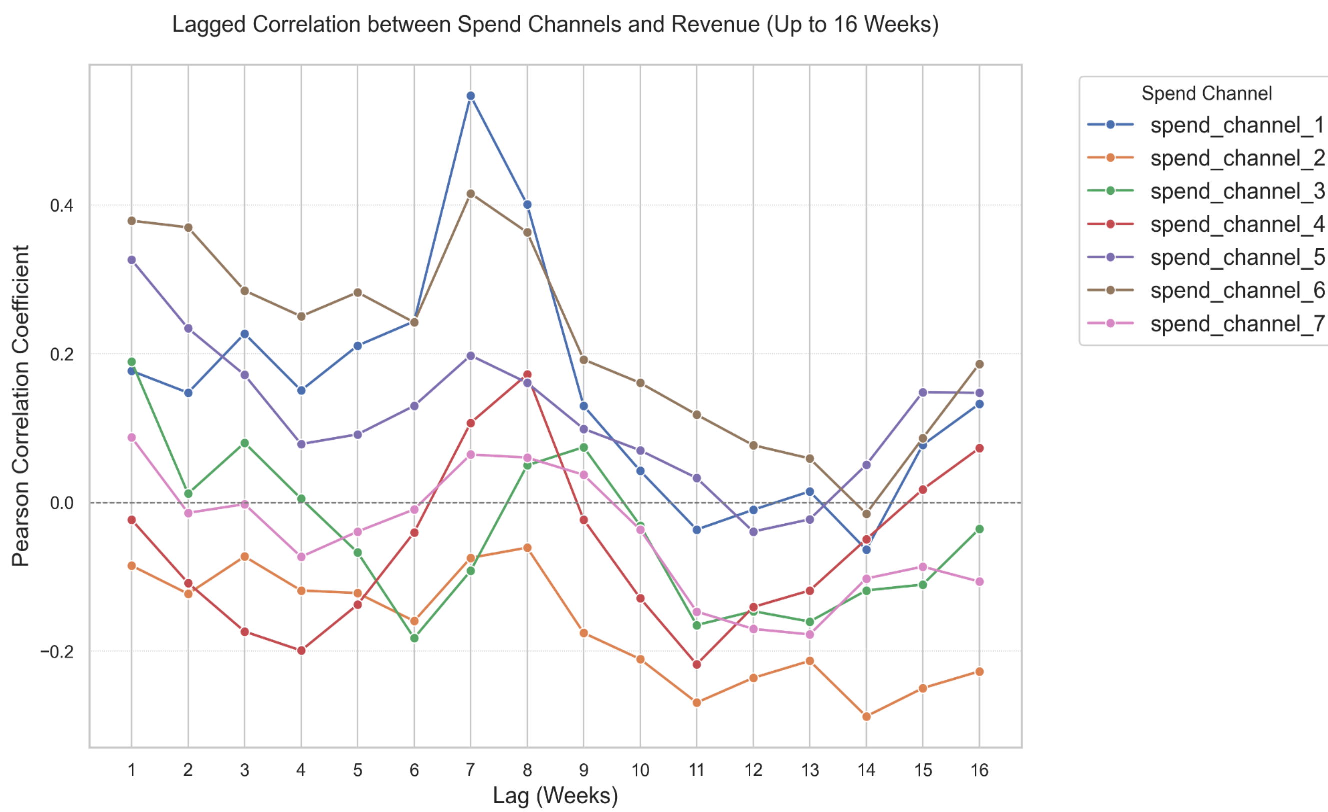
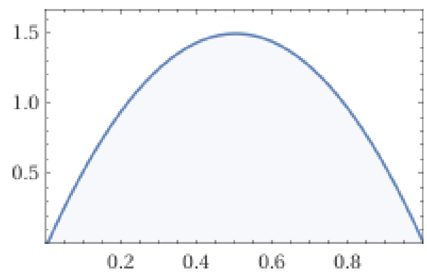
Q&A: Analyze the Marketing Channel spending

1. How do you model spend carryover?
   I modelled spend carry-over using a geometric adstock function, which accounts for the delayed and decaying impact of media spending on future revenue. Specifically, for each channel `i`, the adstocked spend at time `t` is calculated as a weighted sum of the spend in the current week (`t`) and the preceding `l_max - 1` weeks.
   From some EDA, it was evident that at least a few channels especially **channels 1,6 and 4** (also evident from the plot below) show an increased revenue correlation after 7-8 weeks of their actual expenditure and then start decreasing.

