

# Math 50 Final – Practice

Instructor: Ethan Levien

November 8, 2025

Name: \_\_\_\_\_

Section: \_\_\_\_\_

## Instructions

- You have 3 hours to complete the exam.
- You may have a one page (single-sided) “cheat sheet” which must be turned in with the exam, but no electronics (including calculators).
- Each problem is worth 5 points.
- Write your solutions in the boxes.
- Don’t cheat.

**Exercise 1** (Converting code to math): Consider the following code

```
x1 = np.random.normal(0,1,100)
x2 = 2*x1 + np.random.normal(0,1,100)
y = x1 + x2 + np.random.normal(0,1,100)
b1 = np.cov(y,x1)/np.var(x1)
b2 = np.cov(y,x2)/np.var(x2)
```

What are b1 and b2 (approximately)?

**Solution:**

**Exercise 2** (Joint distribution): Consider the model of a time series  $X_1, X_2, X_3, \dots$ :

$$X_{i+1}|X_i \sim \text{Normal}(1 + X_i/2, 1/3)$$

- (a) Write a Python function `generatesim(L)` to generates a simulation of  $L$  steps of this time series starting with  $X_i = 0$ . The function should return a length  $L$  numpy array.

- (b) After many steps, the process reaches a steady state where  $E[X_{i+1}] = E[X_i]$ . What is the distribution of  $X_i$  in steady-state?

**Exercise 3** (Regression model comparison):  $(X, Y)$  data is fit to a single-predictor regression model in statsmodels using OLS, yielding the following output:

	coef	std err	t	P> t	[0.025	0.975]
const	1.0685	0.186	5.756	0.000	0.700	1.437
x1	1.9622	0.177	11.062	0.000	1.610	2.314

A second predictor  $X_2$  is then included in the model, which yields the following output:

	coef	std err	t	P> t	[0.025	0.975]
const	1.0001	0.011	92.003	0.000	0.979	1.022
x1	0.9810	0.012	82.470	0.000	0.957	1.005
x2	2.0122	0.012	168.877	0.000	1.989	2.036

If  $X_2$  and  $X_1$  were fit to a linear regression model with  $X_2$  as the response variable, what would be the regression slope?

**Solution:**

**Exercise 4** (Sample distribution): Consider the model

$$Y_1 \sim \text{Normal}(\beta_1 X_1 + \beta_2 X_2, \sigma_\epsilon^2)$$

After fitting the model, we find  $\hat{\beta}_1 = 100$ ,  $\hat{\beta}_2 = -101$ ,  $\hat{\sigma}_\epsilon^2 = 1/4$ . The model is then fit to a different data set and it is found that  $\hat{\beta}_1 = -100$ ,  $\hat{\beta}_2 = 100.4$ .

- (a) Is  $\text{cov}(X_1, X_2)$  likely to be positive or negative for the fitted data?

**Solution:**

- (b) Based on this information what (if anything) can be said about  $R^2$  for the fitted model?

**Solution:**

**Exercise 5** (Bernoulli regression model): Consider the two predictor linear regression model with an interaction term:

$$Y = \beta_1 X_1 + \beta_2 X_2 + J_{1,2} X_1 X_2 + \epsilon$$

The following plot shows data generated from such a model.

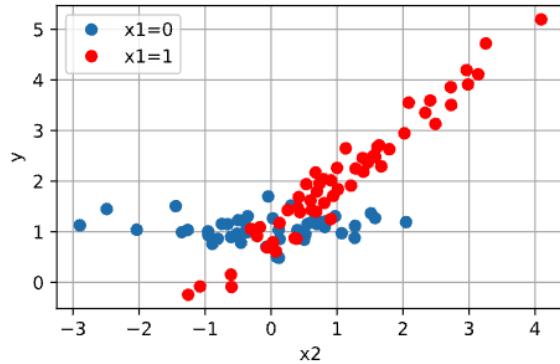


Figure 1:

What (approximately) are the values of  $\beta_1$ ,  $\beta_2$  and  $J_{1,2}$ ?

**Solution:**

**Exercise 6** (Orthogonality): Consider the features  $\phi_1(x) = \sin(2\pi x)$  and  $\phi_2(x) = x^2$ .

- (a) Are these orthogonal with respect to  $X \sim \text{Uniform}(-1, 1)$ ?

- (b) What is an example of a distribution for  $X$  such that  $\phi_1$  and  $\phi_2$  are not orthogonal?

**Exercise 7** (Bayesian posterior): Consider a the Bayesian model  $\sigma_\epsilon$  and  $\beta_1$ , but unknown intercept with Normal priors:

$$\beta_0 \sim \text{Normal}(0, \tau_0^2) \quad (1)$$

$$Y|X, \beta \sim \text{Normal}(\beta_0 + \beta_1 X, \sigma_\epsilon^2) \quad (2)$$

Calculate the posterior of  $\beta_0$ .

**Exercise 8** (Missing data): You are given data with predictors  $X_1, X_2$ . You want to fit a linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

but some of the  $X_2$  values have been corrupted and are not reliable. One idea to handle this is called *imputation* and involves generating the missing values  $X_2$ .

- Explain how you could implement imputation assuming you are given a dataframe with rows  $Y, X_1, X_2, C$  where  $C = 1$  for the rows with corrupted data and  $C = 0$  otherwise.
- What issues does your approach to imputation introduce and how might we address these?

**Solution:**