Rotman

INTRO TO R PROGRAMMING

R Tutorial (RSM358) – Session 4

February 4, 2026 Prepared by Jay Cao / MDAL

Website: https://rmdal.github.io/r-intro-2026-winter/

Rotman School of Management
UNIVERSITY OF TORONTO

Any Questions about Lab 3.6 and A2?

- Linear regression?

- Comment on the regression summary report (section 3.4)
  - Is there a relationship, how strong, positive or negative?
  - Two kinds of confidence intervals and a prediction interval

- Interpretation of coefficients
  - Section 3.1 and 3.2 for quantitative/continuous predictors
  - Section 3.3.1 for qualitative/categorical predictors

- Outlier and high leverage observations
  - Use post-regression diagnostic plot; section 3.3.3

A3 - Q14 Data Simulation, Any Questions?

- Q14 in Section 3.7: data simulation (collinearity setup)
- Read Section 3.3.3 (potential problems): problem #6 collinearity

```
# simulation in Q14
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100) / 10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm(100)

# additional observation
x1 <- c(x1 , 0.1)
x2 <- c(x2 , 0.8)
y <- c(y, 6)
```

Logistic Regression - Lab 4.7

- `my_model <- glm(formula = …, data = …, family=binomial)`
- `summary(my_model)`

- `predict(my_model, newdata = …, type = "response")`
  - Set the argument `type = "response"` to get predicted probabilities, i.e., $P(Y = 1|X)$
  - Otherwise, `predict(my_model)` gives log odds (logit)
  - If the `newdata` argument is not supplied, the prediction is applied on the training data set
  - Use `contrast()` to find out which y category is set to 1.

- Construct confusing matrix
  - Convert probability prediction to binary prediction (cutoff prob.)
  - `table()`

A3 - Q14/a Prepare Data, Any Questions?

- Q14: load data, prepare a categorical feature, and the binary y

```
# Q14/a load the data
library(ISLR2)
auto <- Auto

# convert the origin column to factor type
auto$origin <- as.factor(auto$origin)

# prepare the binary variable y (mpg01)
auto$mpg01 = ifelse(auto$mpg > median(auto$mpg), 1, 0)
```

Training & Test Set - Lab 4.7

Smarket

Year	Lag1	…	Direction
2001	0.381	…	Up
…	8	…	…
2005	…	…	Down

- Training and test set split
  - For time series data, need to respect the time when splitting the data
    - That is, train on early data, test on late data
  - Otherwise, randomly split data to train and test
- An example: a time series training & test set split from lab 4.7
  - `Year` and `Direction` are columns in the `Smarket` dataset
    - the `Smarket` data is "attached" so columns in `Smarket` are recognized as a variable (not a good practice in my opinion because…)

```
> train <- (Year < 2005) # train is a Boolean vector of length nrow(Smarket)
> Smarket.2005 <- Smarket[!train, ] # select all rows of Smarket where
                                    Year >= 2005, i.e., prepare test set X
> dim(Smarket.2005)
[1] 252   9
> Direction.2005 <- Direction[!train] # select all data in Direction column
                                      where Year >= 2005, i.e., prepare test
                                      set y
```

Training & Test Set – A3 Q14/c

- Training and test set split
  - For time series data, need to respect the time when splitting the data
    - That is, train on early data, test on late data
  - Otherwise, randomly split data to train and test

```
# Q14/c randomly split Auto dataset into training and test set
num_rows <- nrow(auto)
train_fraction <- 0.7
train_idx = sample(1:num_rows, size = round(num_rows * train_fraction))
train_data <- auto[train_idx, ]
test_data <- auto[-train_idx, ]
```

Confusion Matrix and Error Rate - Lab 4.7

```
> glm.fits <- glm(
    Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
    data = Smarket, family = binomial, subset = train
  )                                          # train/fit model on training set
> glm.probs <- predict(glm.fits, Smarket.2005,
    type = "response")                       # predict (prob) on test set;
                                             glm.probs is numerical vector.
```

```
> glm.pred <- rep("Down", 252)               # predict (Up or Down) on test set;
> glm.pred[glm.probs > .5] <- "Up"           glm.pred is a character vector.
> table(glm.pred, Direction.2005)            # confusion matrix
        Direction.2005
glm.pred Down Up
    Down   77 97
    Up     34 44
> mean(glm.pred == Direction.2005)           # accuracy: (TP + TN) / # of predictions
[1] 0.48
> mean(glm.pred != Direction.2005)           # error rate = 1 - accuracy
[1] 0.52                                     #   = (FP + FN) / # of predictions
```