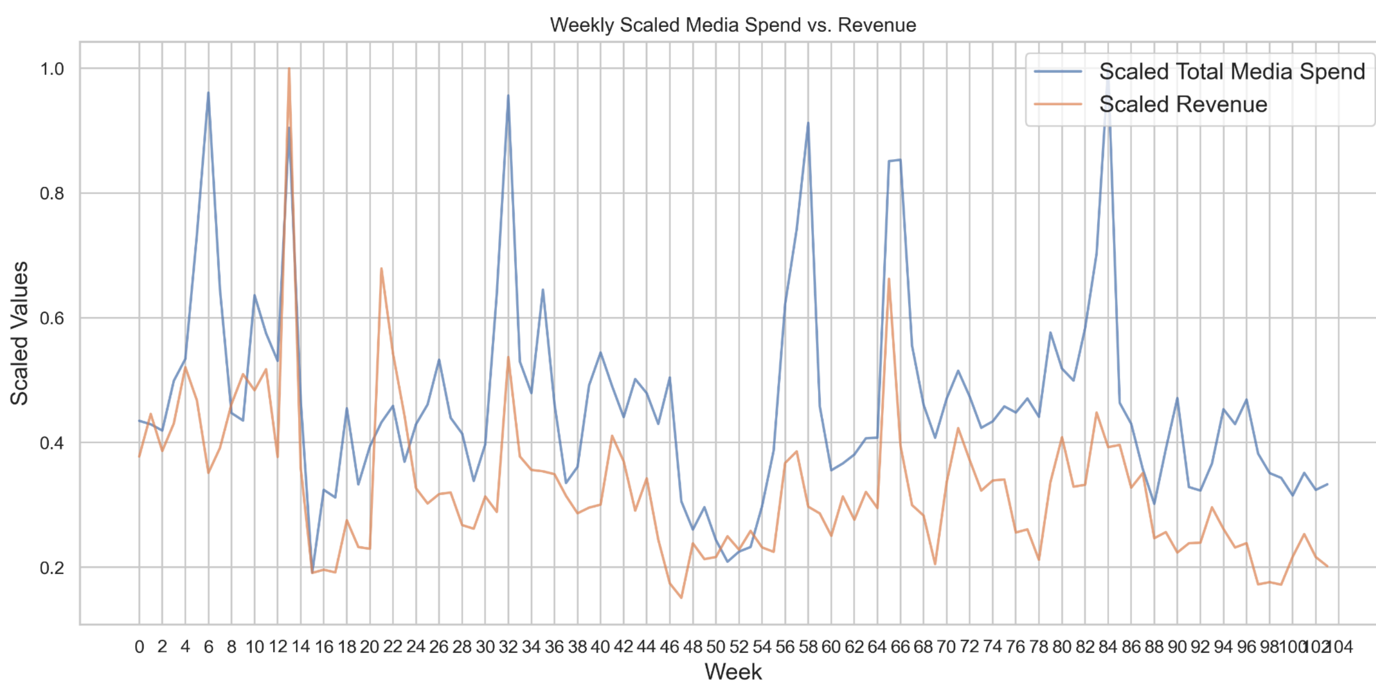


For this reason, the l_max was chosen as 14 (double of 7) since the lag effect seems to peak in this time interval. Additionally, the following plot confirms a few cases where an increase in total ad spend in one week seems to have impacted the revenues in the **7th-8th week** (if we ignore other factors at the moment). Because of this delayed impact, I chose the following distribution to sample alpha.



alpha = pm.Beta("alpha", alpha=2.0, beta=2.0, dims="channel")

The following plot shows the effect of increased channel spend in week 6 on week 13, in week 13 on week 21 and in week 58 on week 65, confirming the 7-day lag's peak impact, ignoring all the other control factors.



