CLAUDE CODE MEETUP

The “Baygentic” Workflow

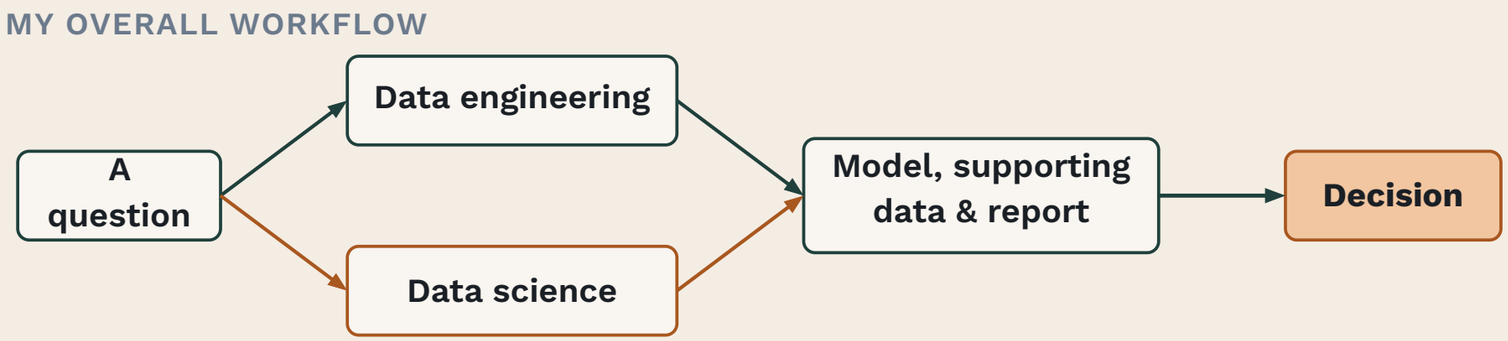
Using agents to automate the model development workflow.

Mason Veilleux · Anchorage · Sept 11, 2026

Why Bayes? “Baygents”? What’s a Baygentic workflow? Why should you care?

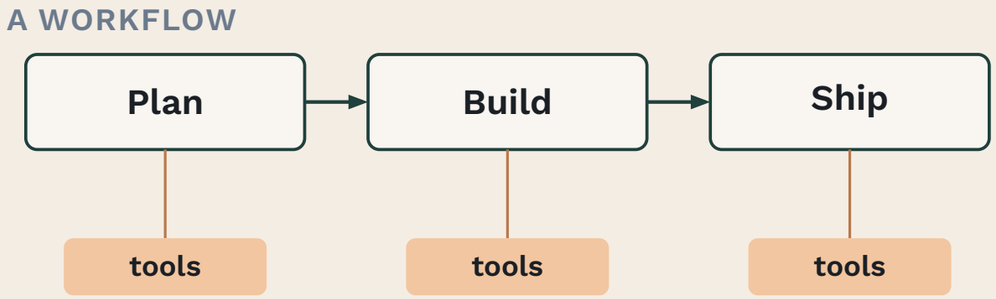
- **The thesis of this talk:** If your goal is to make an optimal decision given some set of choices and data, and the risk of being wrong is high, you want the best model of the world that reflects precisely the uncertainty around each choice.
- With a “Baygentic” Workflow, you don’t have to settle for approximations or black boxes.
- Agents can try all types of model specifications, weed out the bad ideas faster so that you can get closer to a generative, real-world model for you to make the optimal decision.

My general workflow



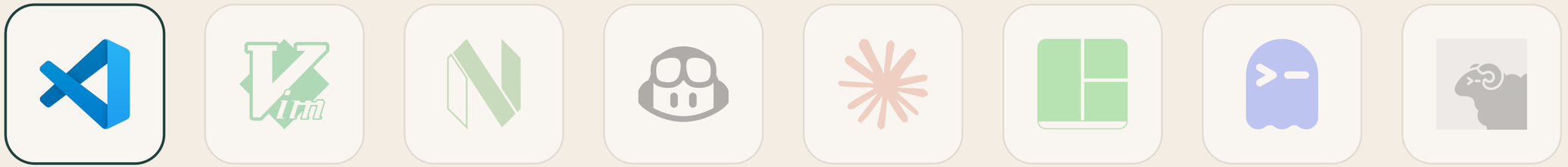
Every question I get starts one of two workflows — **data engineering** or **data science** — and both feed the same thing: a model, its supporting data, and a report that drives a **decision**.

A workflow is tasks done with tools



Every workflow breaks down into **tasks**. Every task leans on **tools**.

