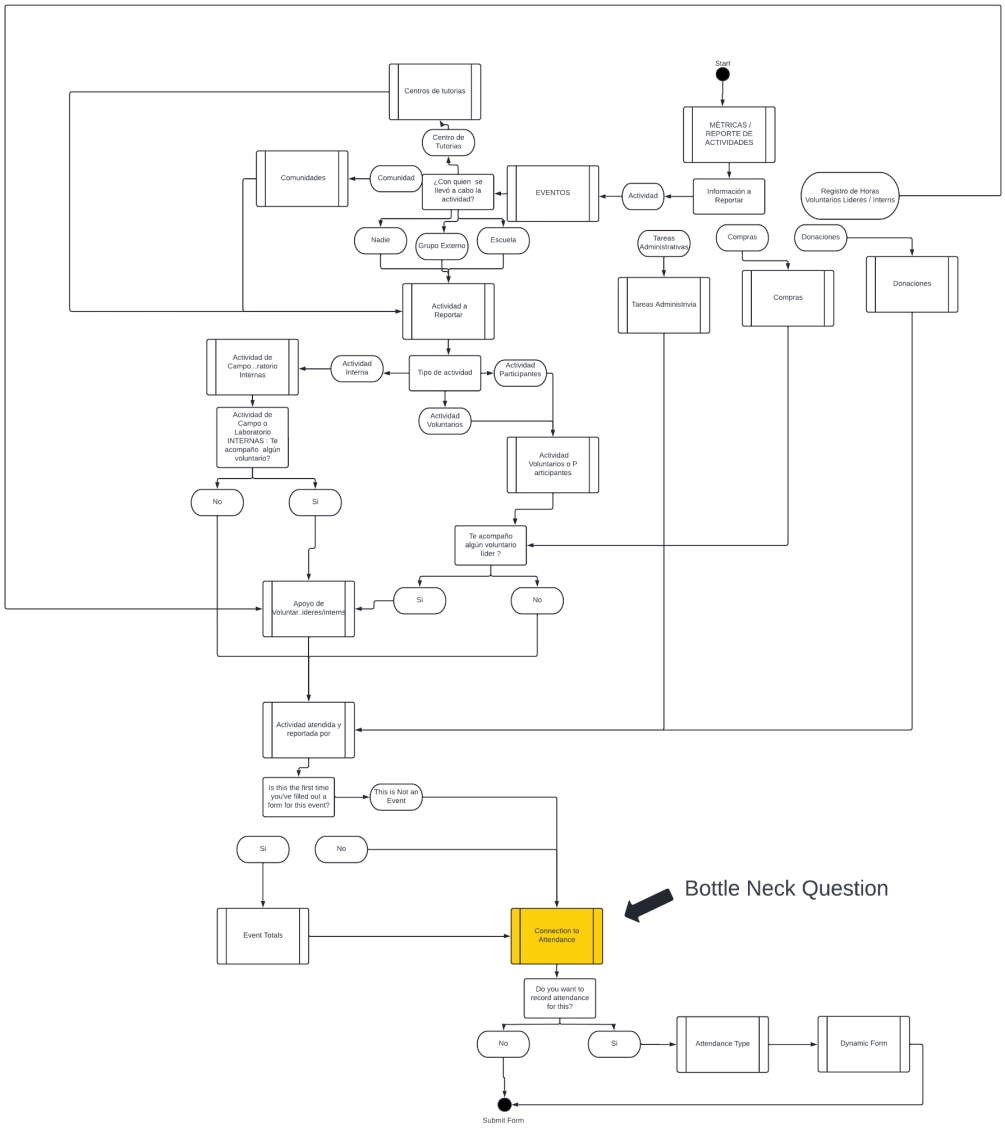


LabCom Form

The new LabCom Form is an adaption of the original. The main difference is the addition of a “bottle neck” question. The term bottleneck is used to indicate that all questions lead into this question, forcing the user to decide if attendance needs to be taken. This simplifies the number of adaptions needed to make the form adaptable, as only one question needs to link to the first part of the form. The alternative would be to link every question where attendance to be taken to the first part of the dynamic form. However this would prevent the user from deleting the dynamic questions easily.

