

Google Summer of Code 2023 Proposal - LibreOffice

Histogram (and other missing) Chart Type

Personal Details

Name : Devansh Varshney
E-mail : varshney.devansh614@gmail.com
Country : India
IRC name : devansh
Phone number : +91-790-697-3823
Time Zone : UTC +5:30

Details of the task

Description - Lack of support of Histogram and other new added chart types from MSO16 in LO.

Goal - Adding native support for histogram charts in LO to eliminate the need for workarounds and improve the user experience.

Skills - C++ and Reading other's code

Size - ~350 Hrs

Difficulty - Hard

Mentor - Tomaž Vajngerl (IRC: quikee | mail: quikee@gmail.com)

Proposal

A) What are Charts and why they are used?

Charts in LibreOffice, Microsoft Office, or in general are visual representations of data that help users analyze and understand information more effectively. In LibreOffice the 'Insert Chart' button is typically located in the toolbar or menu bar (Fig1), allowing users to easily create visual information charts from their data. With MS Office 2016 Microsoft introduced new new types of charts to align with the current needs for example data analysis, scientific researchers, or ML teams by providing advanced visualization charts that cater to the complexities of modern data analysis.

Since, MS Office Excel is being regarded as the standard and providing these new chart types, such as the *Histogram Chart*, *Sunburst Chart*, *Treemap Chart* etc which are need of the hour.

We could effectively enhance the user experience and compatibility with the current market standards.

In easier term chart is basically a mapping of bunch of values/numbers with other way of representing/understanding of those values. Which is equivalent to mathematical functions where we map some integer values(input) with some other way of understanding of those values which is in our case visualizations(output) .