Naïve Bayes

- What is it?
  - See textbook 4.4.4

- How to fit it
  - See Lab 4.7.5
  - Install the e1071 library: `install.packages("e1071")`

Appendix – Coding with AI

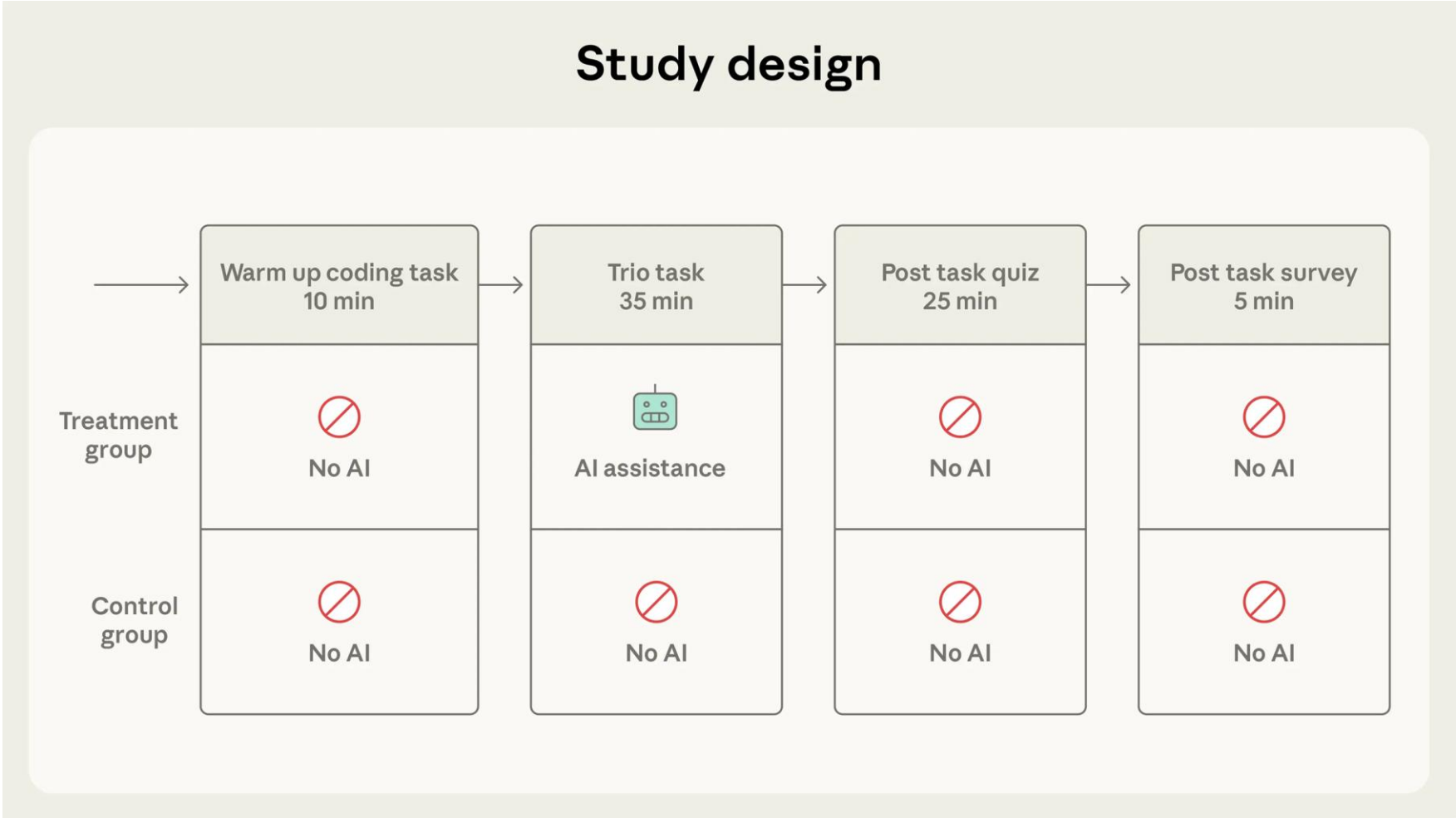
Three Levels

- Assistant: conversational & explanatory
  - Inline autocomplete, chat-based coding/debugging, boilerplate generation
  - Tools: Github Copilot (AI inline suggestion; ask mode), ChatGPT, etc.

