Tenuous Relations and Timeseries Analysis

Nathaniel Forde
Data Science @ Personio

Personio

Agenda

1. Motivation: Background Problem
2. Structural Autoregressive Models
3. Structural Autoregressive Models in PyMC
    - Pause for Questions
4. Vector Autoregressive Models
5. Bayesian Vector Autoregressive Models in PyMC
6. Hierarchical BVARs in PyMC

Motivation: Background Problem

Disclaimers and Acknowledgements

> **Note**
>
> No Personio data used in this presentation. The problem discussed motivated the research, but the models presented below will use fake or public data.

We will motivate the work with a discussion of App performance measurements and customer feedback metrics.

How do they influence one another and how to best distil the relationship between the two?

> **Note**
>
> Acknowledgements: The work on the BVAR discussed here took inspiration and borrowed heavily from the PYMC labs blogpost here. The example PRs benefited from the feedback of the PYMC devs.

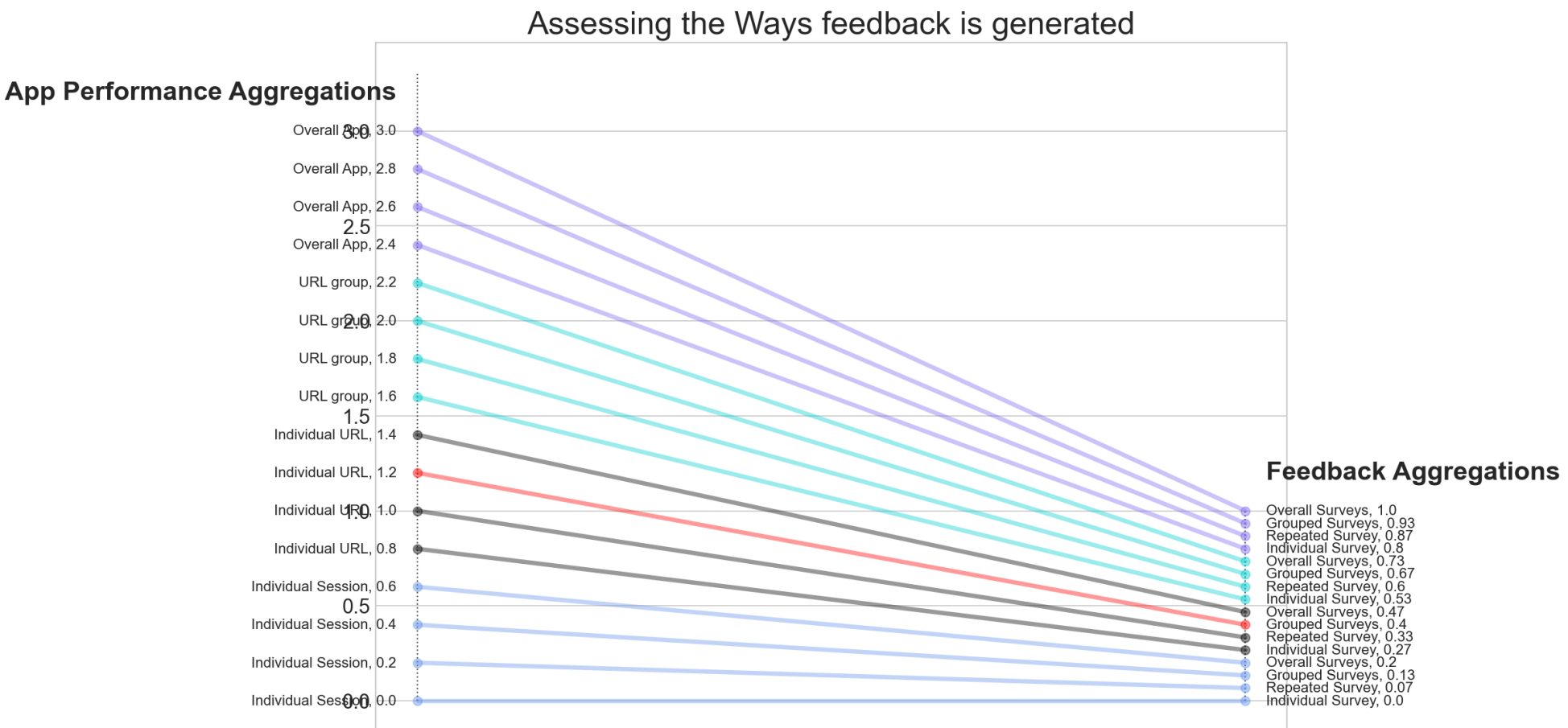
> **Warning**
>
> Not an Economist!!!

Systems and Feedback: App Performance

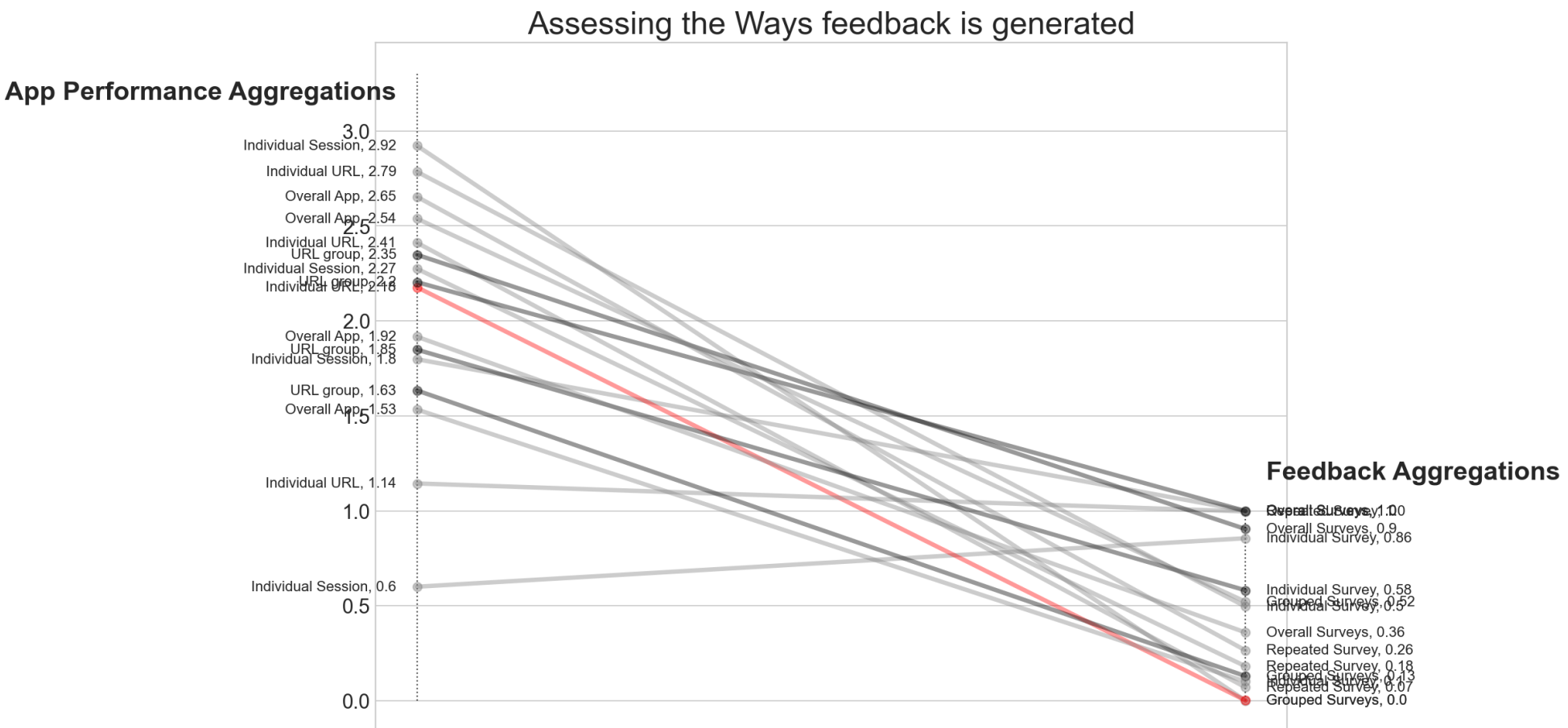
> “Balancing feedback loops are equlibrating or goal seeking structures in systems and are both sources of stability and resistence to change”

> “Allowing performance standards to be influenced by past performance… sets up a reinforcing feedback of eroding goals that sets a system drifting toward low performance.”