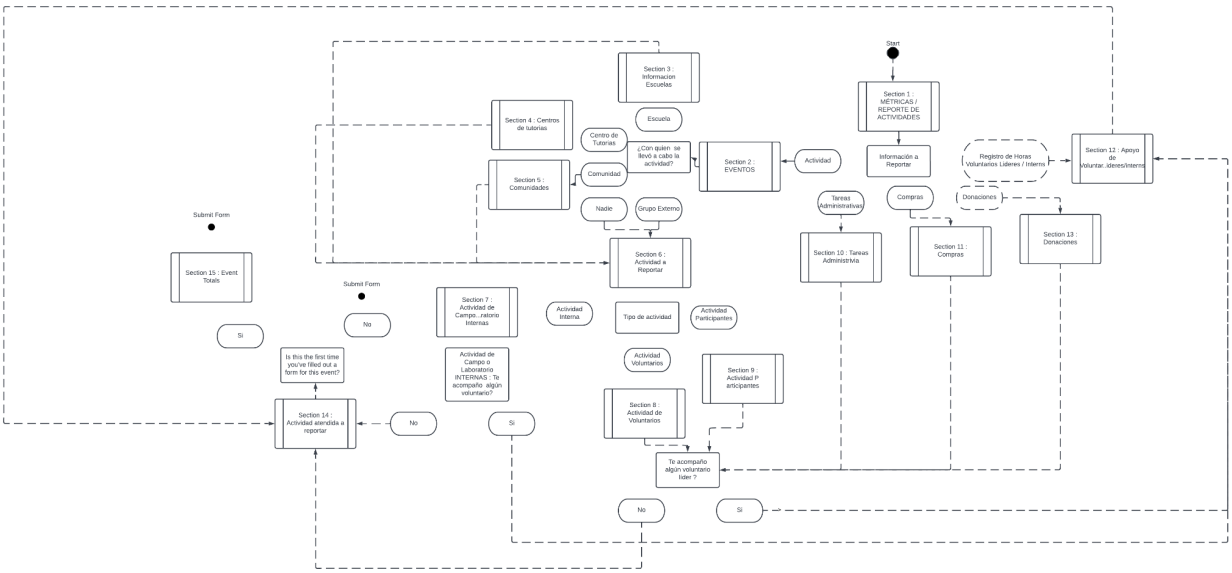
Other differences between the new and original LabCom form include removing unneeded questions including those that reference social metrics recorded by the dynamic questions.

New Format



Note this is clickable!