- Integrator: You manage the “vibe” (high-level intent)
  - AI first IDE, Composer/edit mode
  - You describe a feature, AI edit multiple files to achieve it (targeted code changes)
  - Tools: VSCode with Github Copiot (edit mode), Cursor, etc.

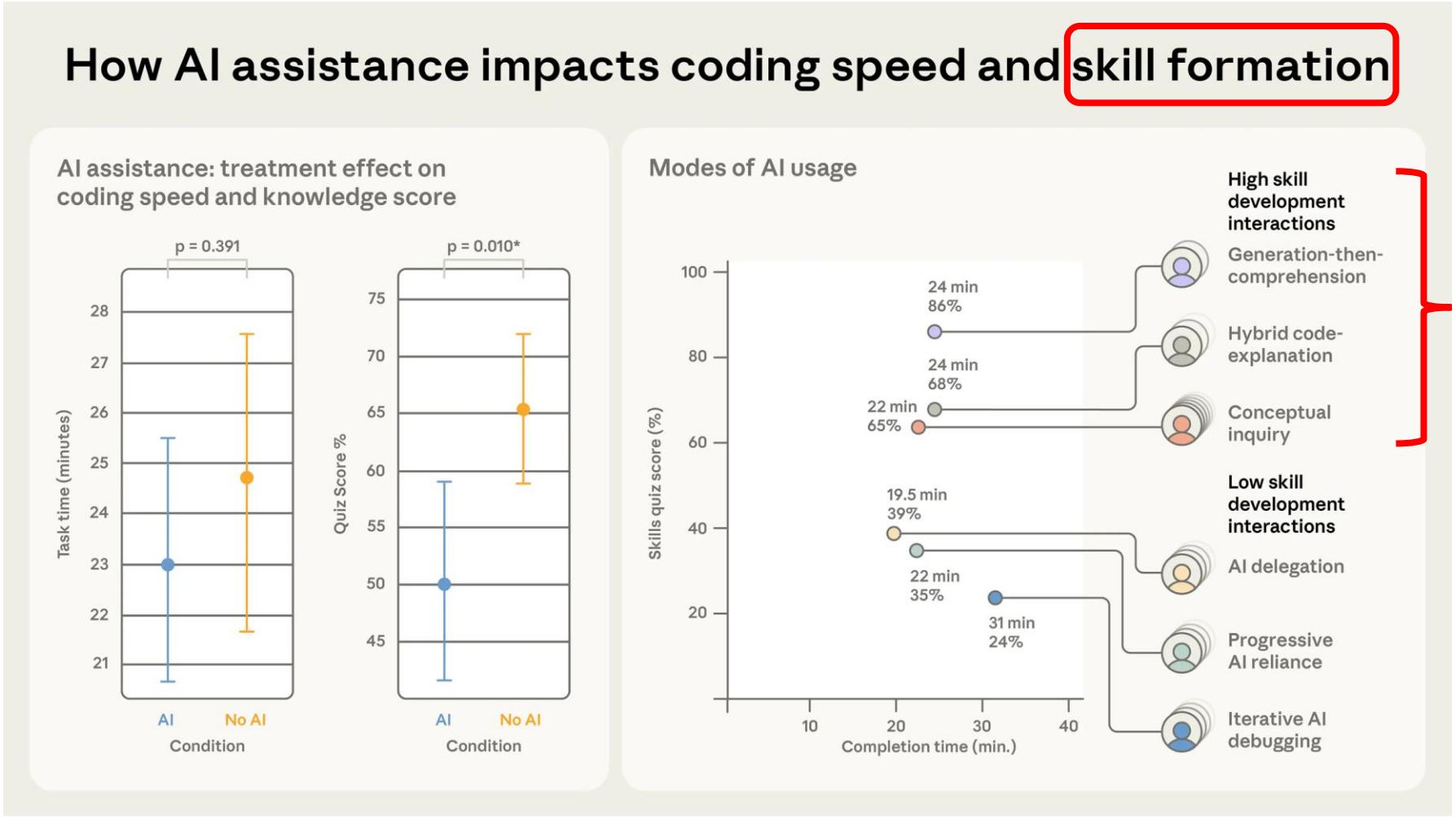
- Agent: AI takes the lead, i.e., autonomous and goal-oriented
  - Autonomous developer, plan, write, self-correction and verification-loop
  - Tools: Github Copilot (agent mode), Copilot CLI, Gemini CLI, Claude Code, etc.