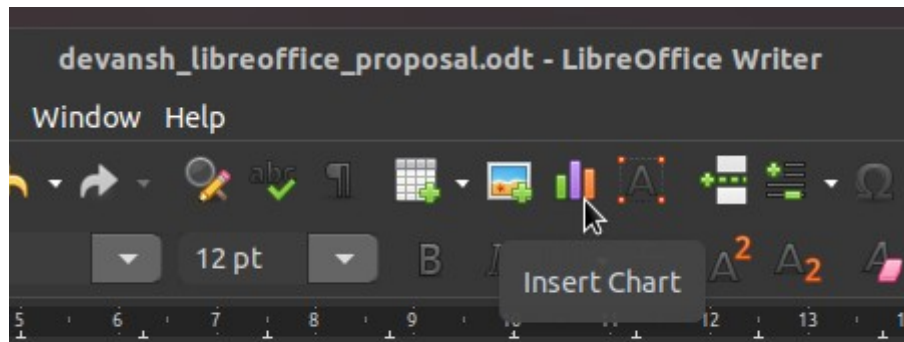


Fig1: Insert Chart in the LibreOffice UI

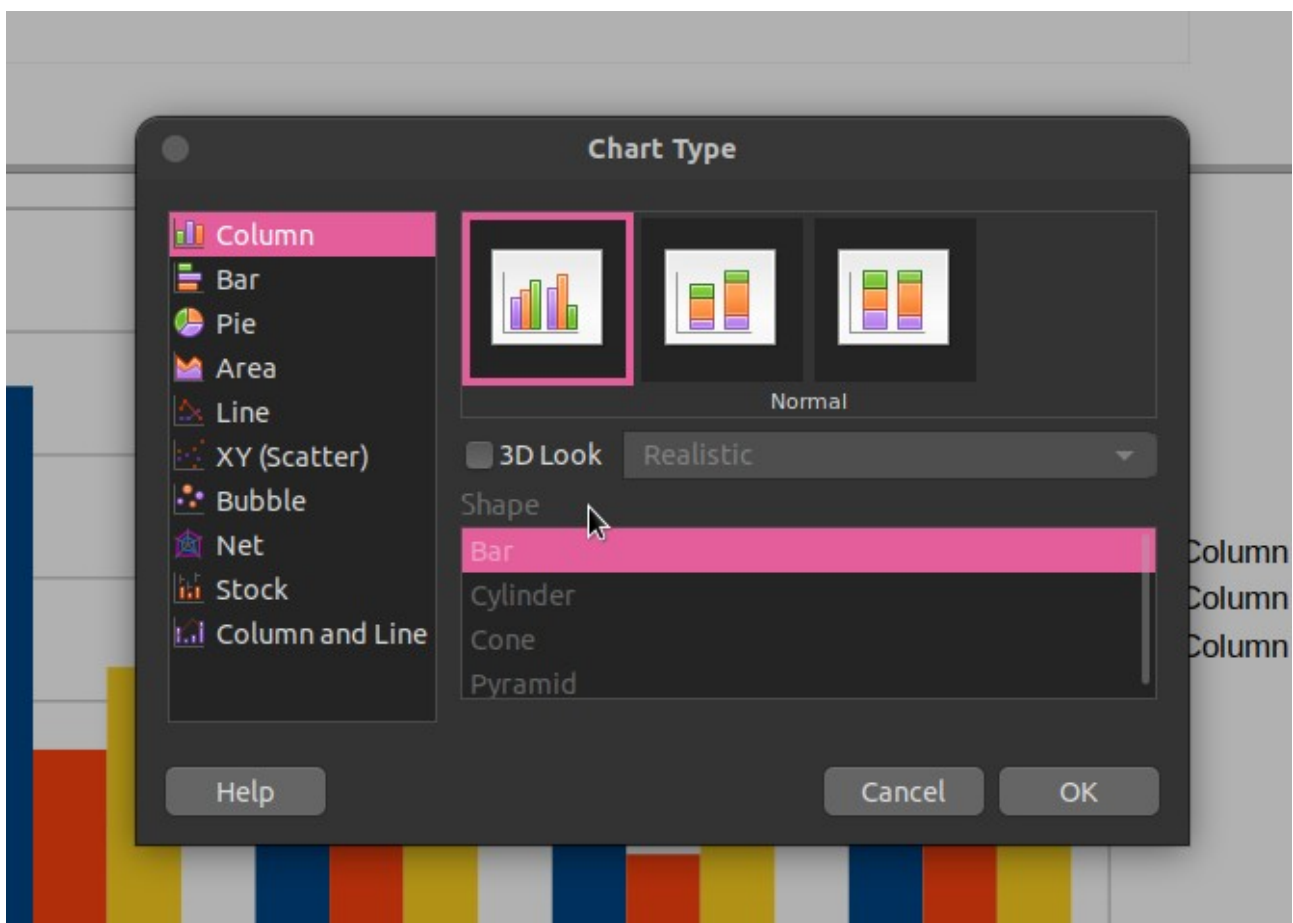


Fig2: Chart Types in the LibreOffice

B) Do LibreOffice have support to these latest chart types?

Currently LibreOffice do not support these new type of charts. But, there is some clunky workaround for really few new types using the already existing chart types. Which can be thought of as a composition of mathematical functions.

There is a case where can draw a histogram in LibreOffice with a workaround where data rows need to be repeated to achieve the desired bin widths. This approach has limitations and can be cumbersome, especially when dealing with large datasets or when the bin widths cannot be evenly divided by a common unit.

We could use the frequency() function provided. It's an "array" function, meaning you need to highlight the range of cells you want filled with the results.

[Here's a FAQ on this function. 937](#)

Basically, you need to start with a column listing your data (e.g. scores on an exam?), a 2nd column with bins (e.g. 0, 10, 20, ..., 80, 90, 100), and a third column for frequencies. Highlight the cells in your frequency column (should be one less cells than your # bins, since the bins indicate the range for each category) and enter the formula as follows:

=frequency(data,classes)

"data" are your data that you want a to create a histogram to describe. "classes" would be the bins you've chosen to specify the ranges for each category.

If you've entered the formula correctly you just need to plot the values it gave with a bar chart. It might take a bit of fooling around, but it works well, and it's a very effective way to make histograms of all types.

In the below figure we are trying to fit the input of histogram using the Column chart to somehow get the Histogram's output. Hence, we are kind of composing the Histogram in Column Charts function.

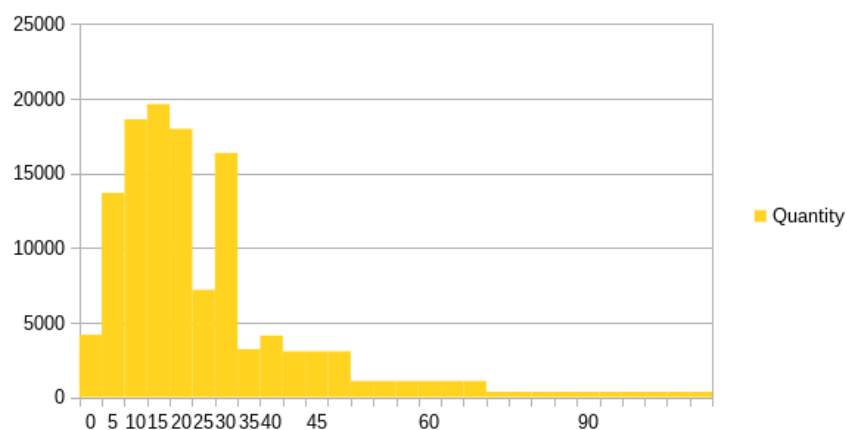


Fig3: Histogram drwan by repeating column chart

C) But what are the inputs for the histogram chart?