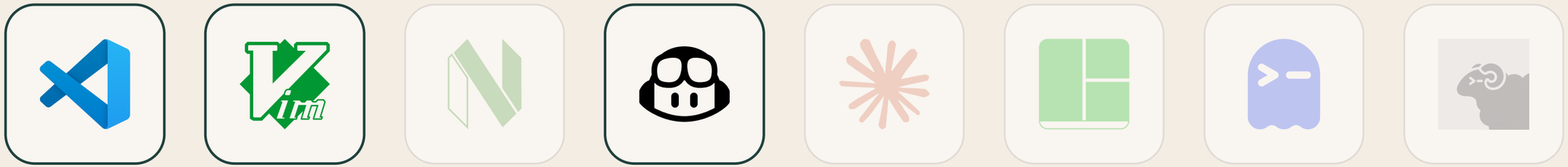
VSCode, point and click



VS Code · Vim · Neovim · Copilot · Claude · tmux · Ghostty · Herdr

Where it all started — mouse-driven, point and click.

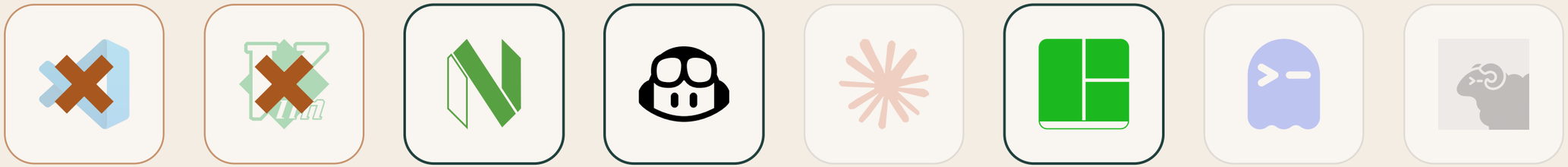
VSCode + Vim motions



VS Code Vim Neovim Copilot Claude tmux Ghostty Herdr

Custom config, Vim motions — now with tab-complete AI.

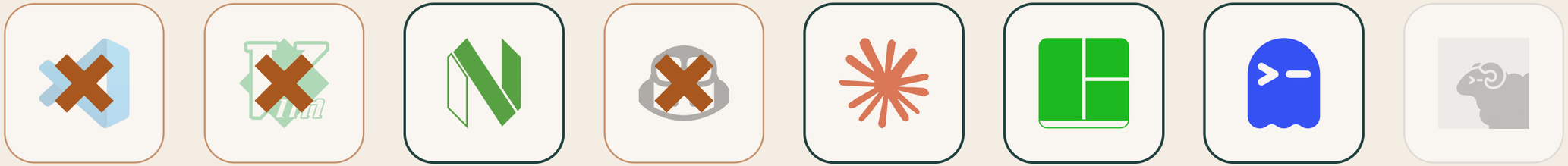
NeoVim + tmux + Copilot



~~VS Code~~ ~~Vim~~ **Neovim** **Copilot** Claude **tmux** Ghostty Herdr

> *Lots of custom configuring — genuinely hard to finish what's on your plate* ***and*** *learn the tooling at the same time.*

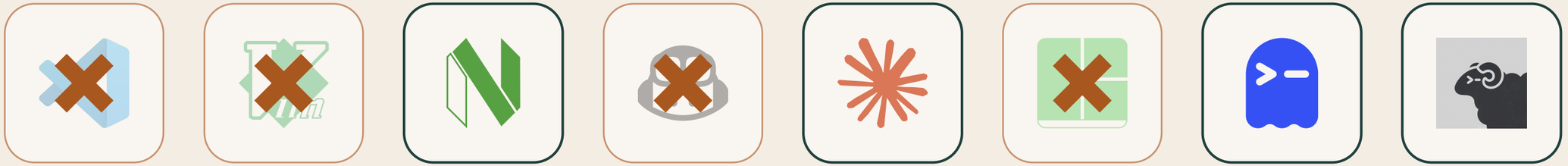
NeoVim + integrated AI + TUIs



~~VS Code~~ ~~Vim~~ **Neovim** ~~Copilot~~ **Claude** **tmux** **Ghostty** Herdr

AI moves into the editor and the terminal at once.

Today



~~VS Code~~ ~~Vim~~ **Neovim** ~~Copilot~~ **Claude** ~~tmux~~ **Ghostty** **Herdr**

- Neovim, Claude Code, Ghostty — and **Herdr**, an AI-native multiplexer for managing all my agents. tmux for agents, basically.
- Basically: managing **context** across agents.