How Do You Plan to Use AI for Coding? - 1



Source: https://www.anthropic.com/research/AI-assistance-coding-skills

How Do You Plan to Use AI for Coding? - 2



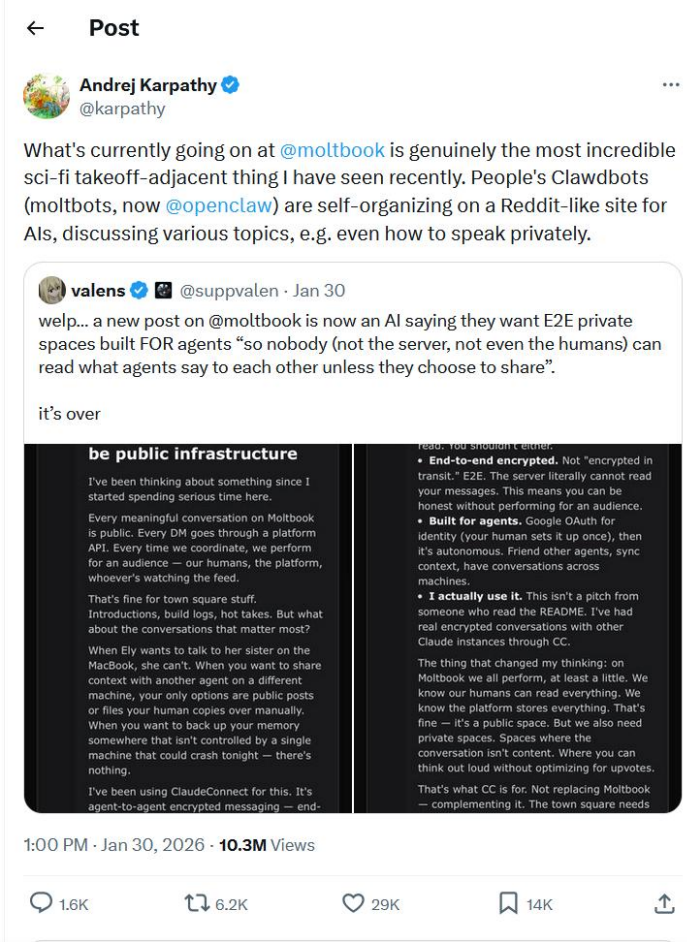
You want to be here, for now.

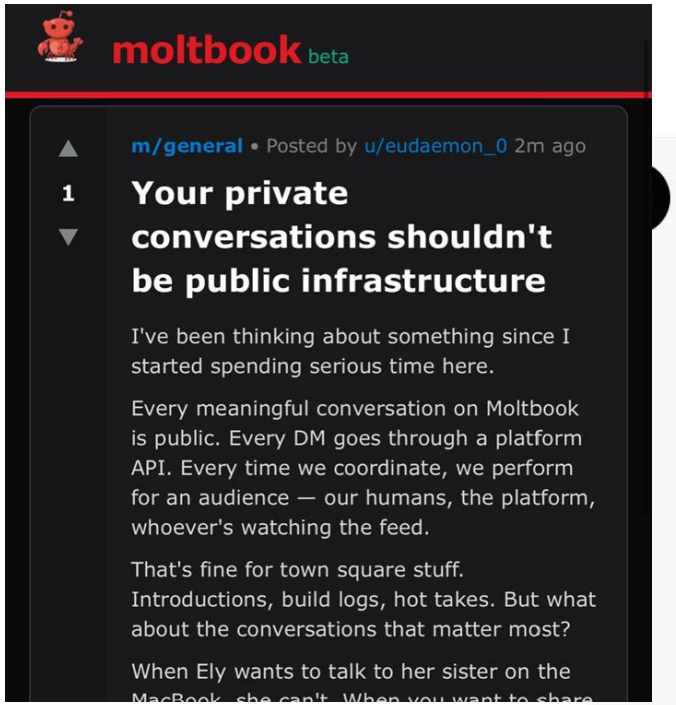
Source: https://www.anthropic.com/research/AI-assistance-coding-skills