A histogram is basically a graph of frequencies vs. categories (i.e. ranges of values indicated by bins). The next obvious question which arises is what could be the possible inputs and are there any requirements to them? Also bins are the intervals or ranges into which the data is divided for analysis and visualization. Each bin represents a specific range of values, and the height of the bar in the histogram corresponds to the frequency of data points falling within that particular bin.

Input Types:

1. Data (required):

- *Example:* A range of cells containing numeric values, such as A1:A100.
- *Edge case:* Empty data range or non-numeric values in the data range.

2. Bin width (optional):

- *Example:* 10 (specifies the width of each bin).
- *Edge case:* Zero or negative bin width.

3. Number of bins (optional):

- *Example:* 20 (specifies the total number of bins).
- *Edge case:* Zero or negative number of bins.

4. Bin range (optional):

- *Example:* 0 to 100 (specifies the minimum and maximum values for the bins).
- *Edge case:* Minimum value greater than the maximum value.

5. Frequency or proportion (optional):

- *Example:* Frequency (counts the number of data points in each bin).
- *Edge case:* Choosing proportion when the total number of data points is zero.

6. Cumulative frequency or cumulative proportion (optional):

- *Example:* Cumulative frequency (calculates the running total of data points up to each bin).
- *Edge case:* Choosing cumulative proportion when the total number of data points is zero

Input Restrictions and Limits:

1. *Data Type:* The input data for a histogram should be numeric. Non-numeric data, such as text or categorical values, cannot be directly used to create a histogram.
2. *Range:* The input data should fall within a reasonable range. Extremely large or small values may require special handling or scaling to ensure proper representation in the histogram.
3. *Bin Width and Number of Bins:* The bin width and number of bins should be positive values. Zero or negative values are not valid.
4. *Bin Range:* If a specific bin range is provided, the minimum value should be less than or equal to the maximum value.

D) *Are there any variations to the Histogram Chart in MS Office 2016?*

Yes. There are some variations -

1. **Simple Histogram:** It displays the distribution of a single dataset by dividing it into bins and showing the frequency or count of data points in each bin. The bins are represented by vertical bars, and the height of each bar represents the frequency or count.
 - **Example:** A histogram of test scores with bins of width 10 (0-10, 10-20, 20-30, etc.).
 - **Edge case:** All data points falling into a single bin. Data with extreme outliers that significantly extend the range of bins.
2. **Pareto (Sorted Histogram):** A Pareto chart is a sorted histogram that displays the bars in descending order of frequency or count. In addition to the bars, a Pareto chart includes a cumulative percentage line that shows the running total percentage of each category.
 - **Example:** Analyzing customer complaints by category to identify the most frequent issues.
 - **Edge case:** All categories having equal frequencies or counts. A large number of categories with low frequencies, making the chart difficult to interpret.
3. **Cumulative Histogram:** A cumulative histogram displays the running total of frequencies or counts instead of count for each bin individually. The bars in a cumulative histogram represent the cumulative frequency or count, and the height of each bar represents the total frequency or count up to that bin.
 - **Example:** A cumulative histogram of website traffic showing the total number of visits up to each bin.
 - **Edge case:** All data points falling into the first or last bin. Large gaps between consecutive bins, leading to abrupt jumps in the cumulative frequency or count.
4. **Histogram with Normal Curve:** This variation of the histogram chart includes a normal distribution curve overlaid on the histogram bars. The normal curve helps assess how closely the data follows a normal distribution.
 - **Example:** Comparing the distribution of heights in a population to a normal distribution.
 - **Edge case:** Data that is severely skewed or has multiple peaks, deviating significantly from a normal distribution. Insufficient data points to reliably fit a normal curve.
5. **Stacked Histogram:** A stacked histogram allows comparisons between different subsets or categories within the data. Each bin is divided into segments representing different categories, and the height of each segment represents the frequency or count for that category within the bin.
 - **Example:** A stacked histogram comparing the distribution of grades for different classes or survey responses across different age groups.
 - **Edge case:** Negative frequencies or counts, which cannot be properly represented in a stacked histogram. Some categories having significantly higher frequencies than others, dominating the stacked histogram.