Tomorrow?

idk, pls make it stop.

Two workflows, one project

Data engineering — a workflow, and its own tasks:
ingest → transform → schema → serve the model.

Data science — a workflow, and its own tasks:
EDA → model development → model output & monitoring.

> *AI has changed the **nature** of how we complete each of these tasks.*

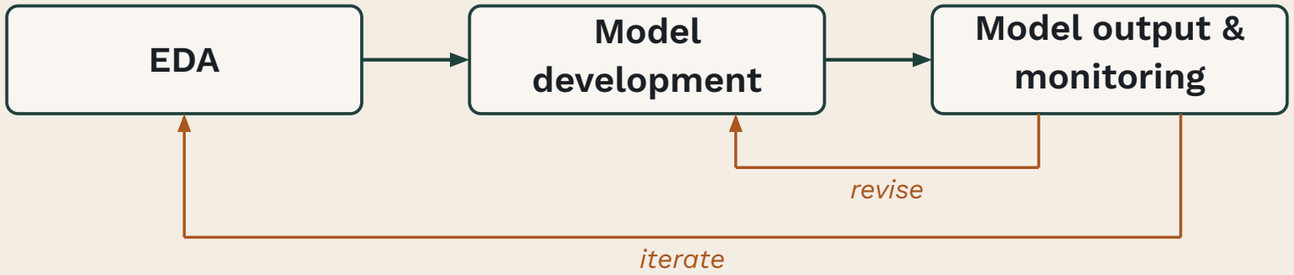
> *The main focus of this talk is on the data science workflow, and how agents have helped me automate the grunt work of model development.*

About me

- **Senior Data Engineer & Data Scientist:** I develop and maintain data pipelines and models for decision science.
- ConstructionSoftware.ai — migrating legacy systems to modern, AI-enabled stacks in rebar manufacturing.
- Run my own consultancy: database performance, forecasting, decision optimization under uncertainty, and other probabilistic modeling for Alaskans.
- Also have a wife and kid, enjoy cooking, and am definitely a normal person.

There's probably a few open questions I'm anticipating — let's clear some things up.