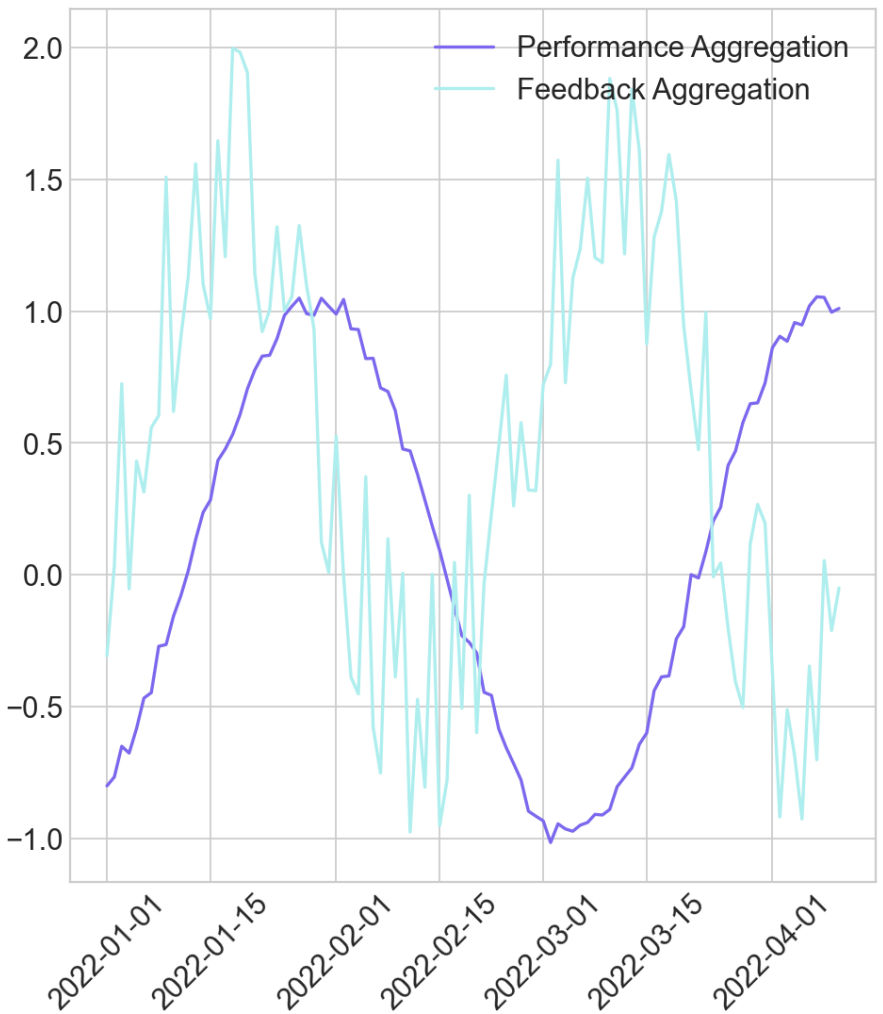
Disentangling Individual effects



Identifying the Channel of Transmission



Lagging Effects in Time



Personio

Structural Autoregressive Models

Autogressive Models

- Regression: The Linear Projection of Y onto X is the OLS solution.
- The Wold Decomposition: If a random variable Y_t is *covariance stationary* then Y_t has a linear representation:

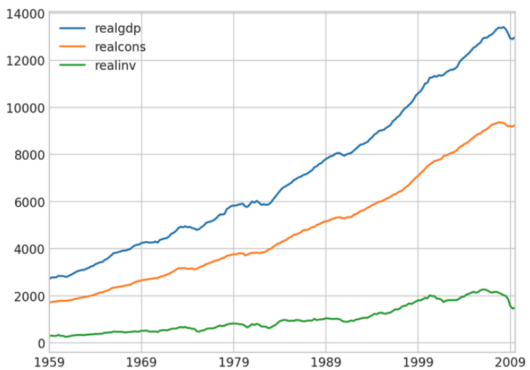
$$Y_t = \mu + \sum_{j=0}^{\infty} b_j e_{t-j}$$

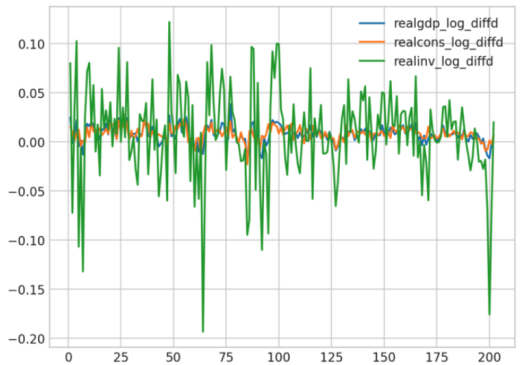
- Auto-regression: Extends this idea to Hilbert space of the infinite past history:
  - $$Y_t = \mu + \sum_{j=1}^{\infty} a_j Y_{t-j} + e_t$$
  - Where the error terms e represent white-noise $N(0, 1)$
- > "…this justifies linear models as approximations" - Hansen *Econometrics*
- Most timeseries data is shorter than their infinite past.

Moving Average Representation and Impulse Response

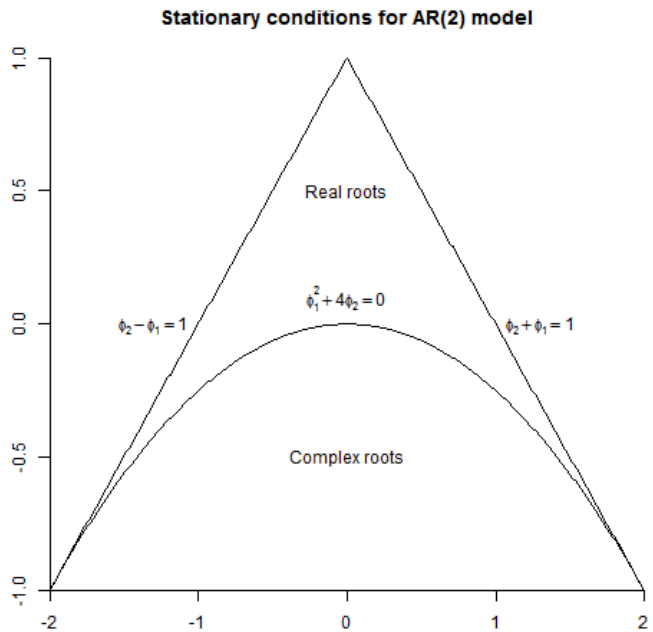
- The coefficients in: $Y_t = \mu + \sum_{j=0}^{\infty} b_j e_{t-j}$ are known as the impulse response function **IRF** representing the change in Y_{t+j} due to a shock at time t.
  - $\frac{\partial}{\partial e_t} Y_{t+j} = b_j$
- They can be recovered from the AR representation by recursive reconstruction.
- IRFs are often reported when scaled by the standard deviation of shocks.

