

This worksheet uses a Google Colab notebook linked on ORTUS (save a copy in your own Google Drive). Random numbers are used in all tasks, and you are encouraged to play around with this input data (change size, range, distribution, ...).

1. Create and modify a **scatter plot**.

- (a) Using the `np.random.rand` function, create a 3×100 array of random values.
Hint: Read the documentation for this function to understand how it works ([link](#))
- (b) Find the index of the highest value in the 3rd variable in the 3×100 array. Using one of the keywords in the `plt.scatter` function, color that point a different color than all the others.
Hint: Read the documentation for this function to understand how it works ([link](#))
- (c) Using the `plt.text` function, place the text “highest” next to the point you colored differently.
Hint: Read the documentation for this function to understand how it works ([link](#))
- (d) Submit the plot on ORTUS (use “right click → Save image as” on the plot).
- (e) **Bonus 1:** Color each point based on its 3rd variable value.
Hint: Use `plt.get_cmap('rainbow')(...)` to get the appropriate color.
- (f) **Bonus 2:** Color each point depending on its distance from the highest point.
Hint: Use `plt.get_cmap('viridis')(...)` to get the appropriate color.

2. Create and modify a **histogram**.

- (a) Call the `plt.hist` function again, but with a larger number of bins (so that the two plots are on top of each other).
- (b) Call the `plt.hist` function a third time, but with an even larger number of bins.
- (c) Add the `density=True` option to all calls of `plt.hist` to keep the plots comparable.
- (d) Add the `alpha=.3` option to all calls of `plt.hist` to be able to see all the plots.
- (e) Submit the plot on ORTUS.

3. Create and modify a **line plot**.

- (a) Change the rolling window of 2 to something larger and plot the result.
- (b) Draw two more lines, each smoother with a larger rolling window.
- (c) Add the `label='...'` keyword to all of the four calls of `plt.plot`, giving a descriptive name for each.
- (d) Show the legend by calling `plt.legend()` just before `plt.show()`.
- (e) Submit the plot on ORTUS.

4. Discuss your results with your group members.

- (a) What would you like to change in the plots so that they look nicer? Think color, font size, shape, ...
- (b) Take a tour around python-graph-gallery.com to see other examples of plots.