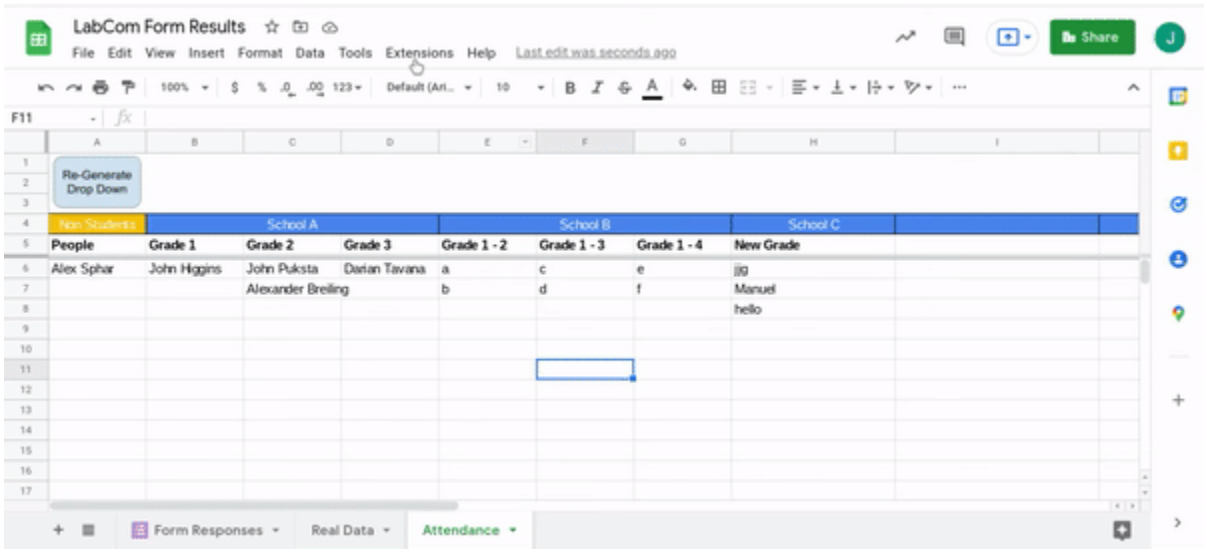
Old Format



Dynamic Form Usage Guide

Accessing Code

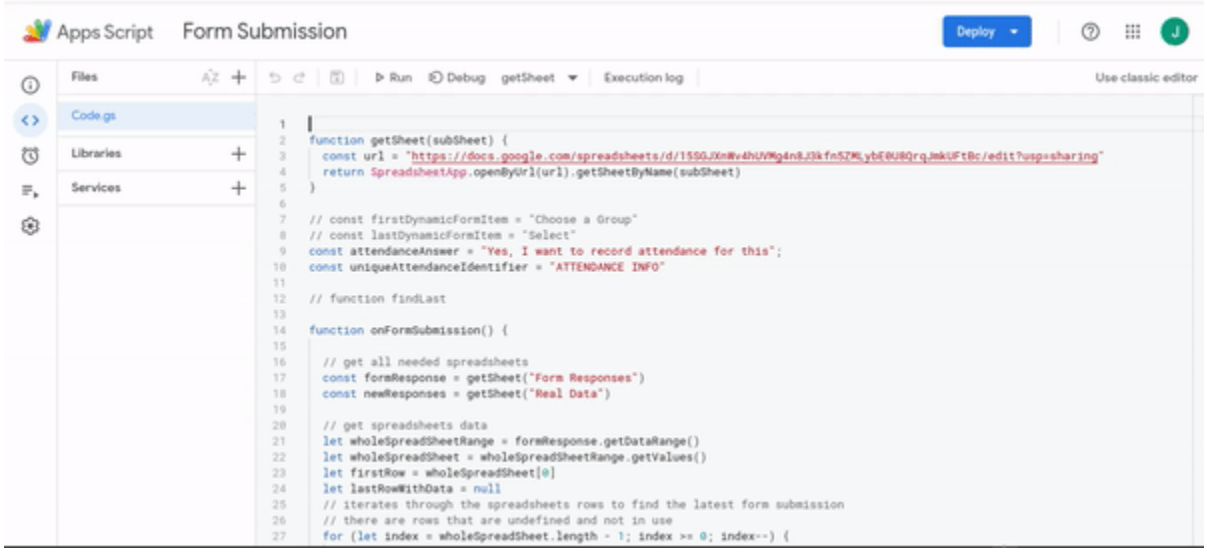
Google Sheets



Google Forms

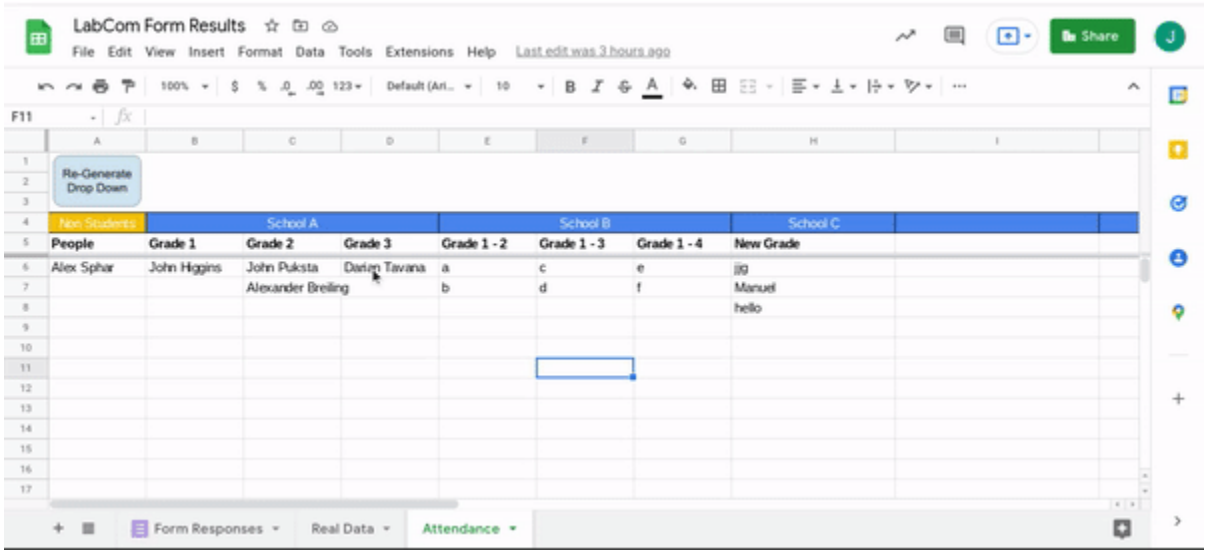
Triggers

Every time an action is done, such as submitting a form, code can be executed. These are called triggers and were used to restructure the data to remove redundant questions.



Features

Accessing Features



Each time the spreadsheet is opened in order to use the features, you must click the regenerate drop down button. This creates a drop down menu that when an item is selected executes the function

Create/update Questions

This feature checks if attendance questions have been created already, if they have it updates the individuals in each subgroup. If they have not been created yet, all the attendance questions based off of the attendance spreadsheet are created on the form passed in. New questions are added to the end of the form
Please note this may take a few minutes to create the question. Sit back, relax and enjoy the automation! :)