Stationarity Requirements





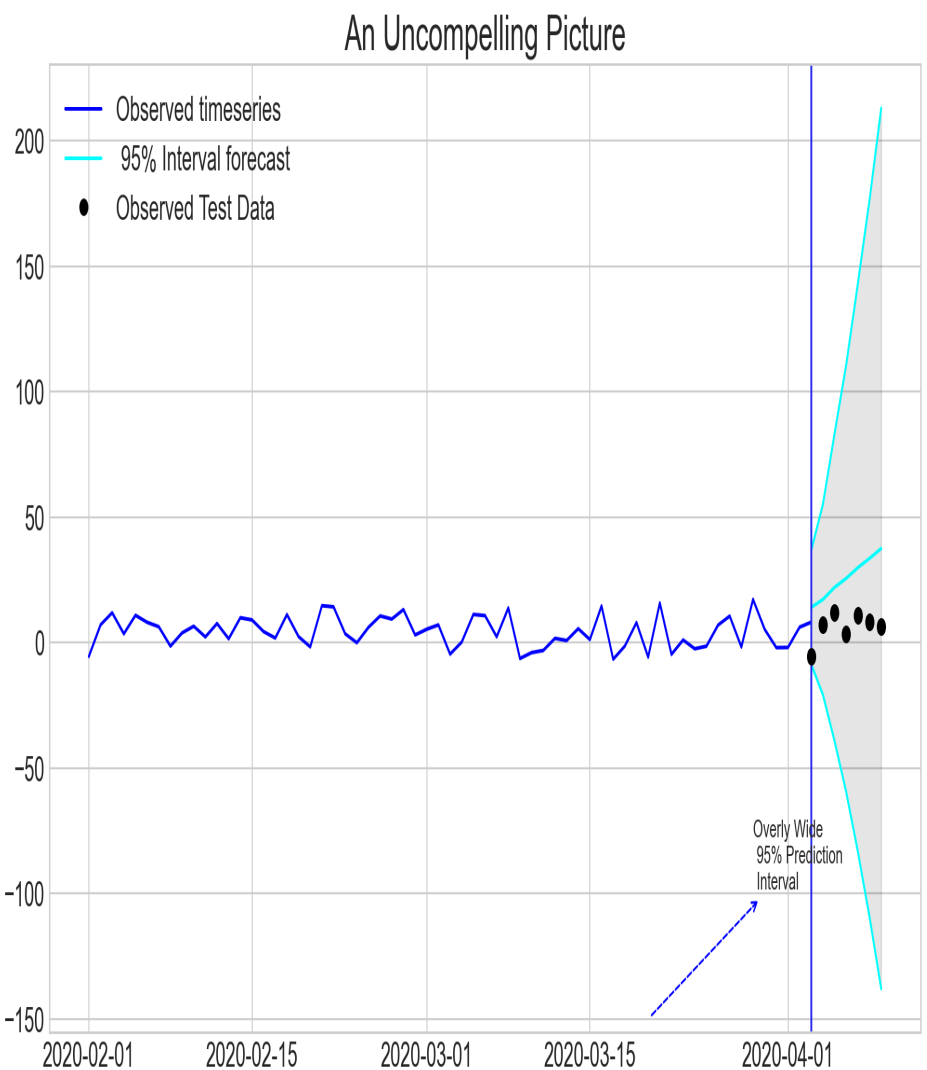
Tricks to achieve stationarity

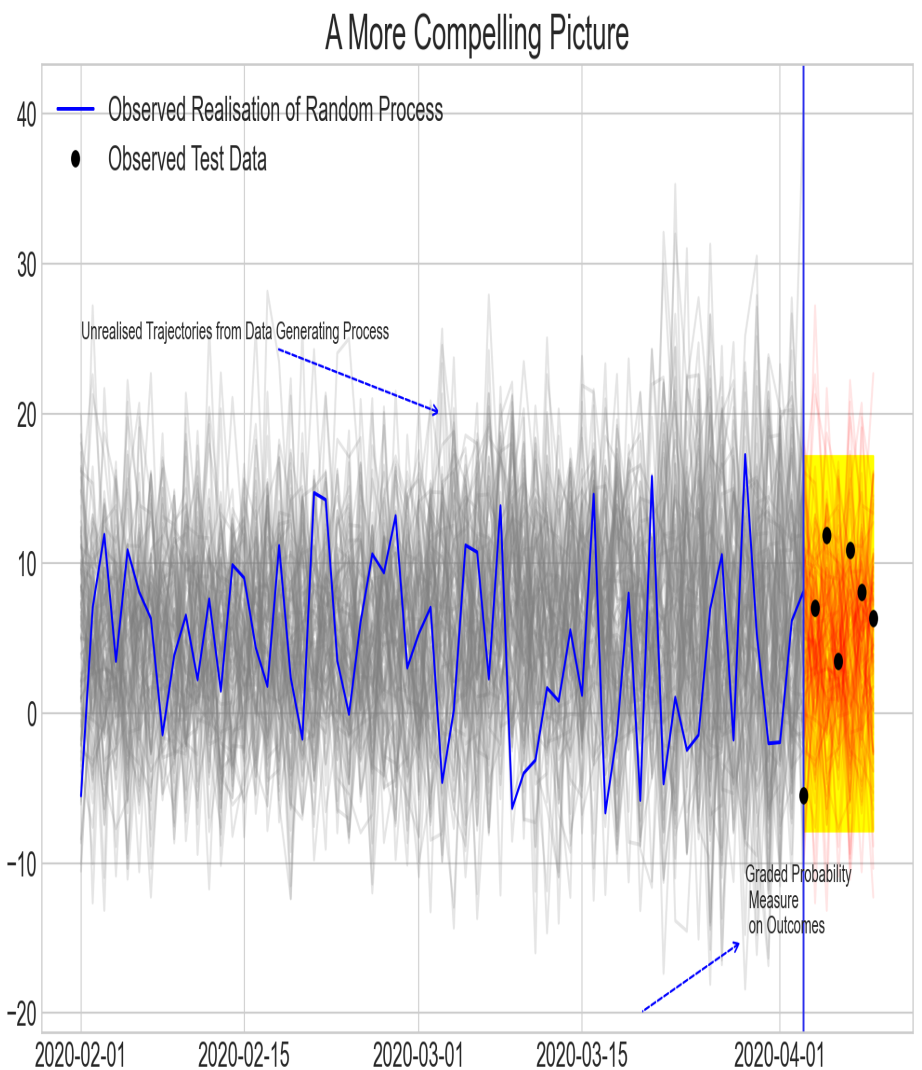


Conditions required for Stationarity

Structure as Story

> "The Child is the father of the Man" - Wordsworth





Characterising Structure

$$Y \sim TruncNorm(\mu, \sigma, 0, \infty)$$
$$\mu = AR_\mu + f(seasonality) + f(trend) + \ldots$$
$$\sigma \sim InverseGamma(3, .5)$$
$$trend = \alpha + \beta \cdot T_i$$
$$\alpha \sim Normal(7, 1)$$
$$\beta \sim Normal(0.009, .1)$$
$$seasonality = FourierFeatures' \beta_{fourier}$$
$$\beta_{fourier} \sim Normal(0, 0.5)$$
$$AR_\mu \sim \mu_{ar} + a_1 \cdot Y_{t-1} + a_2 \cdot Y_{t-2} \ldots + \gamma$$

Learning Structure

$$p(\theta|\mathbf{Data}) \sim p(\theta) \cdot p(\mathbf{Data}|\theta)$$

Bayesian Structural Autoregressive Models

- Bayesian Structural Timeseries (BSTS) are often seen as an alternative to Auto-regressive models[1]
- But BSTS models can also incorporate latent auto-regressive components.
- Benefits of going Bayesian:
  - Transparent understanding of the sources of uncertainty within your model and a capacity to introduce outside information by way of priors or other covariates.
  - Expressive “lego-like” composability of the modelling structure matches the compositionality of thought.
  - Plausible counterfactual inference with posterior predictive distributions -

$$p(\hat{Data}|\mathbf{Data}) = \int_{\Theta} p(\hat{Data}|\theta, \mathbf{Data})\, p(\theta|\mathbf{Data})\, \mathrm{d}\theta$$

1. Nice post by Kim Larsen here

Structural Autoregressive Models in PyMC

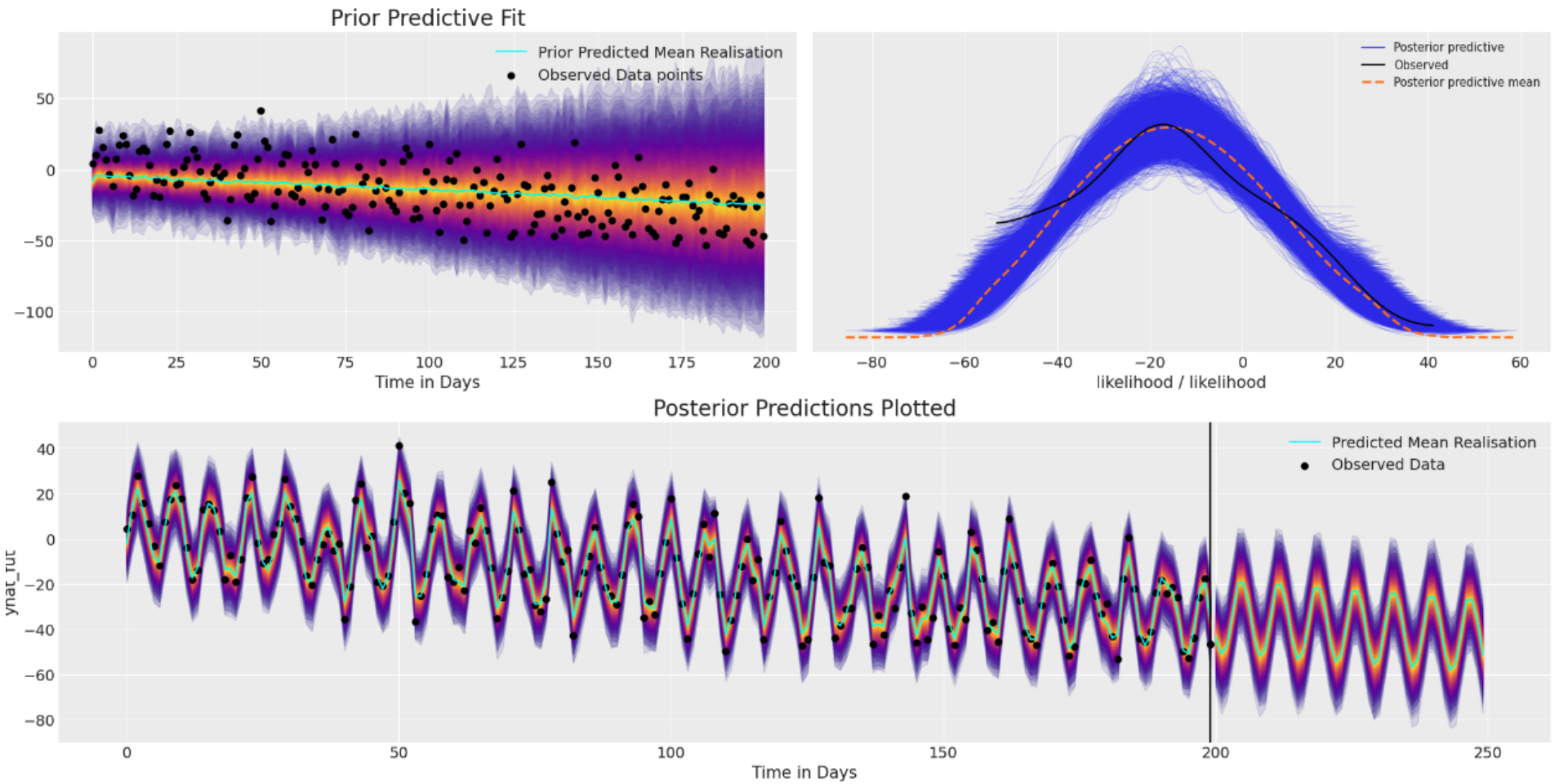
Bayesian Structural Timeseries

Priors | Additive Structural Components | Likelihood

```
with AR:
        ## Data containers to enable prediction
        t = pm.MutableData("t", t_data, dims="obs_id")
        y = pm.MutableData("y", ar_data, dims="obs_id")
        # AR priors: The first coefficient will be the intercept term
        coefs = pm.Normal("coefs", priors["coefs"]["mu"], priors["coefs"]["sigma"])
        sigma = pm.HalfNormal("sigma", priors["sigma"])
        ## Priors for the linear trend component
        alpha = pm.Normal("alpha", priors["alpha"]["mu"], priors["alpha"]["sigma"])
        beta = pm.Normal("beta", priors["beta"]["mu"], priors["beta"]["sigma"])
        ## Priors for seasonality
        beta_fourier = pm.Normal(
            "beta_fourier",
            mu=priors["beta_fourier"]["mu"],
            sigma=priors["beta_fourier"]["sigma"],
            dims="fourier_features",
        )

        #  AR process: We need one init variable for each lag, hence size is variable too
        init = pm.Normal.dist(
            priors["init"]["mu"], priors["init"]["sigma"], size=priors["init"]["size"]
        )
        # Steps of the AR model minus the lags required given specification
        ar1 = pm.AR(
            "ar",
            coefs,
            sigma=sigma,
            init_dist=init,
```

