



## Objective

At the end of this checkpoint, you have a functional version of your visualization mainly consisting of creating the layout and implementing at least **two** of the idioms you have selected on Checkpoint II.

## Requirements

Continue the work you started in the previous checkpoint. You are required to implement **at least two idioms**. This includes all the integration and interaction that makes them work together as part of a single, coherent dashboard.

## Deliverables

Create a **2-page document using the provided template** and submit it online, **inside a zip file including both the document and the code, in Moodle**, until two days before your class (ex: classes on Monday must submit until Friday end of day).

The **document** should describe and illustrate *only deltas from the previous checkpoint regarding*:

- The **code architecture** you have created to load and (pre)process the data, to make the charts (and their interactivity), and to handle the integration.
- The **layout of your visualization**. Include at least one image of the complete interface (yes, even the idioms you have not implemented, in which case use screenshots from Checkpoint II).
- The **data (pre)processing** you must do before the data is ready to be loaded to the charts. We are not talking about the data processing from CPI but all the remaining process you have to do for each visual idiom.
- The idioms you have implemented (**chart visualization**) and their functionality (**chart interactivity**). You must include a description with images and explain the interactivity supported by those idioms.
- The **integration** techniques used and why adding more charts is easy. How are the views linked? How does that mechanism work/will it work even when you have more idioms to link?

If there are any changes from Checkpoints I and II, describe them and justify why you did those changes using the visualization principles learned in class.

The **code** of your prototype must **include the dataset**. You should deliver in a plug-and-play fashion: we run a local server, and the prototype is ready to be tested. Name your deliverable as '**CPIV-<group number>.zip**'. For instance, 'CPIV-02.zip'. The zip should contain the report and your code. Make sure your PDF file follows the same name nomenclature.

## Penalties

- Documents over 2 pages long: **1 grade point penalty per extra page.**
- PDF and ZIP names are different: **0.5 grade points penalty.**
- Incorrect file type or name: **0.5 grade points penalty per file or name.**
- Zip with document and code uploaded after the deadline: **0.5 grade points per hour of delay.**
- Document template altered (wider margins, smaller font, etc.): **1 grade point penalty.**
- No prototype: **10 grade points penalty.**

## Tasks to perform during the lab

Show your working prototype to the professor. The professor will provide feedback. The grade will be made known one week later (see below). If you are not receiving feedback, then you must be peer assessing your colleagues' submissions.

## Grading

Your work will be graded according to the following parameters:

- **Prototype architecture (15%):** describe and illustrate *deltas* regarding the code architecture to produce the prototype.
- **Dashboard layout (15%):** present and depict *deltas* regarding the work done so far according to your Checkpoint II.
- **Additional data processing (5%):** describe *deltas* regarding your additional data processing techniques to achieve the result, e.g., `groupby` or `count` in runtime. If you do not have additional data processing, explain why.
- **Chart visualization (20%):** show *deltas* regarding the actual visualization plotting real data, with a polished look-and-feel.
- **Chart interaction (20%):** present *deltas* regarding the code modules used to achieve chart interaction.
- **Chart integration (20%):** describe *deltas* regarding the integration architecture between charts.
- **Peer assessment (5%):** grade your peers' submissions in the class.

**An important note on grading for this and all other Checkpoints:** always *justify* your choices, based on the basic principles you have learned in this course (adequacy of channels to data types, human perception, etc.). Don't just describe what it is, but especially *why* it is as it is.

## Additional Notes

After you deliver your document, your work will be graded. HOWEVER, this grade **can be improved by up to two grade points** if you correct any faults pointed out by the professor and submit a revised version of the document HIGHLIGHTING THOSE CHANGES up the beginning of the class taking place 7 days after you receive feedback in class. **Only highlighted changes will be considered.**