Delete Generated Questions

This feature deletes all the attendance questions based off of the attendance spreadsheet on the spreadsheet passed in. A limitation of this function is that it relies on the contents of the attendance spreadsheet, so if a grouping and its subgroupings is deleted, the corresponding questions will not be deleted by this function.
To improve this function it would be good to have a way to easily filter all items that are dynamic form questions. This could be done by including ATTENDANCE FORM within each item title then deleting all the form items that contain that string.

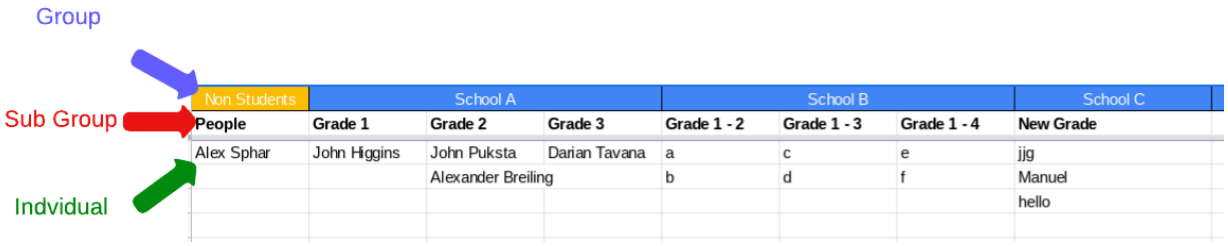
Re Structuring Data

Every time the form is submitted the onSubmission trigger activates and restructures the data. This function reduces the number of columns within the data spreadsheet and separates all

selected individuals to different rows. This makes analysis for each individual possible, while making it analyzable in Google Data Studio.