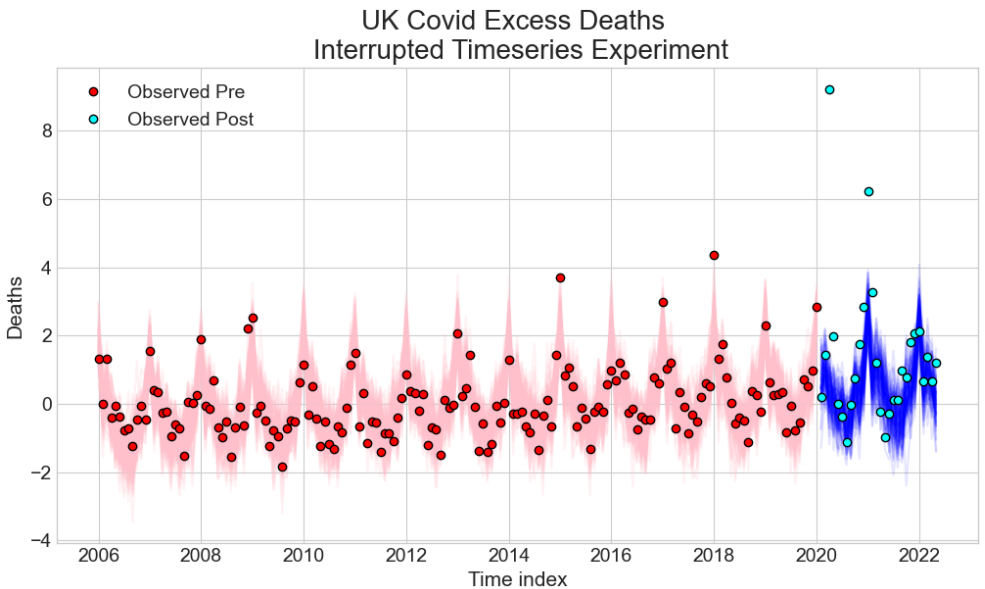
Personio

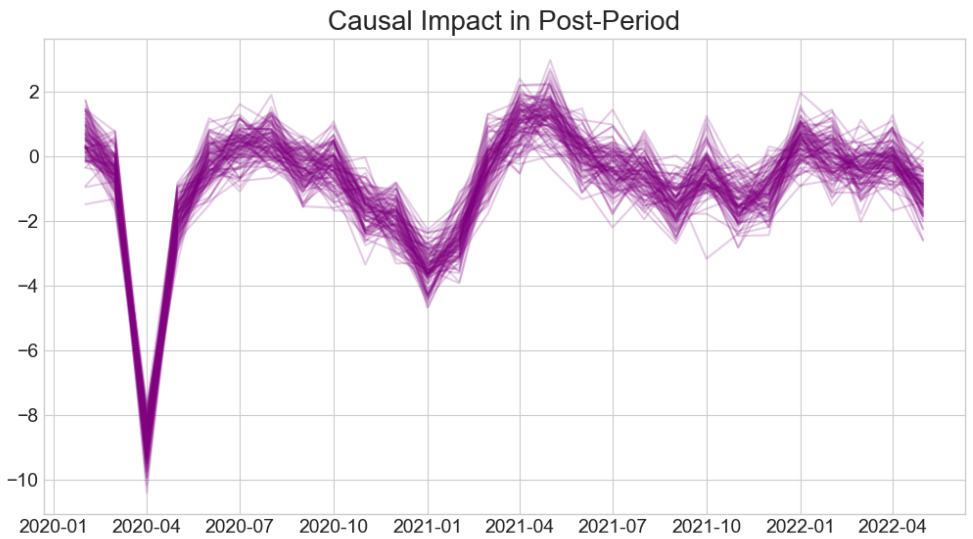
Prior Predictive and Posterior Predictive Distributions



Autoregressive BSTS prior and posterior fits. For details see PYMC docs here

Causal Inference with Interrupted Timeseries Designs





Causal Impact analysis for more detail see CausalPy

Personio

Interim: Q & A

Vector Autoregressive Models

Multivariate Timeseries

- Multivariate Linear Projection Solution applies giving
  - $$\mathbf{y}_T = \nu + A_1\mathbf{y}_{T-1} + A_2\mathbf{y}_{T-2}\ldots A_p\mathbf{y}_{T-p} + \mathbf{e}_t$$
  - Wold Decomposition:

$$\mathbf{y}_T = \mu + \Omega_1\mathbf{e}_{T-1} + \Omega_2\mathbf{e}_{T-2}\ldots\Omega_p\mathbf{e}_{T-p}$$

- Multivariate Auto-regressive representation with MV white noise error terms

$$\begin{bmatrix} gdp \\ inv \\ con \end{bmatrix}_T = \nu + A_1\begin{bmatrix} gdp \\ inv \\ con \end{bmatrix}_{T-1} + A_2\begin{bmatrix} gdp \\ inv \\ con \end{bmatrix}_{T-2} \ldots A_p\begin{bmatrix} gdp \\ inv \\ con \end{bmatrix}_{T-p} + \mathbf{e}_t$$

VAR(2)

A VAR can have n lags and m equations such that the lagged terms of each series appear in all equations.

$$gdp_t = \beta_{gdp1} \cdot gdp_{t-1} + \beta_{gdp2} \cdot gdp_{t-2} + \beta_{cons1} \cdot cons_{t-1} + \beta_{cons2} \cdot cons_{t-2} + \epsilon_{gdp}$$

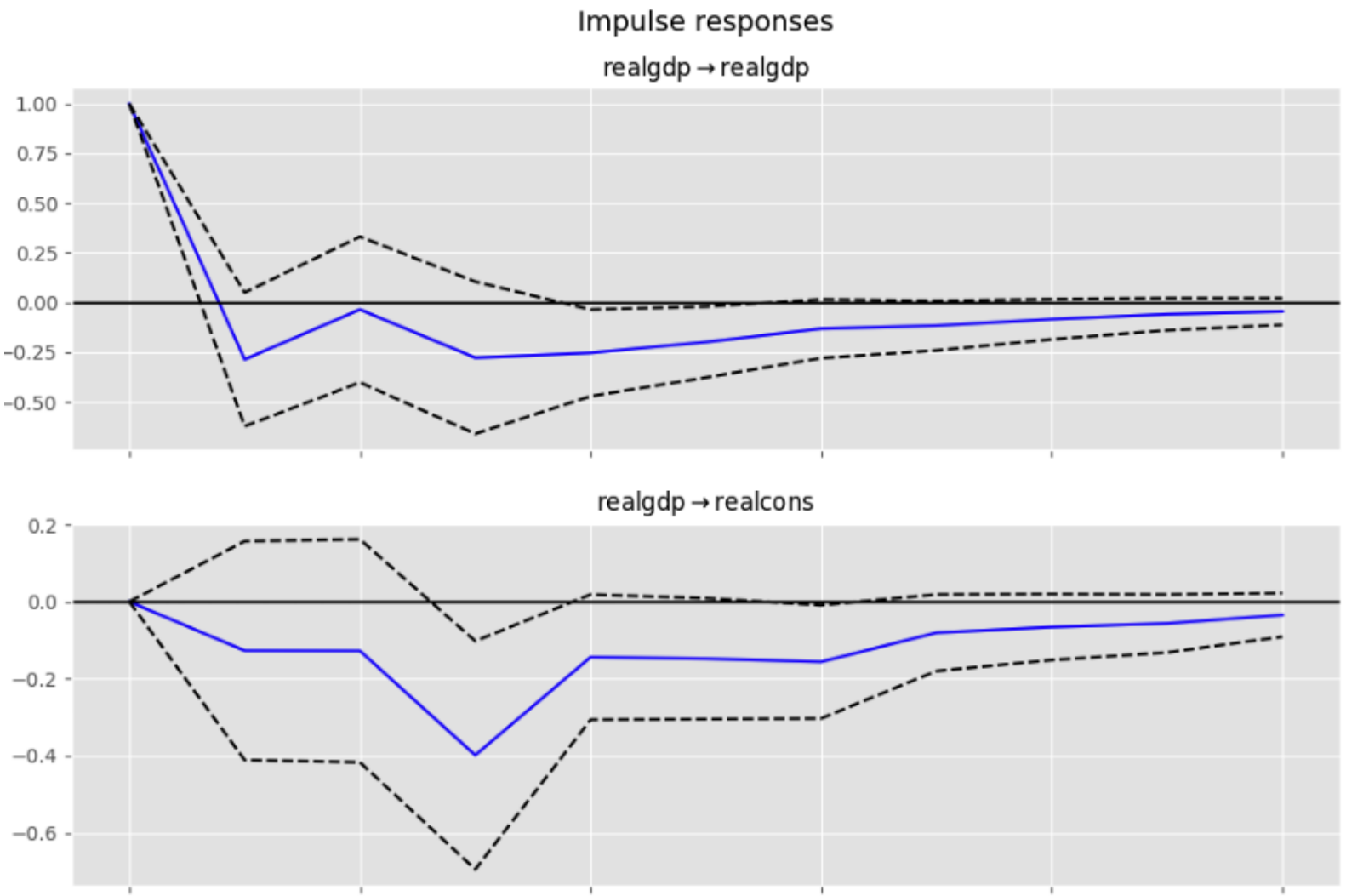
$$cons_t = \beta_{cons1} \cdot cons_{t-1} + \beta_{cons2} \cdot cons_{t-2} + \beta_{gdp1} \cdot gdp_{t-1} + \beta_{gdp2} \cdot gdp_{t-2} + \epsilon_{cons}$$

As such the **coefficients** on the cross-equation variables are of particular interest when investigating the influence of one series on another.

