NRES 746, Week 15, Lecture 1: Filter by Hand (take-home)

Name: _____

Due Monday, Dec. 7. Work on your own or in small groups. Part 1 is on paper; Part 2 involves R. Turn in this sheet only: you'll write code along the way, but no need to submit it. The last question asks how you used AI.

A wildlife agency counts desert bighorn ewes in one mountain range from a helicopter each fall. Wet winters grow more forage, so the agency also records the previous winter's precipitation. It wants to know how strongly the herd regulates itself, and the counts are noisy: a group of ewes behind a ridge is easy to miss.

Notation and key equations

- x_t: the true log number of ewes in year t (hidden). y_t: the log of the count. w_t: winter precipitation, in SDs from its mean.
- **State equation:** $x_t = a + b\,x_{t-1} + c\,w_t + \varepsilon_t$, with process noise $\varepsilon_t \sim N(0, \sigma_p^2)$. b measures density dependence ($b = 1$: none; smaller b: stronger).
- **Observation equation:** $y_t = x_t + \delta_t$, with observation noise $\delta_t \sim N(0, \sigma_o^2)$.
- The **Kalman filter** carries a best guess $\hat{x}$ of the state and its variance P from year to year. Each year it predicts,

$$\hat{x}_{\text{pred}} = a + b\,\hat{x}_{\text{last}} + c\,w_t, \qquad P_{\text{pred}} = b^2 P_{\text{last}} + \sigma_p^2$$

  and then updates with the new count,

$$K = \frac{P_{\text{pred}}}{P_{\text{pred}} + \sigma_o^2}, \qquad \hat{x}_{\text{new}} = \hat{x}_{\text{pred}} + K\,(y_t - \hat{x}_{\text{pred}}), \qquad P_{\text{new}} = (1 - K)\,P_{\text{pred}}$$

Part 1: One year of the filter (on paper)

Suppose the parameters are known: $a = 0.96$, $b = 0.8$, $c = 0.1$, $\sigma_p^2 = 0.014$, $\sigma_o^2 = 0.02$. After the 2024 survey the filter's belief about the herd is $\hat{x} = 4.70$ (about 110 ewes) with variance $P = 0.025$. The winter before the 2025 survey was dry, $w_{2025} = -1$, and the 2025 count was 83 ewes ($y_{2025} = 4.42$).

1a. Work through the predict step and then the update step for 2025, and fill in the table.

	$\hat{x}$	**P**	**K**
after the 2024 survey	4.70	0.025	
2025, predicted (before the count)			(none)
2025, updated (after the count)			

1b. The count (83 ewes) was below the prediction. Did the filter's estimate move all the way to the count, most of the way, or only a little? What in the numbers above decided that?

1c. Suppose the 2025 survey had been flown in poor conditions, so that $\sigma_o^2 = 0.12$ for that year. What would K and the updated $\hat{x}$ be? What would K be if counts were exact?

Part 2: Fitting the model (R)

Each coding question has a label:

- *Recommended: no AI*: this week's core skill. R's help pages, web search, the Week 15 page, and classmates are fine.
- *AI optional/encouraged*: the results matter, but the code isn't the point.

Read the data with

```
bighorn <- read.csv(
  "https://kevintshoemaker.github.io/NRES-746/data/BighornCounts.csv")
y <- log(bighorn$ewes)
w <- bighorn$precip_z
```

The columns are the year, the count of ewes, and the previous winter's precipitation (`precip_z`, in SD units).

2a. *(AI optional/encouraged.)* Fit the model as if there were no observation noise: regress each year's log count on the previous year's log count and that year's precipitation.

$\hat{b} =$ _____ (SE _____)

2b. *(Recommended: no AI.)* Fit the state-space model. Copy the functions `kalman()` and `nll_ss()` from section 3 of the Week 15 companion script; both take the data as arguments, so `nll_ss(par, y = y, w = w)` gives the negative log-likelihood, with `par` holding a, $\operatorname{logit} b$, c, $\log \sigma_p$, and $\log \sigma_o$. Check: `nll_ss(c(1, qlogis(0.8), 0.1, log(0.1), log(0.15)), y = y, w = w)` should return -3.828. Then minimize it with `optim()` (`method = "BFGS"`), starting from the values in the check, and fill in the table.

$\hat{b}$	$\hat{c}$	$\hat{\sigma}_p$	$\hat{\sigma}_o$	**equilibrium, ewes**

(The equilibrium is $\exp\left(\hat{a}/(1-\hat{b})\right)$.)

2c. Which fit, 2a or 2b, says density dependence is stronger? Explain why ignoring observation noise would push the estimate in that direction.

2d. Final question: where did you use AI, if anywhere, and what did you have to check or fix? If you didn't use it, what was hardest to get working?

Bonus (optional)

3. Precipitation might instead affect how many ewes the helicopter crew sees (animals bunch up near water in dry years) without changing how many there are. `kalman()` has an argument `d` for a covariate in the observation equation. Fit that model (set $c = 0$ and estimate d) and compare its log-likelihood with the model in 2b. Which does the data favor, and by how much?