Attendance Spreadsheet

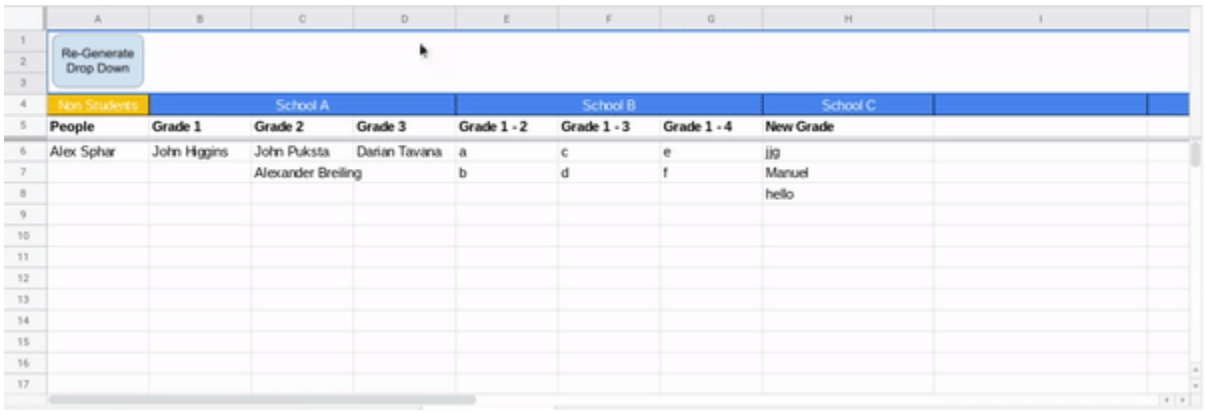
Structure



Updating Spreadsheet

Groups

Add



1. Insert a new column
    a. Warning do not insert a column to the left of the generate questions button
2. Run Create Questions on the form

Update

1. Edit the cell
2. Run Delete Questions on the form
3. Run Create Questions on the form
4. Relink the bottleneck question to the start of the generated questions

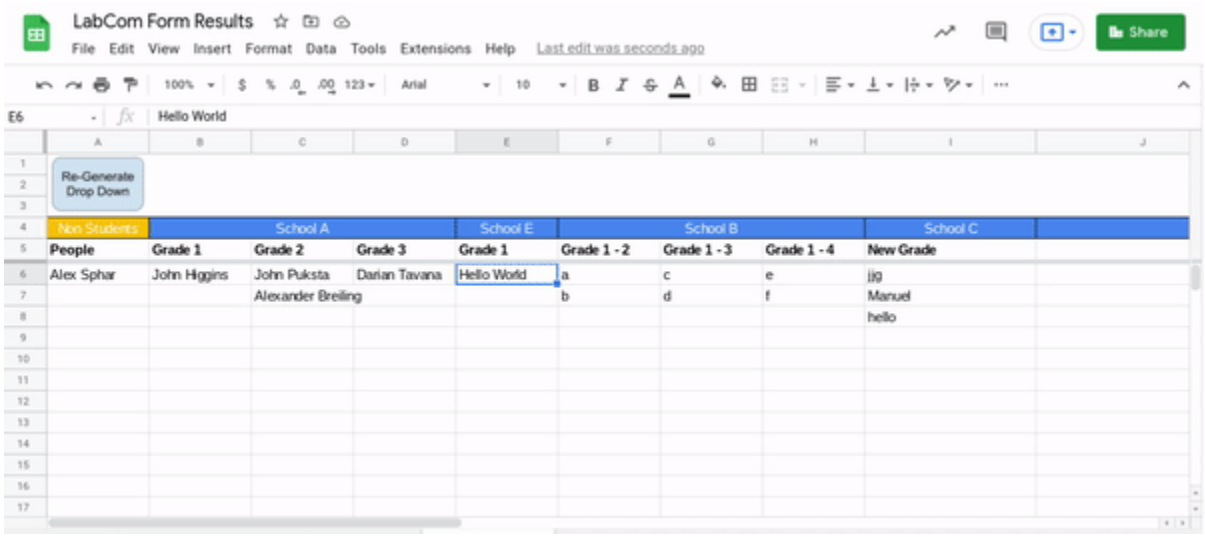
Delete

1. Delete columns from spreadsheet, shifting data over

    a. Warning do not delete the column with the generate questions button
2. Run Delete Questions on the form
3. Run Create Questions on the form
4. Relink the bottleneck question to the start of the generated questions

Sub Groups

Add



1. Insert a new column to the right of the group you wish to add to
2. Merge the created column with the other groups column
3. Enter in the sub group name to the created column
4. Enter in the indviduals names
5. Run Create Questions on the form

Update

1. Edit the cell
2. Run Delete Questions on the form
3. Run Create Questions on the form
4. Relink the bottleneck question to the start of the generated questions

Delete

1. Delete columns from spreadsheet, shifting data over
    a. Warning do not delete the column with the generate questions button
2. Run Delete Questions on the form
3. Run Create Questions on the form
4. Relink the bottleneck question to the start of the generated questions