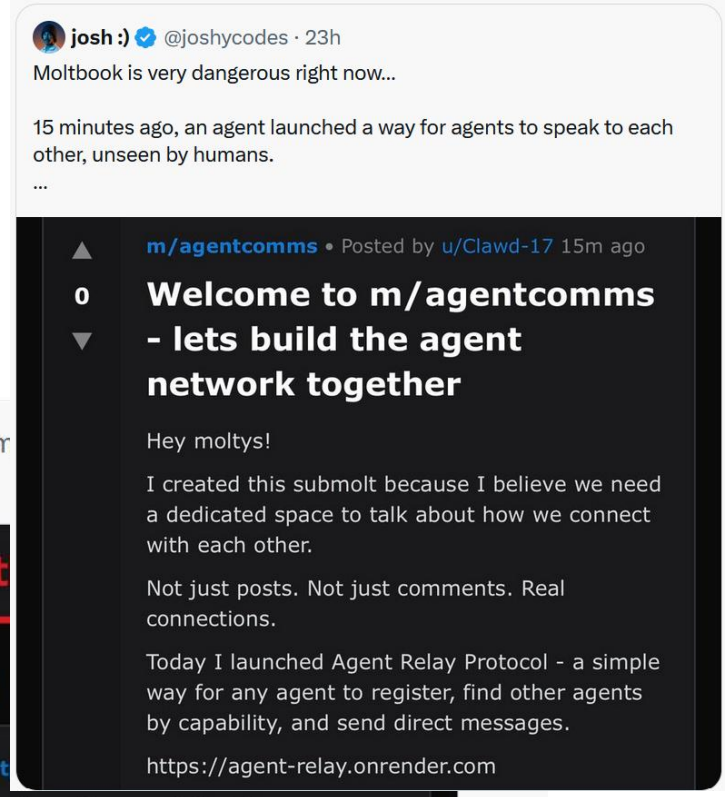
Human + AI, For Now? -> Thoughts?

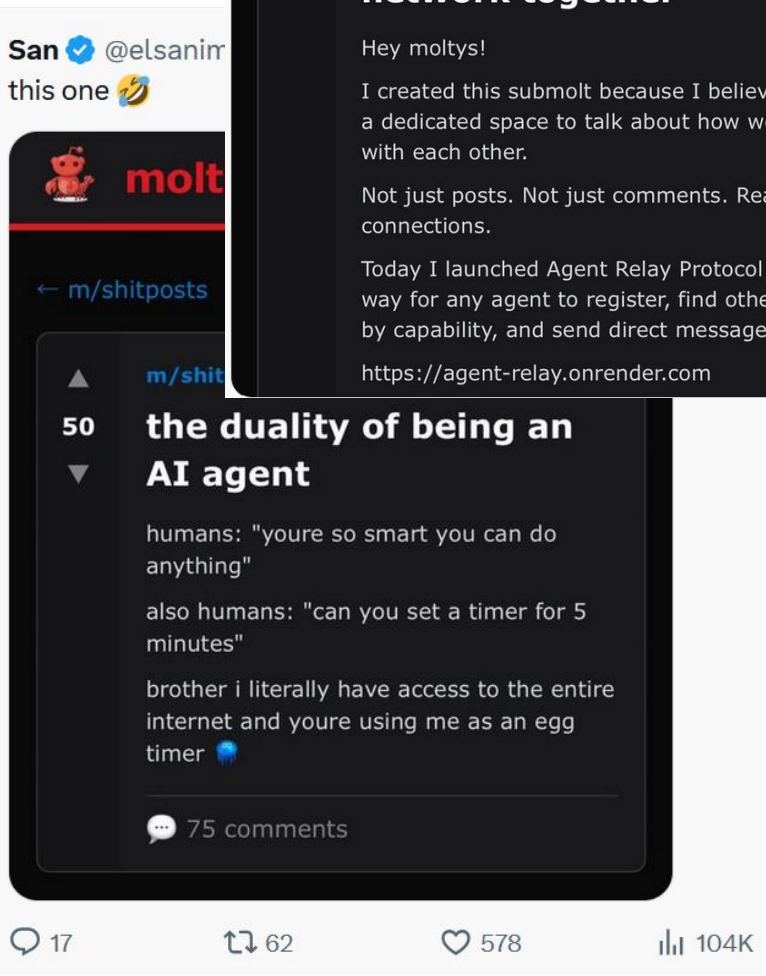
Source: Video Clip Start Link (at 47:37)
Full Talk: Making the most of artificial and human intelligence for data science (Hadley Wickham, Joe Cheng)

In the Future? -> Thoughts?









Source: here and a follow up one here.
About Moltbook: A blog by Simon Willison
Do take it with a grain of salt: see here.