Metaculus Coding Challenge

This is a **4-hour coding challenge**. The task is to create a **probability distribution drawing web app** that allows users to create forecasts by drawing (as on paper) probability distributions.

You are **required** to use Javascript/Typescript for the frontend, and Python for any backend. We **recommend but do not require** that you use d3 for the drawing, React for the frontend, and Django for the backend. Prioritize the quality of your solution over using these frameworks.

If you'd like, you can fork this Replit with React + Django, and we'll evaluate your solution there. Or, if you prefer other frameworks and/or your local environment, write your solution locally, and email it to us. If you choose this option, make sure to include a README with instructions to run your application, say with `python manage.py runserver` or `npm run dev`.

Further guidelines are based on which role you are applying for:

Front-End	**Full-Stack**	**Back-End**
You may choose to not build a Python backend, you may use JS/TS only, but your application must still be functional.	Find your ideal balance.	Your front-end can be minimal, i.e. just numerical text input for the drawing, with your focus on data models, distribution math, & API endpoints, etc.

Application Requirements:

- A way to draw/enter a new probability distribution. Bonus: if the user hits the delete key, the last stroke should be removed (or last action reversed).
- Ability to title and save their distribution.
- Ability to navigate to already-entered probability distributions and see them.
- A way to edit an existing distribution, and save your changes.

Evaluation Criteria

You will be evaluated **primarily on functionality**, **secondly on design**, and **thirdly on quality of code**. We don't expect you to do everything perfectly. Focus on what you consider to be the most important. For example you might try to figure out the math behind the distributions, or focus on a good drawing experience.

At the 4 hour mark, take some quick notes on what you'd improve next if you had more time. If you worked in Replit, make sure it's functioning, and if you worked in your own environment, zip your solution (along with a README on how to run it) and email it to us.

Thank you very much for applying to work at Metaculus! We look forward to your solution.