Note: I sampled all channel alphas using the same beta distribution in this solution (sampled values would be different but following the same distribution).

2. Explain your choice of prior inputs to the model.
   Below is a list of priors and their explanation:

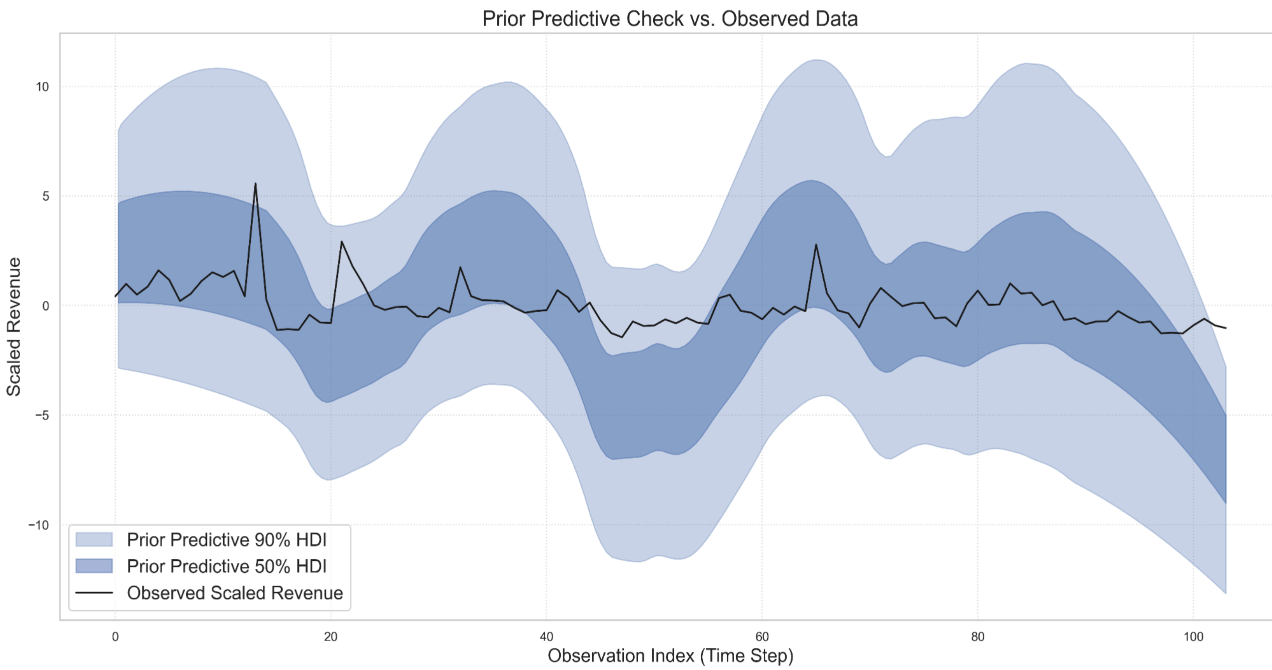
- `baseline = pm.Normal("baseline", mu=0, sigma=1.0)`: Assumes the baseline (average scaled revenue when all other effects are zero) is centred around 0 (because the revenue is scaled) with a standard deviation of 1.0. This is a weakly informative prior allowing the baseline to vary reasonably around the mean of the scaled data.

- `beta_trend = pm.Normal("beta_trend", mu=0, sigma=1.0)`: Centred around 0 (no trend a priori) with a standard deviation of 1.0. This allows for a moderate positive or negative trend to be learned.
- `beta_season = pm.Normal("beta_season", mu=0, sigma=0.5, dims="fourier")`: The `sigma=0.5` allows for less to moderate seasonal effects, suggesting seasonality might explain less variance than the overall baseline or trend but still be present.
- `alpha = pm.Beta("alpha", alpha=2.0, beta=2.0, dims="channel")`: This Beta distribution first persists the decay with higher alpha until the half of lag days and then starts decaying towards the end of the lag interval adhering to our findings above.
- `beta_channel = pm.HalfNormal("beta_channel", sigma=1, dims="channel")`: Sets the prior for each channel's effectiveness coefficient. The HalfNormal restricts effects to be non-negative (marketing doesn't hurt revenue as is standard practice in marketing).
- `sigma = pm.HalfNormal("sigma", sigma=1.0)`: It assumes the typical residual error, after accounting for all model components, will be around 1.0 standard deviations of the scaled revenue.