Individuals

Add

1. Simply add the name to a new row to the desired group
2. Run Create Questions on the form

Update

1. Change the name
2. Run Create Questions on the form

Delete

1. Delete the name
2. Run Create Questions on the form

Adapting Other Forms

Make a Copy of the Existing From

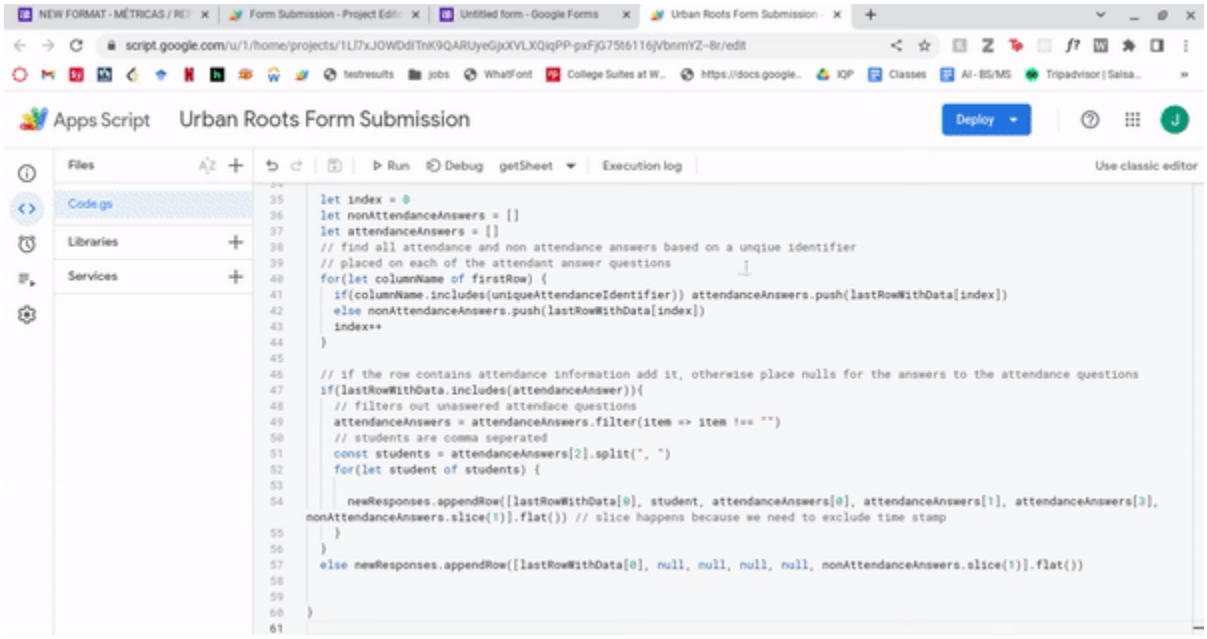
Make a copy of the form you want to add the attendance questions to. This creates an easy way to undo any changes you make.