The data science workflow

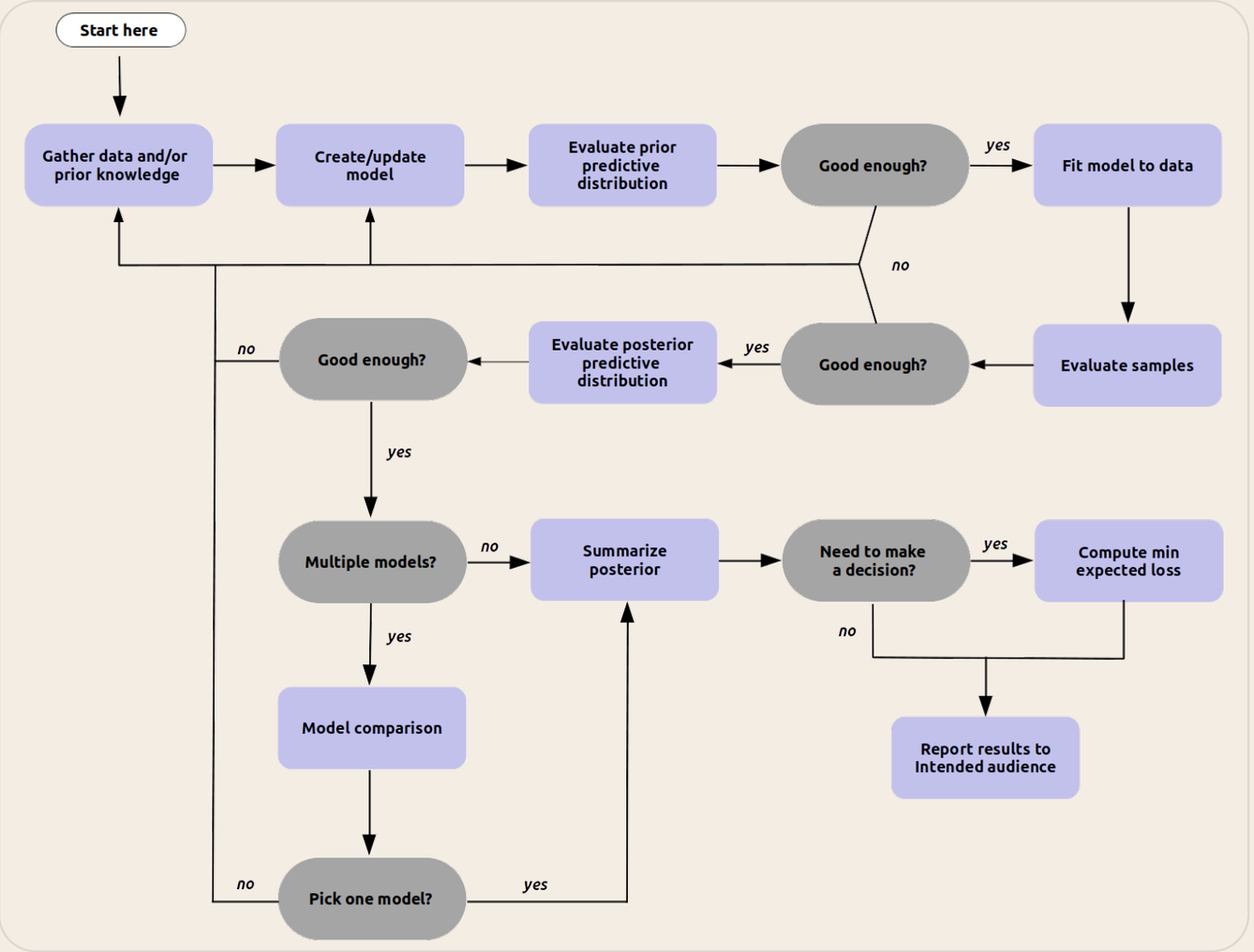


> *Lots of pausing, waiting for sampling, leaning back to think and research.*
> *Constant context switching — inspiration, a crashed run, a fix, repeat.*

Agentic data science

- Agents can automate huge chunks of the exploratory phase — armed with data-engineering context and source tables.
- We go from **zero to one** on a data science model, without hand-holding.
- Today's talk: how we automate the **Bayesian workflow** itself, with Claude Code.

This is the Bayesian workflow — we'll walk through it together.



Agentic data science with a Bayesian twist

- The goal is to build a **generative** model of the data-generating process.
- What makes a generative model? Two requisites:
  - Your posteriors should reasonably resemble the distribution you're trying to model.
  - You should be able to include, organize, and interpret all available information in your model.
- Agents can be given skills and harnessed to travel through the Bayesian Workflow with more grit than anyone is willing to admit (humans need a lot of grit to get through this workflow as it is).

Instead of explaining in theory how a generative model should behave — let's go through an example: modeling NFL wins.

Let’s build one together: NFL wins

> *What causes a team to win?*

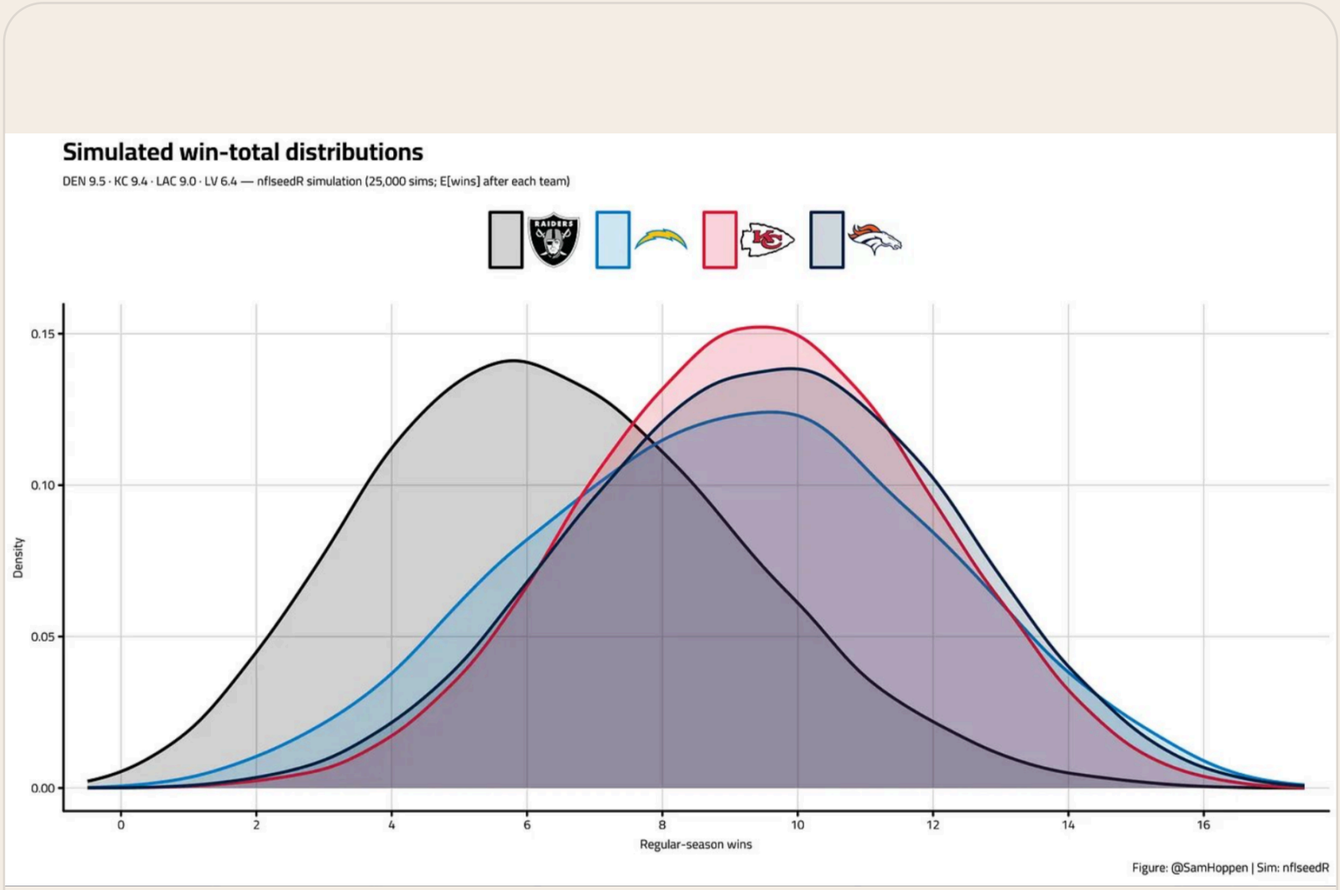
We have information — and we want to transform it into an understanding of how wins actually occur.