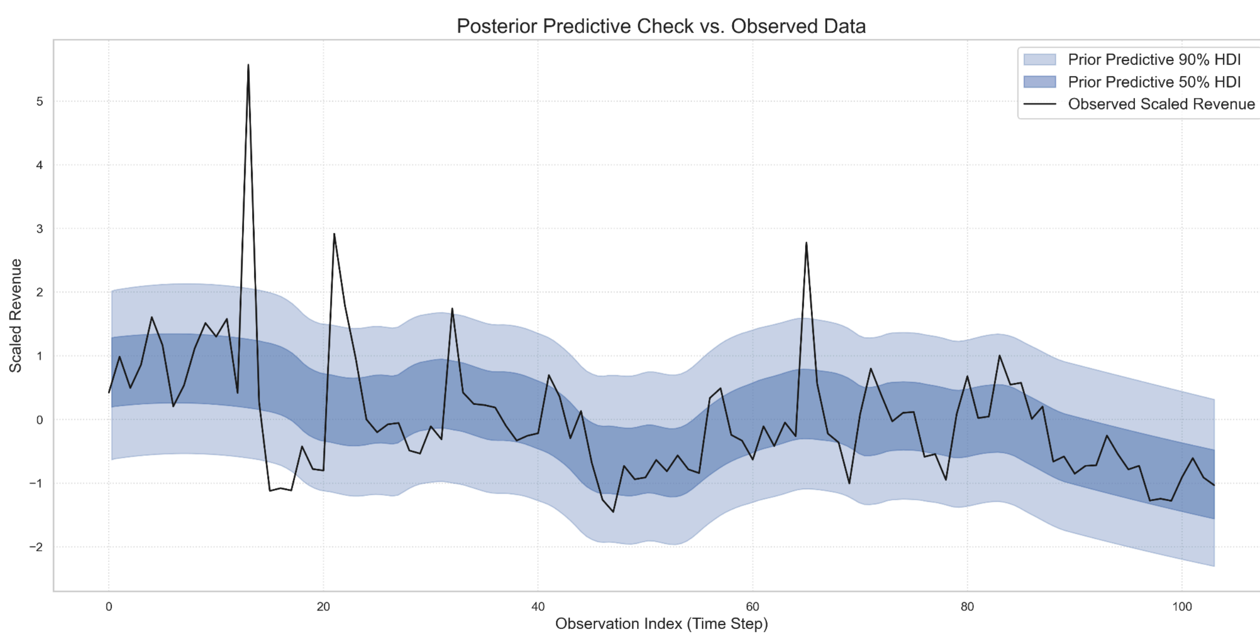
3. How are your model results based on prior sampling vs. posterior sampling?

- **Prior Predictive Sampling:** The first plot ('Prior Predictive Check vs. Observed Data') shows the model's predictions based only on the priors, before learning from the data. It visualizes the range of outcomes (revenue patterns over time) considered possible by the model's initial assumptions using the 50% and 90% HDIs. Typically, these prior HDIs are very wide, reflecting the initial uncertainty. It's essential to compare the actual observed data (`y`) to these bands to see if my prior assumptions are reasonable enough to generate something like the real data.

  The wide bands in the prior distribution generally mean a good start- it shows the model is not overly confident without data. Would be better if the revenue (y) lies inside the 50% HDI.