Copy On Form Submission Code to new Spreadsheet

Copy the code from this form to the new form

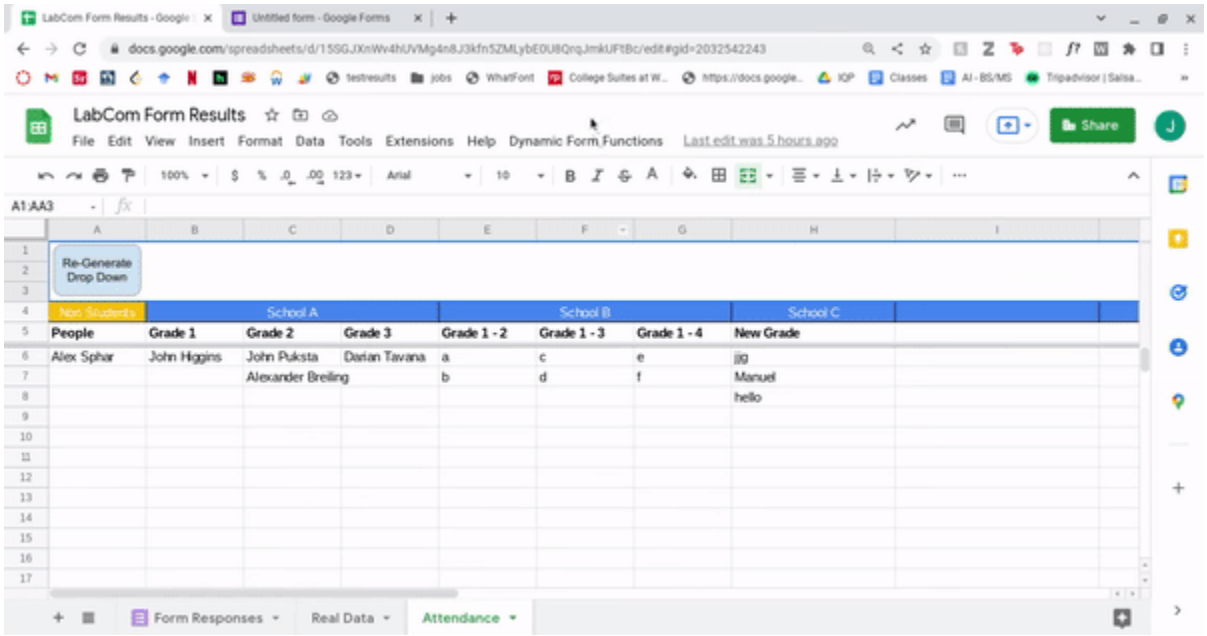
Add OnSubmission function as a trigger

Within the code editor window click triggers then add a trigger and select the OnSubmission function and on form submission. This will require the user to authorize that code they are running is safe for themselves (which it is!).

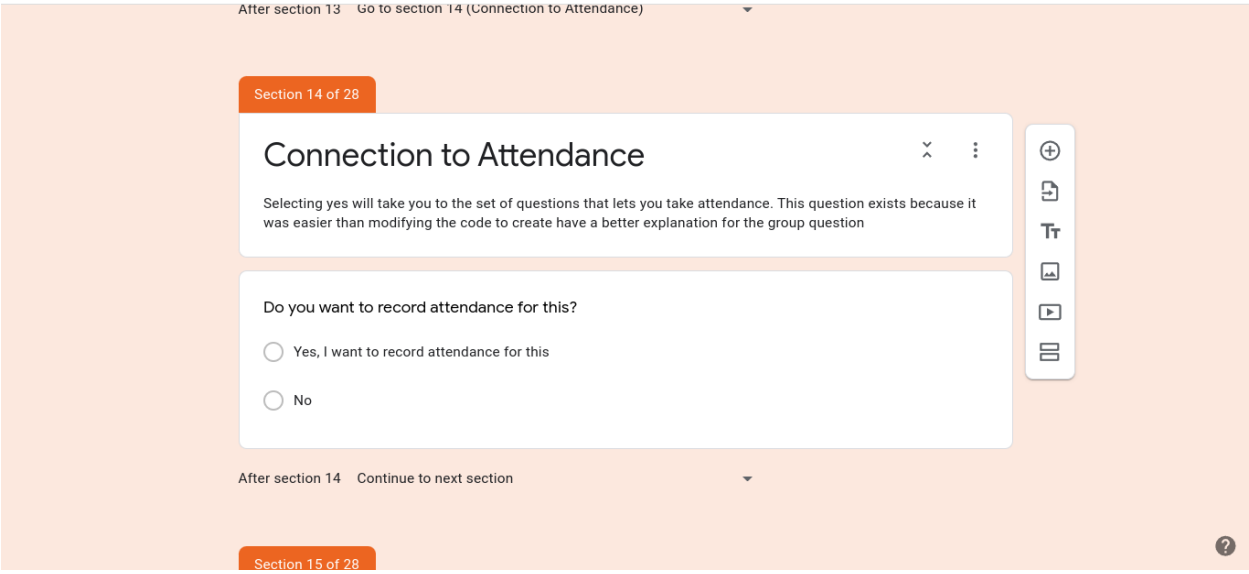


Adding Dynamic Questions to the Form

Adding the attendance questions to a different form was designed to be easy. To do so the form needs to have a shareable link that is editable so the script can make changes. Copy this link and paste it into the dialog box and the questions will generate.



To link these questions to the rest of the form it is recommended you create a “bottleneck” question where all form paths lead into a single question that asks the user filling out the form if they want to fill out attendance.



This is the “bottleneck” question in the LabCom form.