The MV **IRF** function can be created in an analogous way to the univariate timeseries function giving an interpretation of e.g the change in gdp_t due to a shock in $cons_{t-2}$.

This opens up the **possibility of testing** how influence between variables percolates over time.

Impulse Response



Statsmodels implementation of IRF

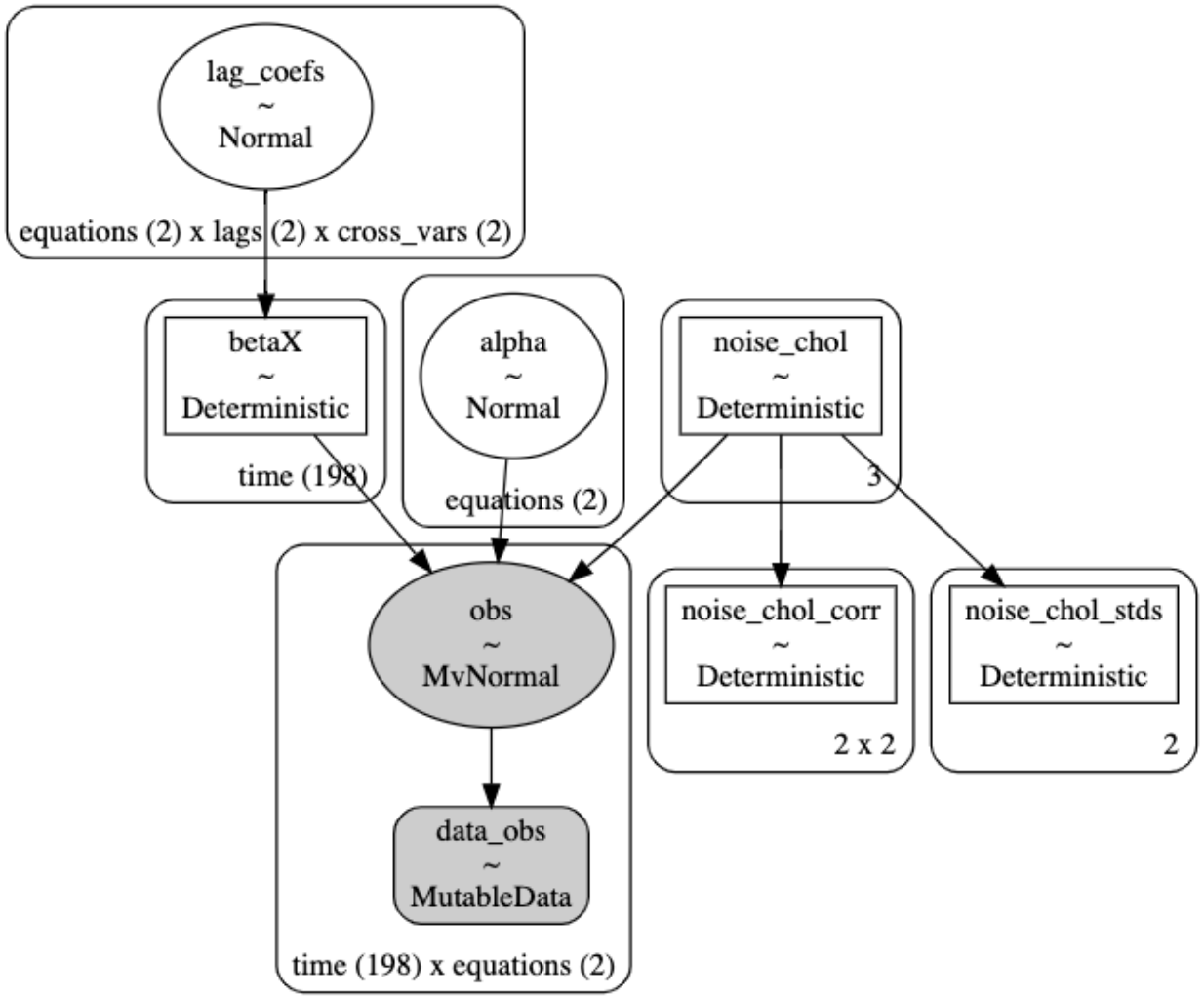
Vector Autoregressive Models in PyMC

Arranging the Coefficient Matrices

- There are allot of indices and variables in a VAR model.
- They can be concisely expressed in matrix notation, but it's not intuitive to keep track of all the indices.
- Good Bayesian workflow abstracts the component parts of the model definition.

```
### Define a helper function that will construct our autoregressive step for the marginal contribution of each lagged
### term in each of the respective time series equations
def calc_ar_step(lag_coefs, n_eqs, n_lags, df):
    ars = []
    for j in range(n_eqs):
        ar = pm.math.sum(
            [
                pm.math.sum(lag_coefs[j, i] * df.values[n_lags - (i + 1) : -(i + 1)], axis=-1)
                for i in range(n_lags)
            ],
            axis=0,
        )
    ars.append(ar)
    betaX = pm.math.stack(ars, axis=-1)

    return betaX
```

The Broad Structure



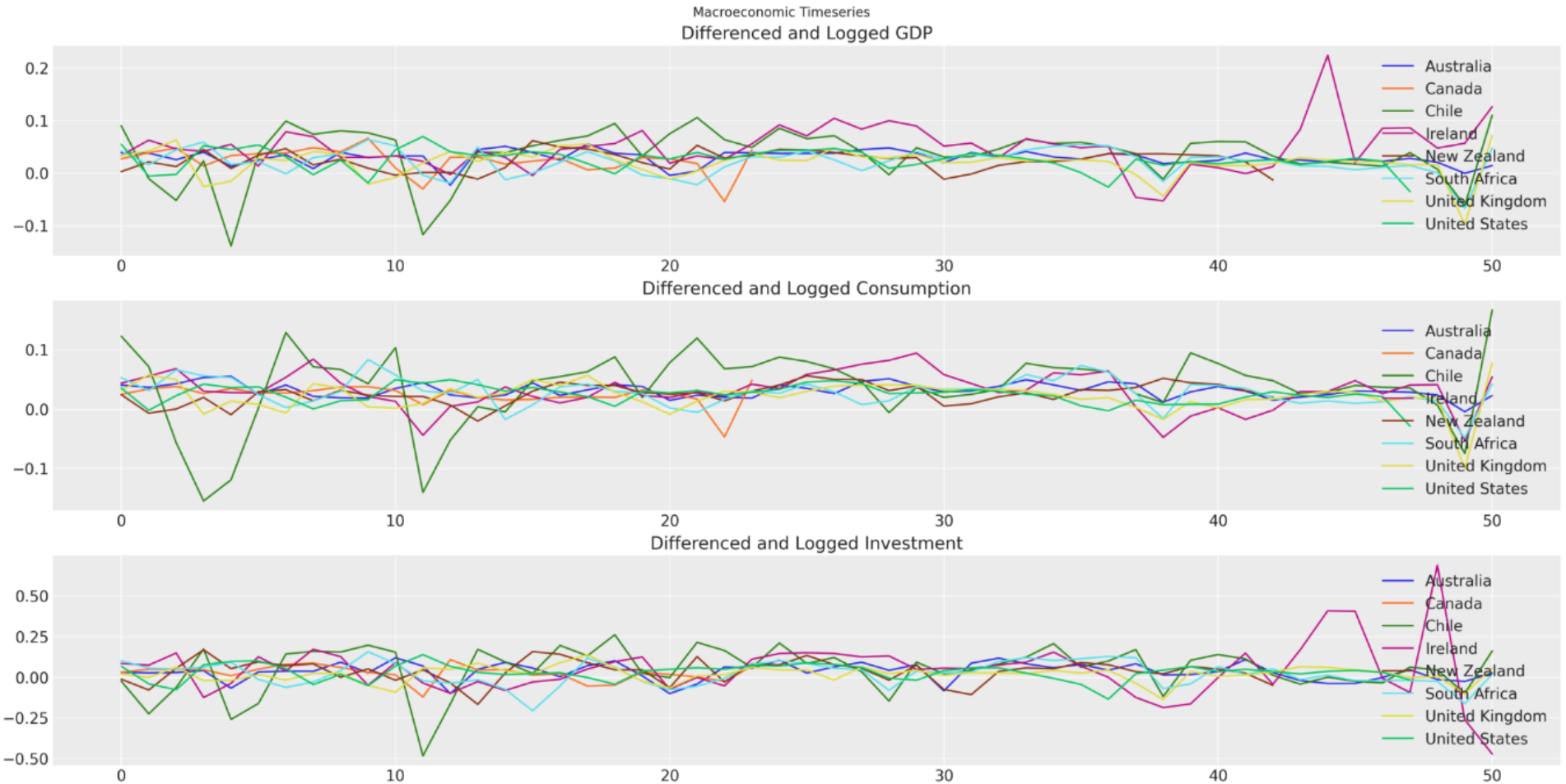
VAR(2)