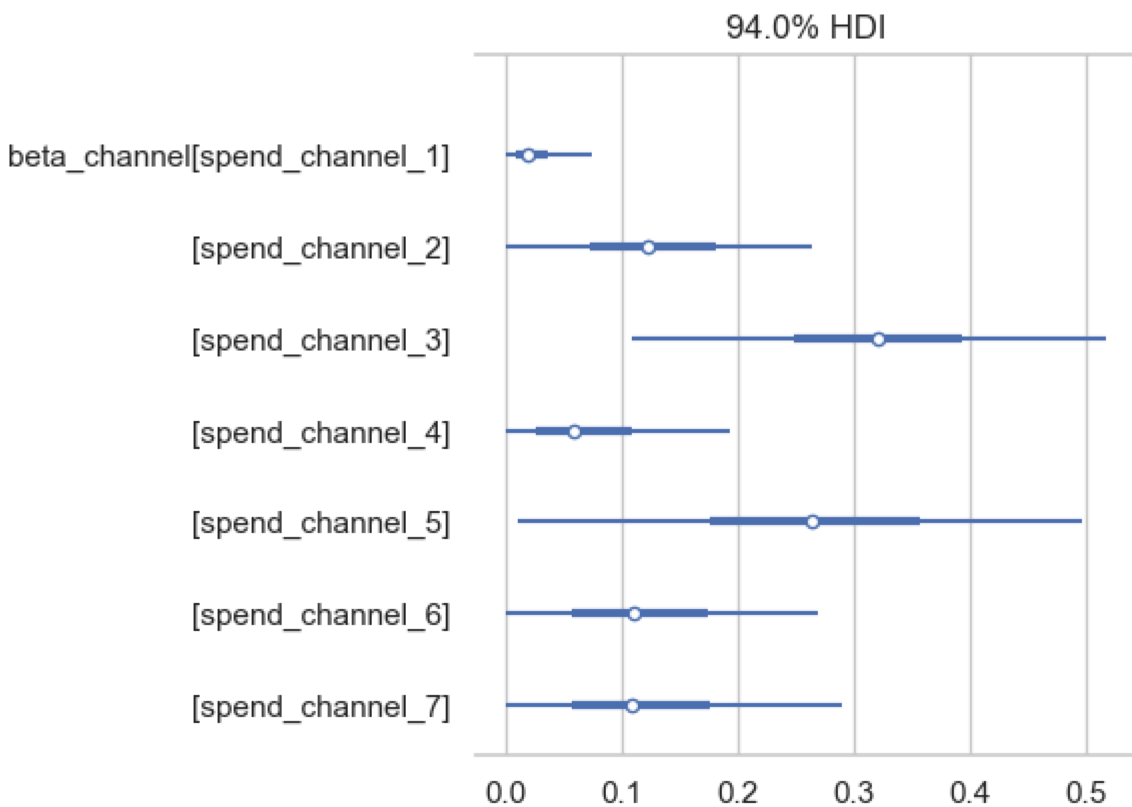
- **Posterior Predictive Sampling:** The following plot shows the model's predictions *after* it has been fitted to the observed data (y). These posterior HDIs are much narrower than the prior ones and encompass most of the actual observed data in the 90% HDI.



4. How good is your model performing? How you do measure it?

- Visually: By observing the 'Posterior Predictive Check vs. Observed Data' we can comment that the intervals generated are able to partially cover the ground truth in its 90% confidence interval (90% HDI) though there are still points that our posterior misses completely. It looks like these exceptional revenue periods fall inside the holiday season and thus including holidays could give us a better fit.
- Quantitatively: I calculated two standard regression metrics (R2 and MAE), comparing the model's average prediction (y_pred) against the true values (y_true) on the original revenue scale. We can still improve R2 by modelling other control factors.

5. What are your main insights in terms of channel performance/ effects?
   Channel 3 seems to have the highest mean value of the beta parameter followed by channel 5. Below is a plot showing different beta distributions for channels with 94% HDI.



6. (Bonus) Can you derive ROI (return on investment) estimates per channel? What is the best channel in terms of ROI?
   Yes, I implemented a rather simple calculation using the average effectiveness (channel betas) of each channel in standardized units (i.e., change in standardized revenue per 1 std increase in standardized spend). This can be used to calculate the dollars of revenue we get for each real $1 spent on the channel.
   According to this calculation, **channel 2** has the highest ROI.