- at what level? drives? quarters? games? season?
- continuous or discrete?
- negative values possible?
- is there a maximum?
- split by offense & defense, or an overall effect?
- bye weeks & rest?
- an injured QB — how injured?
- a hot streak — enough info to model it?

Data science in the wild

Someone shared their season-wins analysis online — what do we notice?

Directionally sensible ranking — but the **shape** smuggles in an assumption of normality.



The cost of being wrong: sub-optimal likelihood

- Approximations — usually a normal distribution — are fine as an initial stage in the model, or for academic purposes.
- But if you had a significant amount of money on the line, or lives — would you be okay with just an approximation?
- **Baygents** make it easier to move from an approximating likelihood to one that better reflects reality — harnessed by the Bayesian workflow.

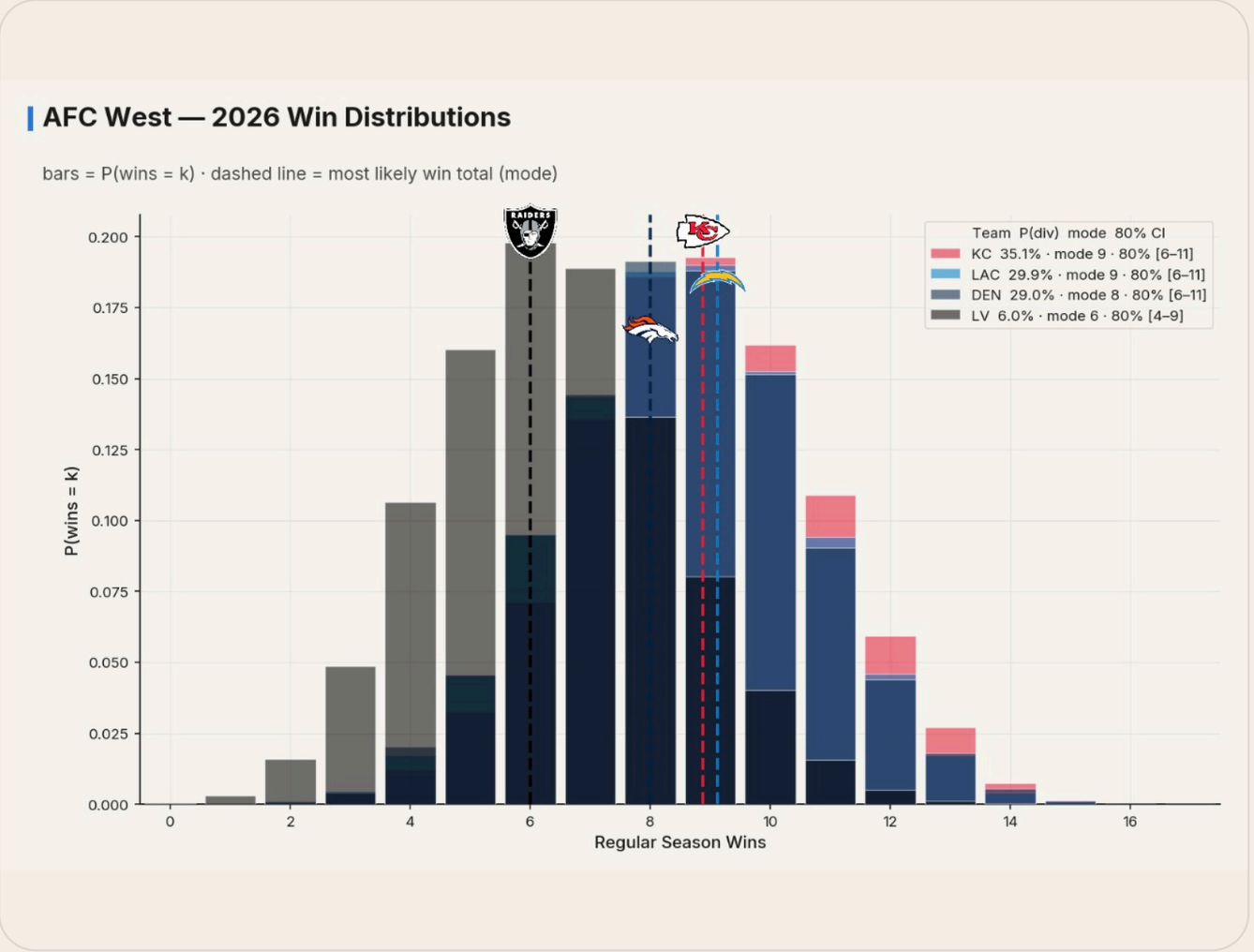
The cost of being wrong: including, organizing, and interpreting all information

- If the goal is strictly prediction, many would go straight for a black box — typically a tree-based method like XGBoost.
- But what if we wanted to generate **counterfactuals**?
- What if there's a specific structure — a **hierarchy** — in the data that we want to inform the model about?
- A causal model can be just as good as a predictive one at prediction — and we get incredible benefits in understanding how all the variables, data, and information interact, and their effects on the outcome.

How we'd actually build a generative model of NFL wins

- You can model scores by **drive outcomes**, field goals, conversions, and extra-point conversions — including the choice of one or two, given the state of the game.
- Then simulate games given those models.

Check out GamePlan Science for more information on this.



We’d rather have the real thing

- The Bayesian workflow can be automated by agents — to help us iterate and review, verify, and throw away the variety of plausible model specifications.
- The right model specification is one that reflects the richness of the data, without leaning on approximations or un-interpretable effects.
- Higher-stakes decisions should demand sharper models — ones that reflect the world in which that decision will be made.