Personio

Implementation in Code

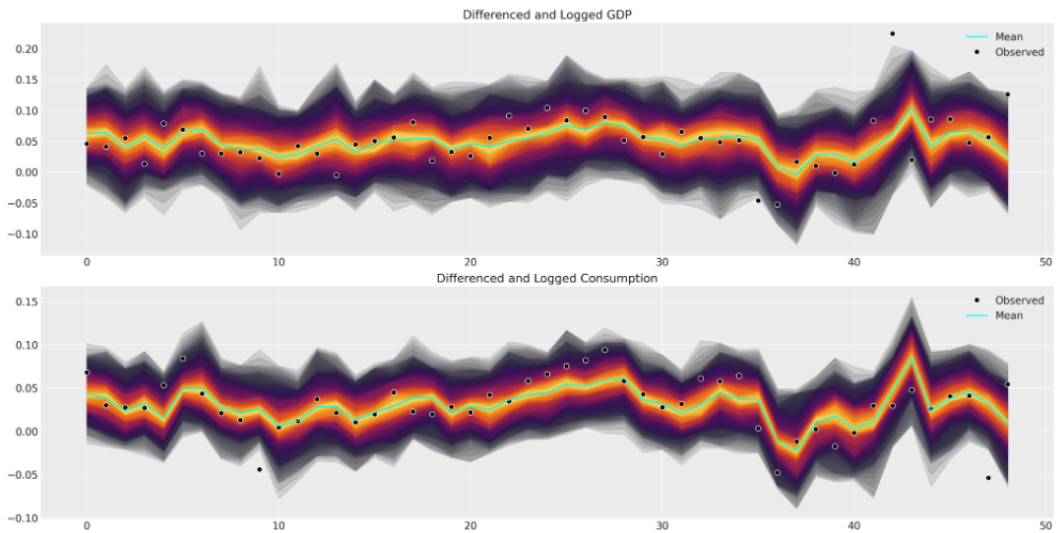

```
with pm.Model(coords=coords) as model:
        lag_coefs = pm.Normal(
            "lag_coefs",
            mu=priors["lag_coefs"]["mu"],
            sigma=priors["lag_coefs"]["sigma"],
            dims=["equations", "lags", "cross_vars"],
        )
        alpha = pm.Normal(
            "alpha", mu=priors["alpha"]["mu"], sigma=priors["alpha"]["sigma"], dims=("equations",)
        )
        data_obs = pm.Data("data_obs", df.values[n_lags:], dims=["time", "equations"], mutable=True)

        betaX = calc_ar_step(lag_coefs, n_eqs, n_lags, df)
        betaX = pm.Deterministic("betaX", betaX, dims=["time",])
        mean = alpha + betaX

        if mv_norm:
            n = df.shape[1]
            ## Under the hood the LKJ prior will retain the correlation matrix too.
            noise_chol, _, _ = pm.LKJCholeskyCov(
                "noise_chol",
                eta=priors["noise_chol"]["eta"],
                n=n,
                sd_dist=pm.HalfNormal.dist(sigma=priors["noise_chol"]["sigma"]),
            )
            obs = pm.MvNormal(
                "obs", mu=mean, chol=noise_chol, observed=data_obs, dims=["time", "equations"]
            )
```

Applying the Theory



Macroeconomic Data

The Special Case of Ireland



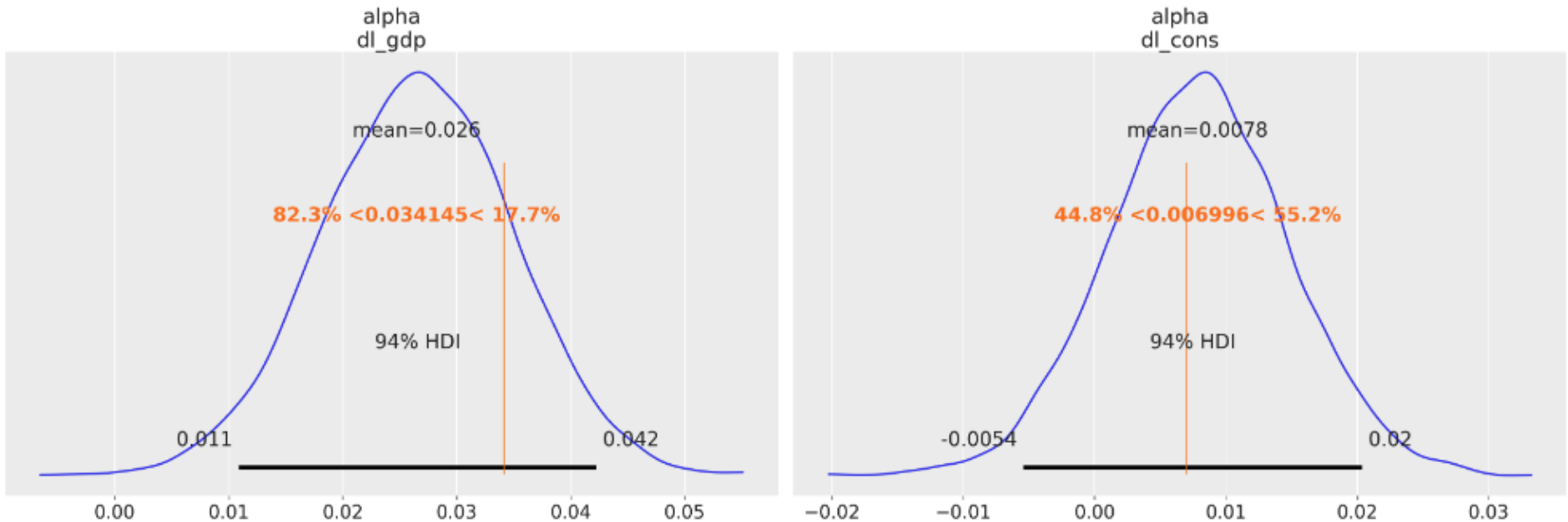
Ireland's Posterior Predictive Checks

Table 1: Posterior Mean Correlation

	GDP	CONS
GDP	1	0.43
CONS	0.43	1

Comparison with Statsmodels MLE fit

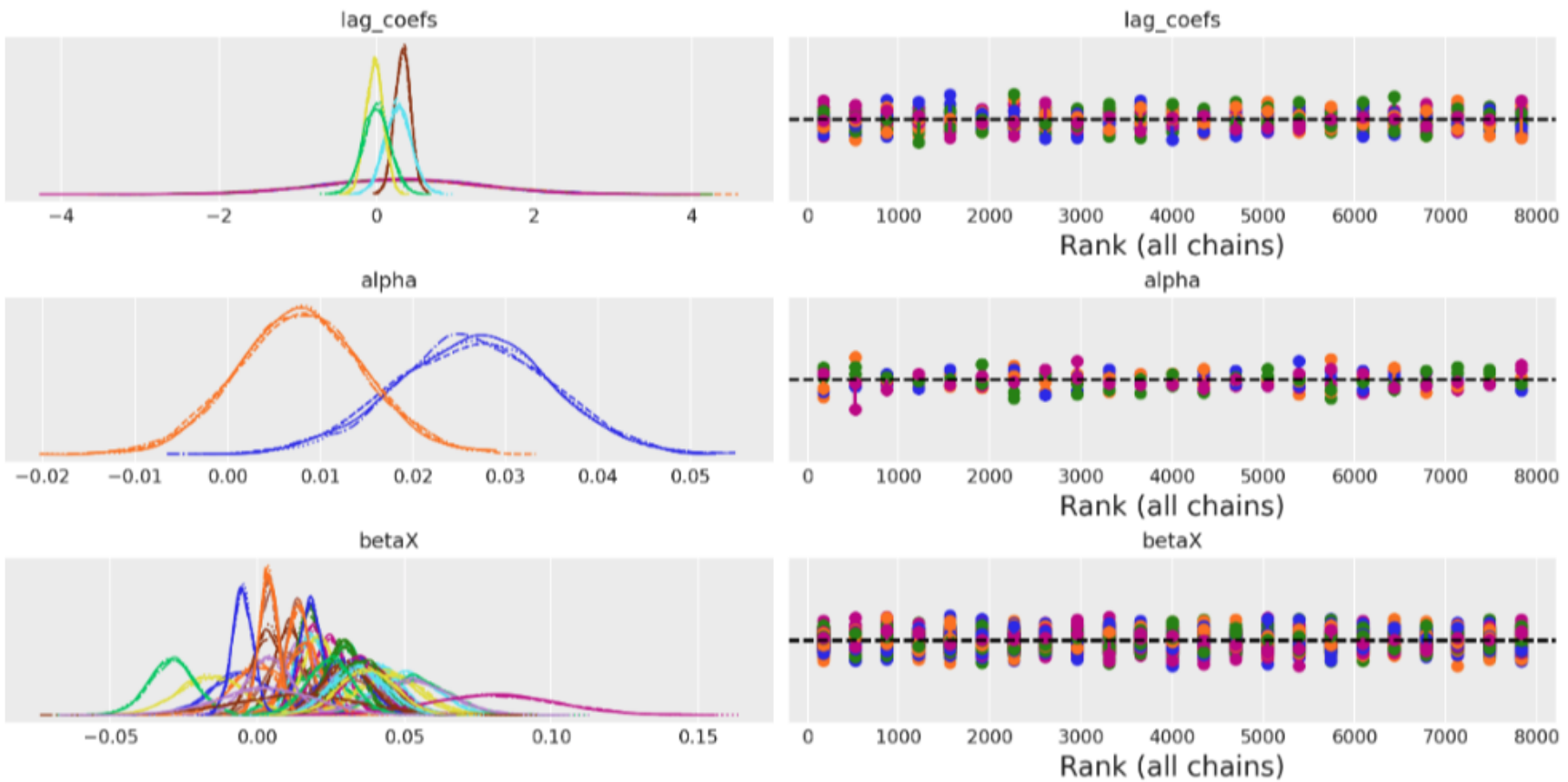
```
az.plot_posterior(idata_ireland, var_names=["alpha"], ref_val=[0.034145, 0.006996])
```



