Assignment 5

Brian Sohn

1. Fitting a Mixture Model to Estimate Probabilities and Mean Number of Events

Parameters to estimate are:
p_w: probability that a person is a whale
pzero_0: probability that a person is a non-purchaser, given that a person is not a whale, when group 0
pzero_1: probability that a person is a non-purchaser, given that a person is not a whale, when group 1
mu_0: mean number of events for casual purchaser, when group 0
mu_1: mean number of events for casual purchaser, when group 1
mu_w_diff: mean number of events for whale – mu_0

Priors are set as:
p_w ~ invlogit(N(logit(0.005, tau=1)))
pzero_0 ~ invlogit(N(logit(0.85, tau=1)))
pzero_1 ~ invlogit(N(logit(0.85, tau=1)))
mu_0 ~ HN(tau=1)
mu_1 ~ HN(tau=1)
mu_w_diff ~ HN(tau=0.01)

For group i, a mixture model is fit using
X_i ~ Mixture ([f_w, f_n], [p_w, 1-p_w])
f_w ~ Poisson((mu_0+mu_w_diff)) : distribution for whales
f_n ~ ZeroInflatedPoisson(psi=1-pzero_i, mu=mu_i) : distribution for non-whales

Results from Pymc are summarized in the following table.

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
mu_0	4.305	0.076	4.163	4.443	0.001	0.001	5114.0	3296.0	1.0
mu_w_diff	59.088	1.611	56.139	62.172	0.021	0.015	5830.0	3105.0	1.0
mu_1	3.044	0.071	2.911	3.174	0.001	0.001	5551.0	3062.0	1.0
p_w	0.002	0.000	0.002	0.003	0.000	0.000	6417.0	3325.0	1.0
pzero_0	0.845	0.005	0.835	0.855	0.000	0.000	5792.0	3178.0	1.0
pzero_1	0.855	0.005	0.845	0.864	0.000	0.000	5572.0	3196.0	1.0

2. Deciding Price to Minimize Expected Loss

If the price is $3.99 (group 0), then the expected revenue per user is

rev_0 = 3.99 * expected purchases

= 3.99 * (p_w * (mu_0 + mu_w_diff) + (1 - p_w) * (1 - pzero_0) * mu_0)

Similarly, the expected revenue per user when the price is $4.99 is

rev_1 = 4.99 * (p_w * (mu_0 + mu_w_diff) + (1 - p_w) * (1 - pzero_1) * mu_1)

For each sample in the posterior, we can calculate values for both rev_0 and rev_1. In each sample, decision 0's loss will be 0 if rev_0 >= rev_1, and rev_1 – rev_0 if rev_0 < rev_1. Decision 1's loss will be 0 if rev_0 <= rev_1, and rev_0 – rev_1 if rev_0 > rev_1.

Finally, to calculate the expected loss per user for each decision, we simply compute the mean of losses over the posterior samples.

The result is:

```python
print(f"Expected Loss per person when price is $3.99: ${loss_0:.4f}")
print(f"Expected Loss per person when price is $4.99: ${loss_1:.4f}")
```

```
Expected Loss per person when price is $3.99: $0.0011
Expected Loss per person when price is $4.99: $0.2896
```

Therefore we decide to price our in-app feature at $3.99 so that expected loss is minimized.