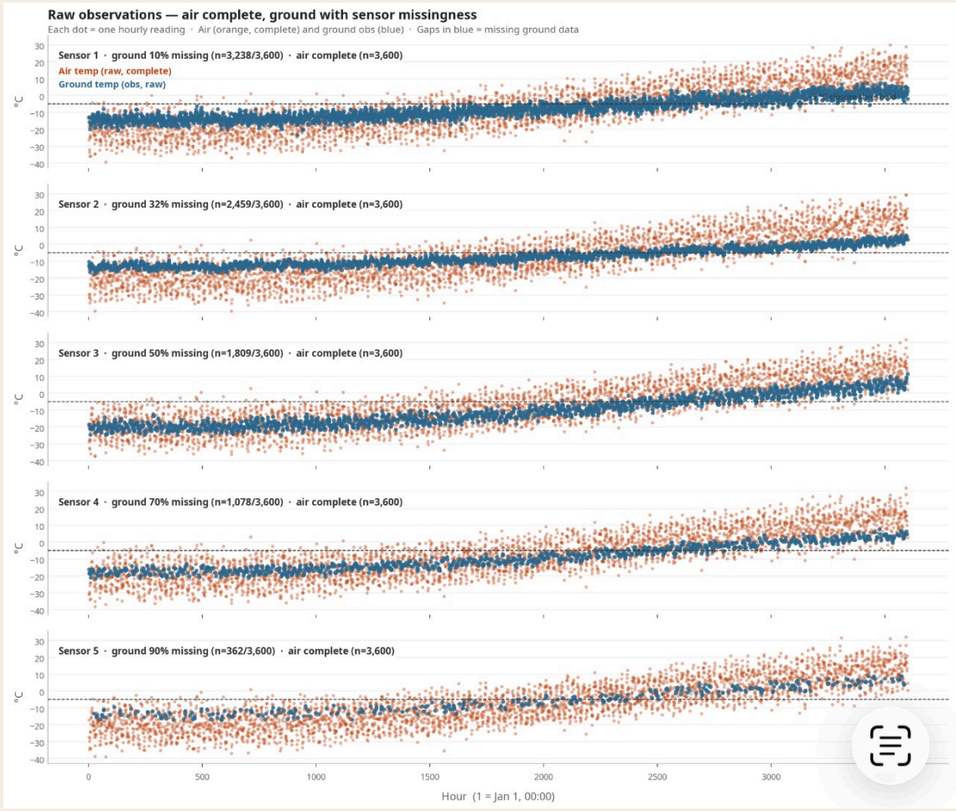
The Spring Thaw Example

- We’re in charge of deciding when to close the ice road for spring thaw.
- Close too early → lose (a lot of) money. Close too late → potential death. Once closed, no re-opening.
- The shut-off rule: ground temperature −5°C (27°F) at 0.3 m (1 ft).
- Each road has a ground-temperature sensor.
- Air-temperature sensors from a third party aren’t co-located with the ground sensors — but they’re reliable.

What does the raw data tell us?

> *What do we see in the raw observations?*

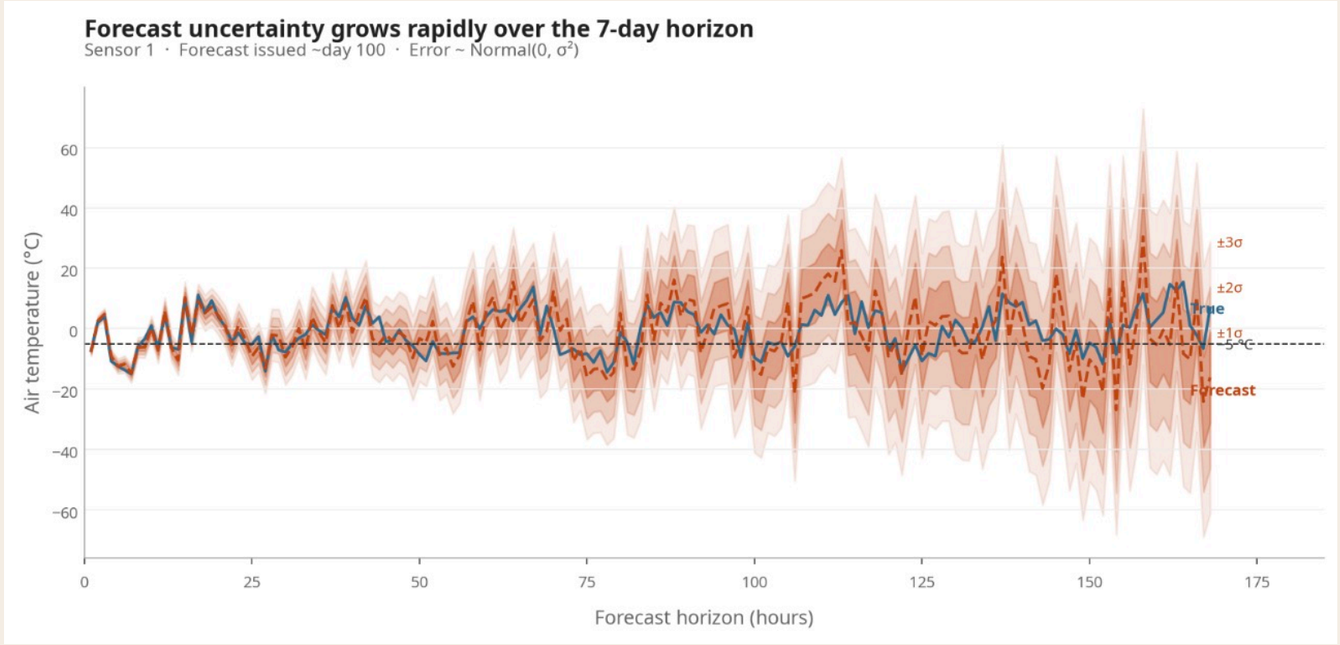
- long-run trends?
- short-run trends?
- different variances across sensors?
- different sensor locations — how do we combine them?
- measurement bias between sensors?