Compare

Convergence Checks


```
az.plot_trace(idata_ireland, var_names=["lag_coefs", "alpha", "betaX"], kind="rank_
```



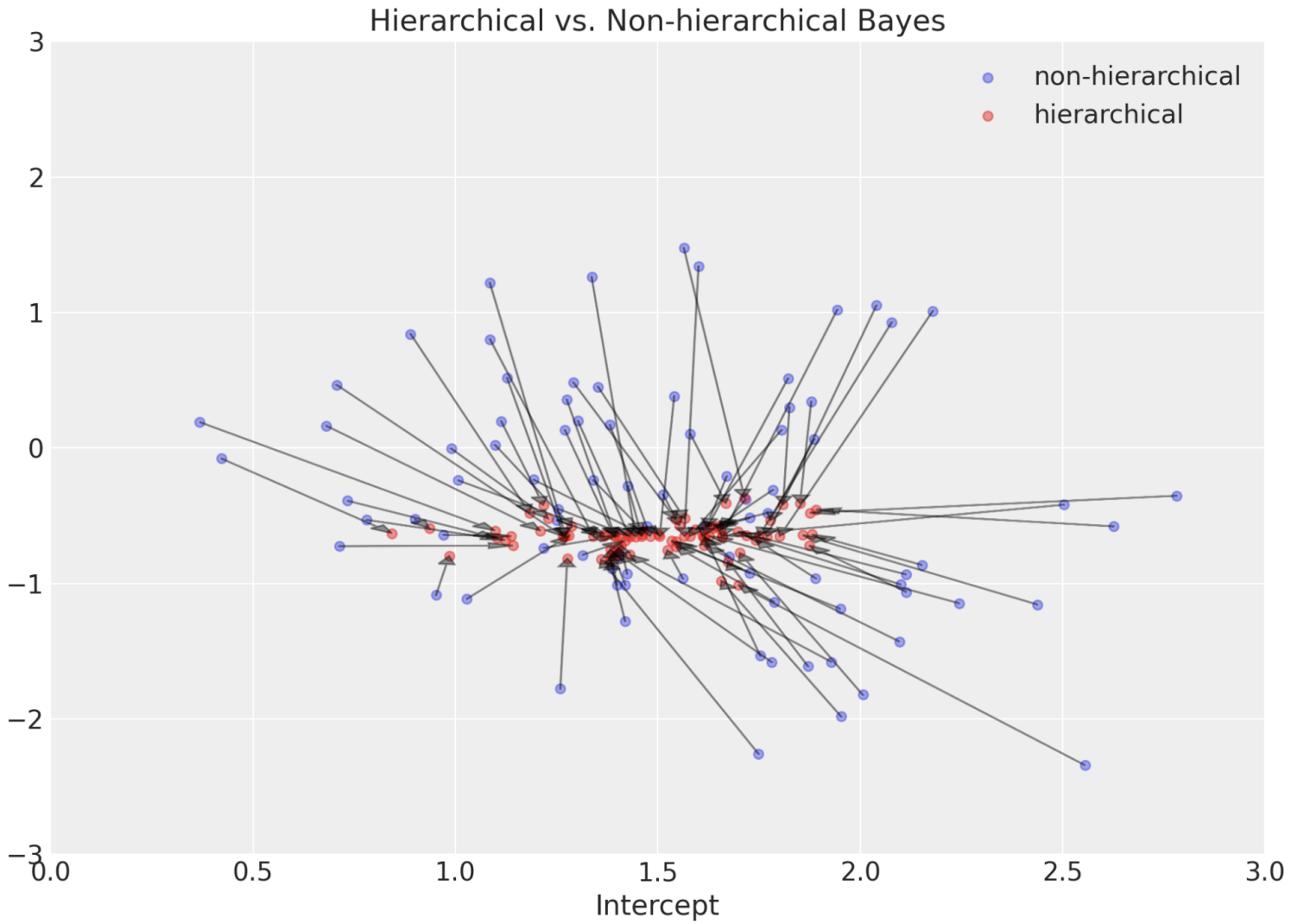
Sampling Chains

Personio

Hierarchical Bayesian VARs

> “Those who only know one country, know no country” – Seymour Martin Lipset

Pooling, Shrinkage and Hierarchy

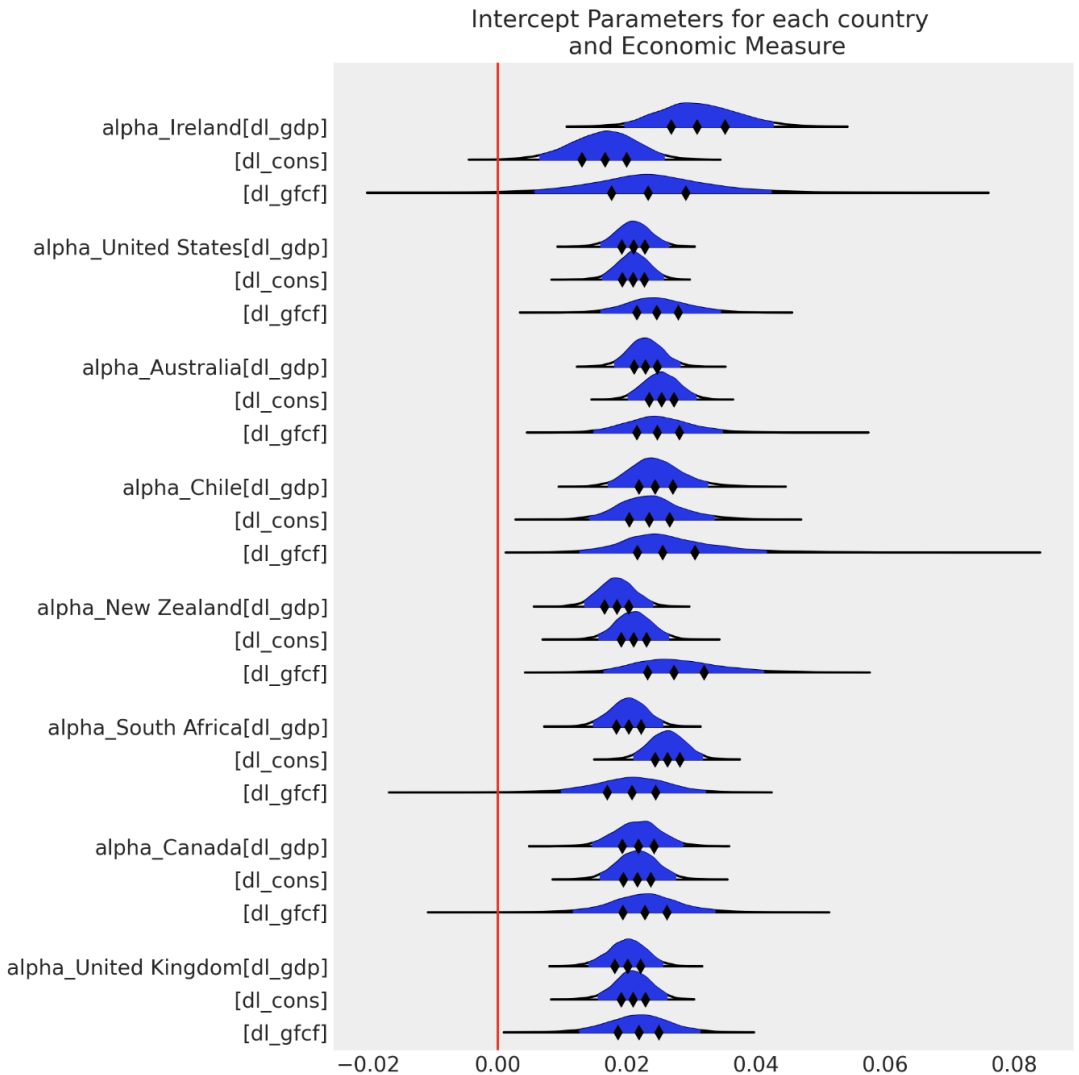


Shrinkage to Centre Effect

How to model Hierarchical Structure

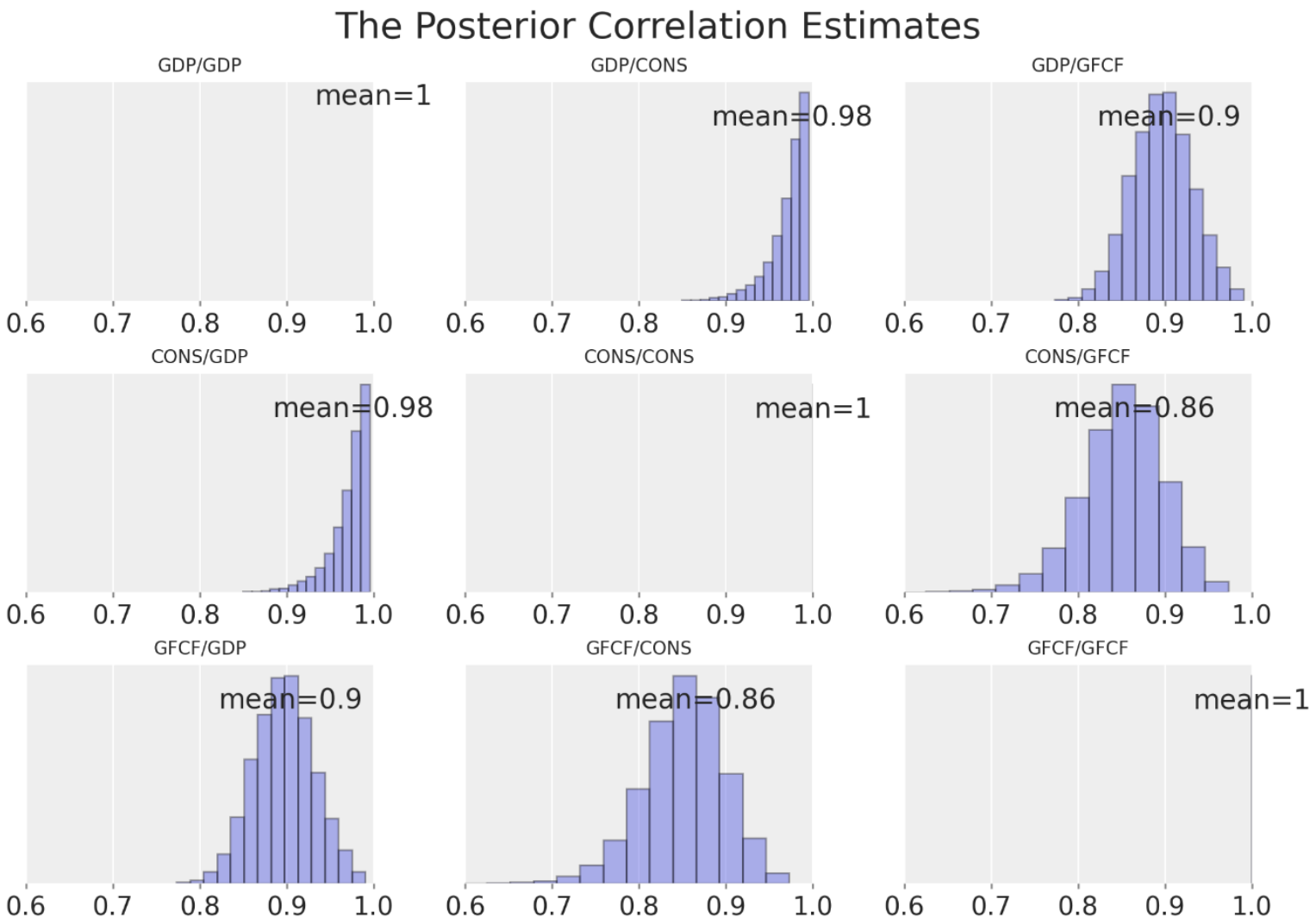

```
groups = df[group_field].unique()

    with pm.Model(coords=coords) as model:
        ## Hierarchical Priors
        rho = pm.Beta("rho", alpha=2, beta=2)
        alpha_hat_location = pm.Normal("alpha_hat_location", 0, 0.1)
        alpha_hat_scale = pm.InverseGamma("alpha_hat_scale", 3, 0.5)
        beta_hat_location = pm.Normal("beta_hat_location", 0, 0.1)
        beta_hat_scale = pm.InverseGamma("beta_hat_scale", 3, 0.5)
        omega_global, _, _ = pm.LKJCholeskyCov(
            "omega_global", n=n_eqs, eta=1.0, sd_dist=pm.Exponential.dist(1)
        )
        ## Group Specific Parameter
        for grp in groups:
            df_grp = df[df[group_field] == grp][cols]
            z_scale_beta = pm.InverseGamma(f"z_scale_beta_{grp}", 3, 0.5)
            z_scale_alpha = pm.InverseGamma(f"z_scale_alpha_{grp}", 3, 0.5)
            lag_coefs = pm.Normal(