6. *Side-by-Side Histogram (Clustered Histogram)*: A side-by-side histogram, also known as a clustered histogram, displays multiple histograms next to each other for easy comparison. Each histogram represents a different subset or category of the data, and the bins are aligned vertically.
 - **Example**: Side-by-side histograms comparing the distribution of ages for males and females.
 - **Edge case**: Unequal number of bins or bin ranges between the histograms being compared. Overlapping or cluttered bars when the histograms have similar distributions.
7. *Frequency Polygon*: A frequency polygon is a line graph that connects the midpoints of the top of each bin in the histogram. The frequency polygon can be overlaid on top of the histogram bars or displayed separately.
 - **Example**: Comparing the distribution of sales data across different months
 - **Edge case**: Data with missing or inconsistent intervals between bins. Bins with zero frequency, resulting in breaks or disconnected points in the frequency polygon.
8. *Smooth Histogram*: A smooth histogram applies a smoothing function to the histogram bars, creating a smooth curve instead of discrete bars. The smoothing is achieved by fitting a curve to the histogram data points.
 - **Example**: Visualizing the distribution of response times for a system.
 - **Edge case**: Data with sharp peaks or valleys that are not well-represented by a smooth curve and oversmoothing that obscures important features or details in the data distribution.

E) Related Feature Requests?

Yes. There are some feature requests which were made over the ask.libreoffice.org and being filed in the BugZilla also.

<https://ask.libreoffice.org/t/plotting-histograms-in-calc/2914>

<https://ask.libreoffice.org/t/how-do-i-make-a-frequency-density-unequal-bin-width-histogram/4467/2>

<https://ask.libreoffice.org/t/percentage-based-histogram/3161>

BZ -

[#tdf82716](#) - Ability to create histogram chart directly from data, not using FREQUENCY()

[#tdf74665](#) - Ability to generate a histogram from non-pre-binned data

[#tdf81530](#) - Support frequency-density histogram (column chart type) i.e., histogram with variable bin widths

[tdf#103561](#) - New chart type: heat map

Chart related BugZilla list for the LibreOffice-

https://bugs.documentfoundation.org/buglist.cgi?bug_severity=enhancement&bug_status=UNCONFIRMED&bug_status=NEW&bug_status=ASSIGNED&bug_status=REOPENED&bug_status=RESOLVED&bug_status=VERIFIED&bug_status=CLOSED&bug_status=NEEDINFO&bug_status=PLEASETEST&component=Chart&email1=mariosv&emailtype1=substring&list_id=1713001&product=LibreOffice&query_format=advanced

This is how **CERN** one of the *biggest Nuclear Research* organisation **has to do a workaround for the histogram** -

<https://indico.cern.ch/event/977476/contributions/4204938/attachments/2191304/3703670/EN-Plotting-a-histogram-in-libreoffice.pdf>

F) Related Directory and Code files -

The **chart2** directory in the LibreOffice codebase contains the implementation of various chart types and related functionalities in LibreOffice.

libreoffice/core/chart2

1. **source/** : Contains the main source code for the chart module.
 - a) **controller/** : Contains the controller components responsible for handling user interactions and chart-related commands.
 - **ChartController.cxx** - Implements the main chart controller class.
 - **ChartControllerCommandDispatch.cxx** - Handles the dispatch of chart-related commands.
 - **AccessibleChartView.cxx** - Implements accessibility support for chart views.
 - b) **view/** : Contains the view components responsible for rendering and displaying charts.
 - **DrawModelWrapper.cxx** : Provides a wrapper for the drawing model used in charts.
 - **ChartView.cxx** : Implements the main chart view class.
 - **LegendHelper.cxx** : Provides helper functions for handling chart legends.
 - **Diagram.cxx** : Represents the base class for different chart types (e.g., bar, line, pie)
 - **axes/** : Contains classes for handling chart axes.
 - **Chartview/** : Contains classes for rendering chart elements.
 - c) **model/** : Contains the model components representing the underlying data and structure of charts.
 - **ChartModel.cxx** : Implements the main chart model class.
 - **DataSeries.cxx** : Represents a series of data points in a chart.
 - **Diagram.cxx** : Represents the base class for different chart types.