Plot shows ground sensor data with faulty POSTs to the remote server. We have perfect air temperature, as well as air temperature forecasts. Data is hourly.

What about forecasting and projecting?

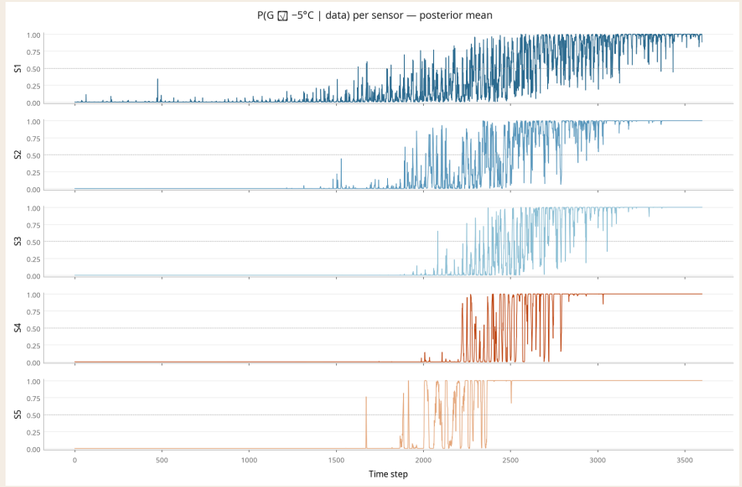
- We don’t just have observations — we have **forecasts**, and forecasts carry their own uncertainty.
- That uncertainty has to **propagate** into the model, not get thrown away at the point estimate.
- The farther out the forecast horizon, the wider the error band — and the less confident the shut-off decision should be.



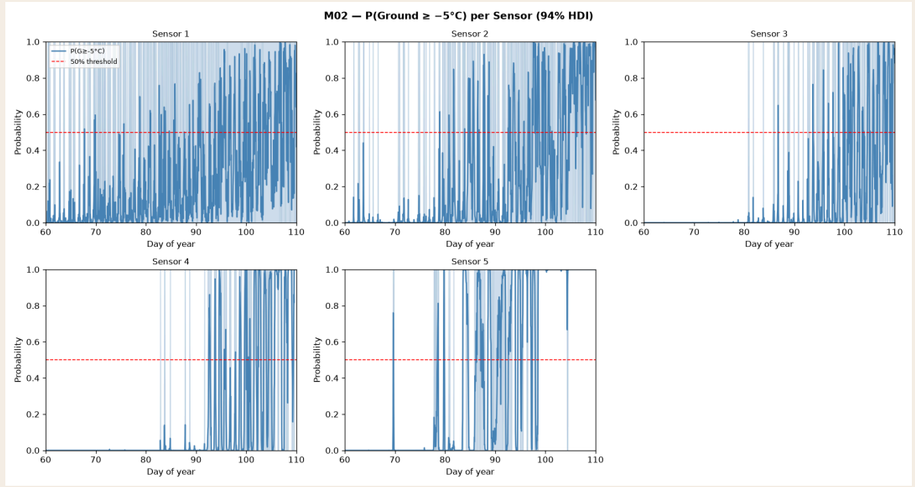
Forecast uncertainty grows with the horizon — the further out we project, the wider the error band around the true value.

Results — make your choice

This is what the posteriors are doing: once P(ground ≥ −5°C) crosses the threshold, you have your answer.



Posterior mean of P(ground ≥ −5°C) per sensor, over the full run.



Same probability, zoomed to the thaw window, with 94% HDI and the 50% threshold.

Got ‘em

While that runs: yes, this is a MacBook. No, that’s not macOS.

- Incredibly flexible and **agent-first** — see the icon up top tracking agent usage.
- Beautiful, with flexible theming.
- Lots of plugins and a rich community.
- This Mac used to die within 10 minutes unplugged. Under Omarchy, I get 4+ hours.
- Everything I need out of the box: Docker, dev tools, Neovim, AUR + package management.

Recap

- Our workflows have changed dramatically in the last year or so — adopting new tools is vital to better harnessing agents, from Claude Code to Omarchy.
- The model development workflow, for me, has improved dramatically — it helps me quickly throw away bad ideas and get to a sharper model that reflects the real world.
- Making data-informed, high-risk decisions should demand higher-resolution models — and we can use agents to help us iterate through the model development phase.

THANKS

Q & A

Mason Veilleux · ConstructionSoftware.ai · Anchorage

We are all Bayesians now.