                f"lag_coefs_{grp}",
                mu=beta_hat_location,
                # Hierarchical offset
                sigma=beta_hat_scale * z_scale_beta,
                dims=["equations", "lags", "cross_vars"],
            )
            alpha = pm.Normal(
                f"alpha_{grp}",
                mu=alpha_hat_location,
                # Hierarchical Offset
                sigma=alpha_hat_scale * z_scale_alpha,
```

Ireland is an Outlier



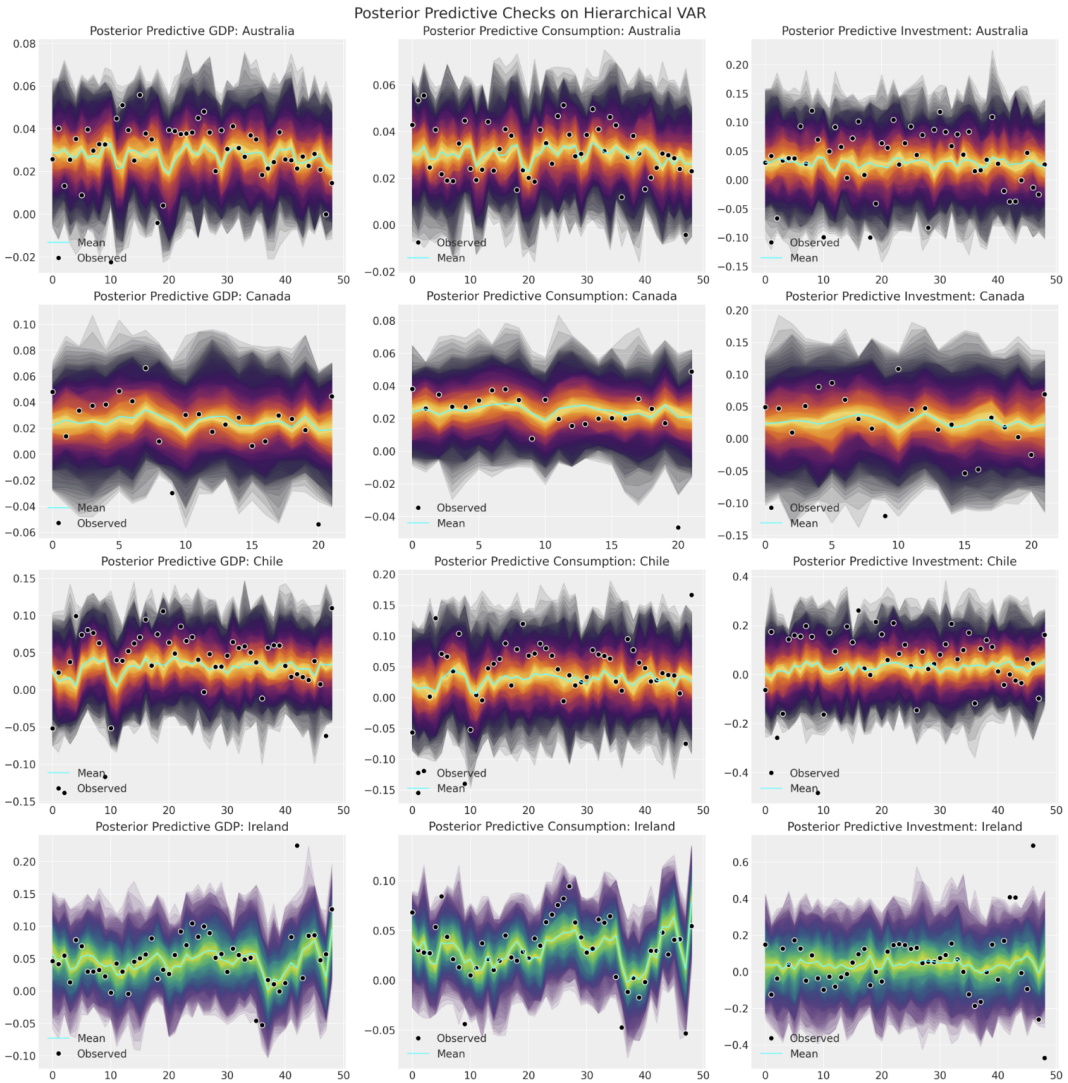
Country Specific Estimates

The Global Correlation Structure



Global Correlations

The Posterior Predictive Fits



Posterior Predictive Fits by country. See PYMC docs here

Wrap Up

- Bayesian Timeseries models are flexible and transparent tools for forecasting and causal inference.
- VAR models can help interrogate questions of the relationships between timeseries data.
- Bayesian VAR models allow the specification of regularising priors and encode structural assumptions about direction of influence.
- Hierarchical Bayesian VAR models offer the chance to borrow information across groups and interrogate the questions about the relationships between timeseries within and across the groups.
- In the context with sparse or short timeseries the hierarchical model can augment our understanding of the lagged relationships of influence.

For more Bayesian Content

PYMCON