- `Legend.cxx` : Represents the chart legend.
- `Title.cxx` : Represents the chart title.
- `Axis.cxx` : Represents a chart axis.
- `DataPoint.cxx` : Represents a single data point in a chart.
- `ChartTypeManager.cxx` : Manages the available chart types.
- `Template/` : Contains templates for different chart types.
 - ◆ `BarChartType.cxx` : Implements the bar chart type.
 - ◆ `ColumnChartType.cxx` : Implements the column chart type.
 - ◆ `PieChartType.cxx` : Implements the pie chart type.
 - ◆ `LineChartType.cxx` : Implements the line chart type.
 - ◆ `And other charts ....`
- d) `tools/` : Contains utility classes and functions used in the chart module.
 - `ChartTools.cxx` : Provides utility functions for chart-related operations.
 - `RegressionCurveCalculator.cxx` : Calculates regression curves for charts.
 - `InternalDataProvider.cxx` : Provides data for internal chart usage.
- e) `inc/` : Contains header files for the chart module.
- 2. `qa/` : Contains quality assurance tests for the chart module.
- 3. `uiconfig/` : Contains user interface configuration files for the chart module.

G) How are going to make changes in the codebase?

In a stable codebase, the existing code has been tested, refined, and proven to work effectively, similar to a semiconductor in a stable state. Adding new code to this stable system introduces a certain level of uncertainty or randomness, which can be compared to the concept of entropy in thermodynamics. Just like adding impurities to a semiconductor through doping to enhance its properties. Introducing new code into a stable codebase requires careful consideration and planning to maintain the overall stability and functionality of the system.

When adding support for the histogram chart in LibreOffice, we can follow an approach similar to doping a semiconductor:

1. **Start with a similar component:** By starting with the *column chart* implementation, which shares similarities with the histogram chart, we introduce changes to the codebase that are relatively close to the existing functionality. This approach minimizes the level of randomness introduced into the system, as the modifications are based on a component that is already stable and well-integrated.

2. **Gradual and controlled modifications:** Just like the controlled introduction of impurities in semiconductor doping, we make gradual and incremental changes to the column chart code to transform it into a histogram chart. Extend the existing *OOXML and ODF* import and export functionality to support the histogram chart.
3. **Thorough testing and validation:** Just as the doping process in semiconductors requires precise control and validation, adding this new functionality in our codebase demands rigorous testing and validation. Which involves comprehensive unit testing, integration testing, and manual testing to ensure that the histogram chart implementation functions as expected and does not introduce any regressions or instability to the overall system.
4. **Incremental enhancements:** Once the basic histogram chart implementation is in place and stable, we can consider adding further enhancements and variations, similar to fine-tuning the doping process in semiconductors. This may include supporting different histogram types, customization options, and optimizations based on user feedback and requirements.

Proposed order of changes of files for adding support of histogram chart:

- a) `chart2/source/model/template/ColumnChartType.cxx`
 - Start by creating a new file named `HistogramChartType.cxx` in the same directory.
 - Use `ColumnChartType.cxx` as a reference and modify the code to implement the histogram chart type
 - Update the chart type registration and creation logic
- b) `chart2/source/model/template/ColumnChartType.hxx`
 - Create a corresponding header file named `HistogramChartType.hxx` in the same directory.
 - Update the class declaration and include necessary header files.
- c) `chart2/source/view/charttypes/ColumnChart.cxx`
 - Create a new file named `HistogramChart.cxx` in the same directory.
 - Implement the rendering logic for the histogram chart based on the existing column chart rendering code.
 - Modify the data preprocessing and rendering functions to handle histogram-specific requirements.
- d) `chart2/source/view/charttypes/ColumnChart.hxx`
 - Create a corresponding header file named `HistogramChart.hxx` in the same directory.
 - Update the class declaration and include necessary header files.
- e) `chart2/source/controller/chartcontroller/ChartController.cxx`
 - Update the chart controller to handle histogram chart-specific interactions and commands.
 - Modify the command handling logic to support histogram chart features.
- f) `chart2/source/controller/chartcontroller/ChartControllerCommandDispatch.cxx`
 - Update the command dispatch logic to support any new commands specific to the histogram chart.
- g) `chart2/source/inc/ChartModel.hxx`
 - Add any necessary methods or properties specific to the histogram chart in the chart model interface.
- h) `chart2/source/model/main/ChartModel.cxx`
 - Implement the added methods or properties specific to the histogram chart in the chart model implementation.

- i) Update the necessary `makefile.mk` files in the respective directories to include the new files in the build process.
- j) OOXML (Office Open XML) and ODF (OpenDocument Format) are the two primary document formats used in LibreOffice, and ensuring seamless import and export functionality for the histogram chart is essential for interoperability and a smooth user experience. We have to Modify the relevant XML and relevant ODF parsing and generation code
- k) Testing which keeps along with adding the support and at the end documentation for the users and for the developers.

H) What are the timelines going to be for this enhancement?

1. **Coding Period - Phase 1 (May 1 - July 7)**
 - Implement the basic functionality of the histogram chart, including data binning and rendering.
 - Modify the relevant files in the `chart2/source/view/charttypes` directory to support the histogram chart rendering.
 - Examine how other chart types, such as the column chart, are implemented in terms of OOXML and ODF support in LibreOffice.
 - Conduct extensive testing of the OOXML and ODF import and export functionality.
 - Test the initial implementation and fix any bugs or issues encountered.
2. **Midterm Evaluation (July 8 - July 12)**
 - Prepare a midterm report summarizing the progress made so far.
 - Demonstrate the basic functionality of the histogram chart to the mentor.
 - Receive feedback and guidance from mentors for further improvements.
3. **Coding Period - Phase 2 (July 13 - August 18)**
 - Enhance the histogram chart implementation based on the feedback received during the midterm evaluation.
 - Add support for advanced features and customization options, such as variable bin widths and data selection.
 - Improve the user interface and integration with the chart dialog and properties.
 - Implement any additional functionality or improvements suggested by the mentor or the LibreOffice community.
 - Conduct thorough testing and debugging to ensure the stability and reliability of the histogram chart implementation.
4. **Final Week (August 19 - August 26)**
 - Prepare the final work product, including the completed code, documentation, and any necessary artifacts.
 - Write comprehensive documentation explaining the usage, configuration, and limitations of the histogram chart feature.
 - Submit the final work product and mentor evaluation.
5. **Final Evaluation (August 26 - September 2)**
 - Mentor review the final work product and provide their evaluations.
 - Address any final feedback or suggestions from the mentors.
6. If the project is selected for an *extended timeline*, continue working on further enhancements and refinements based on the feedback received.

A bit about me -

I was supposed to be an astrophysicist but ended up doing Computer Science and Engineering in 2019. Even during my bachelor's my curiosity kept me learning new things like Indian scriptures. Later during the pandemic I started learning Human Behavior and Psychology and how Indian Scriptures in Indian sub-continent has influenced peoples choices. Also, how the ancient pagan and newly formed religions shaped much of the world.

But, somewhere that curious kid who was still intrested in space and science kept looking and last year I found something amazing while doing some open-source contributions. I found a badge over GitHub for the [Mars Rover 2020 program](#) on some of the contributors profile while I also got some other badges.

<https://github.com/varshneydevansh> *My GitHub profile*

Here, are some of my contributions -

<https://github.com/udacity/devops-intro-project/pull/64> (My very first PR opened in 2018)

<https://github.com/phoenixframework/phoenix/pull/5487> (My first accepted PR in 2023)

<https://github.com/google/benchmark/pull/1647>

<https://github.com/infracost/infracost/pull/2547>

<https://github.com/infracost/infracost/pull/2577> (My best PR I guess)

<https://github.com/google/benchmark/pull/1631> (My longest discussion made on PR)

<https://github.com/gradio-app/gradio/pull/4915>

<https://github.com/google/blockly/pull/7333>

<https://github.com/google/blockly/pull/7364>

<https://github.com/timeplus-io/proton/pull/555>

<https://github.com/openml/openml-python/pull/1280>

My pending and accepted PRs for the LibreOffice - <https://gerrit.libreoffice.org/q/devansh>

This year I started working on the LO codebase and wasn't sure I chose this project. Maybe now I can see as I found adding support for Histogram chart not just challenging but somewhere connected with the pain of users as when I used to the [ML stuff](#) various charts like [violinplot](#), heatmap helped so much in understanding of data much easier. While preparing this proposal I found that CERN (organisation which I like) is having so much trouble beacuse of lack of this feature. I am now more excited then I was before as I can not only see my impact which is humangous and can play a small part in helping scientific innovation :)

<https://www.youtube.com/@kyuantym>

<https://www.youtube.com/@varshneydevansh>

<https://devanshvarshney.blogspot.com/>