Comparing Tips by Day of the Week with a Hierarchical Gamma Model (Bayesian)

Introduction:

As a new waiter, I have the chance to work one day a week and want to choose the most lucrative day based on tips. To make an informed decision, I'm analyzing the restaurant's tips data using a Bayesian hierarchical gamma model, which captures tipping variability across days while accounting for uncertainty. To establish my priors, I also surveyed experienced waiters about typical weekly tips. This approach will help me identify the best day to work.

Libraries and Data

In [19]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

import arviz as az
import preliz as pz
import pymc as pm
```

In [20]:
```
tips = pd.read_csv(r'C:\Users\baile\Downloads\tips.csv')
```

In [21]:
```
tips.head()
```

Out[21]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Cleaning

In [22]:
```
print(f'shape:{tips.shape}\n')
print(f'missing:{tips.isna().sum().sum()}\n')
print(f'duplicates:{tips.duplicated().sum()}\n')
print(f'dtypes:\n{tips.dtypes}')
```

```
shape:(244, 7)

missing:0

duplicates:1

dtypes:
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

In [23]:
```
tips.drop_duplicates(inplace=True)
```

In [24]:
```
print(f'duplicates:{tips.duplicated().sum()}')
```

```
duplicates:0
```

- The one duplicate was dropped.

Hierarchical Model

In [25]:
```
tip = tips['tip'].values
categories = np.array(['Sun', 'Sat', 'Thur', 'Fri'])
idx = pd.Categorical(tips['day'], categories=categories).codes
coords = {'days':categories, 'days_flat':categories[idx]}
```

- Here the day column was converted into integer codes based on the defined categories. This is used to match tips to their corresponding day in the model.

- Coordinates were defined for the hierarchical model, with days for the main categories and days_flat for the flattened observations.

In [26]:
```
with pm.Model(coords=coords) as comparing_days:
    #hyper priors
    μ_sd = pm.HalfNormal('μ_sd', 5)

    #priors
    μ = pm.HalfNormal('μ', sigma=μ_sd, dims='days')
    σ = pm.HalfNormal('σ', sigma=1, dims='days')

    #likelihood
    γ = pm.Gamma('γ', mu=μ[idx], sigma=σ[idx], observed=tip,
                dims='days_flat')
```

```
    idata_cd = pm.sample(chains=4, random_seed=1, idata_kwargs={'log_likelihood':True})
    idata_cd.extend(pm.sample_posterior_predictive(idata_cd))
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 2 jobs)
NUTS: [μ_sd, μ, σ]
Output()
```

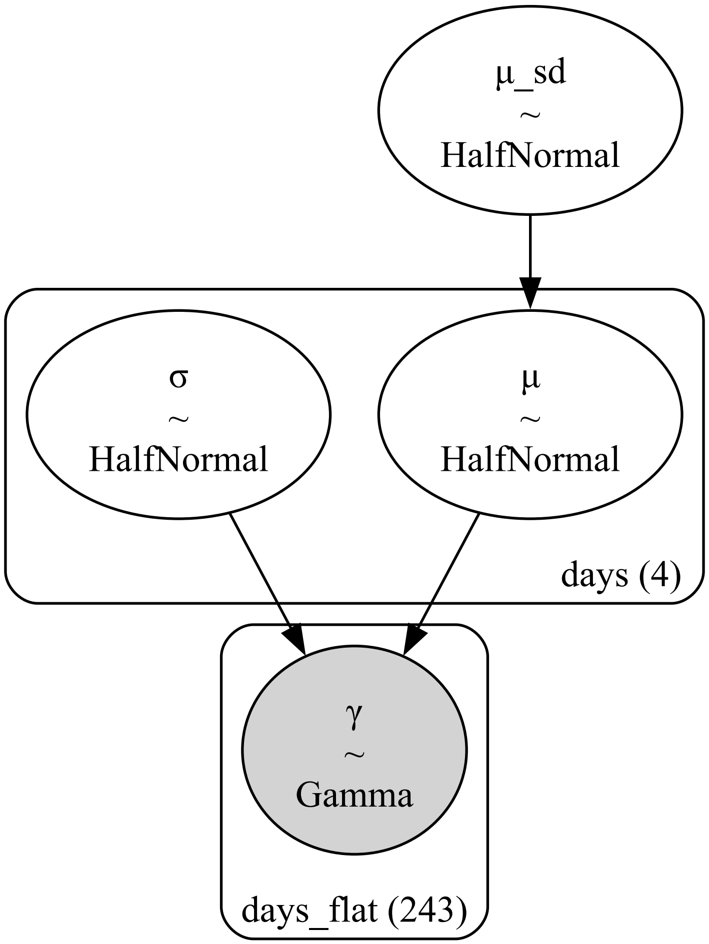
```
Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_000 draws total) took 51 seconds.
Sampling: [γ]
Output()
```

- **μ_sd**: day-to-day differences in tipping behavior
- **μ**: mean tips for each day
- **σ**: within-day differences in individual tipping behavior
- **γ**: likelihood of observed tips, using a Gamma distribution

---

In [27]:
```
pm.model_to_graphviz(comparing_days)
```

Out[27]:



- The model assumes that the daily mean tips come from a common distribution defined by the hyperprior. This was done with the belief that while tips on different days might vary, they are related because they come from the same restaurant and tipping behavior has some consistency.

- By using this hyperprior, the model can shrink extreme estimates of μ for days with limited data toward the overall average. This helps prevent overfitting to noisy data.

Model Checks

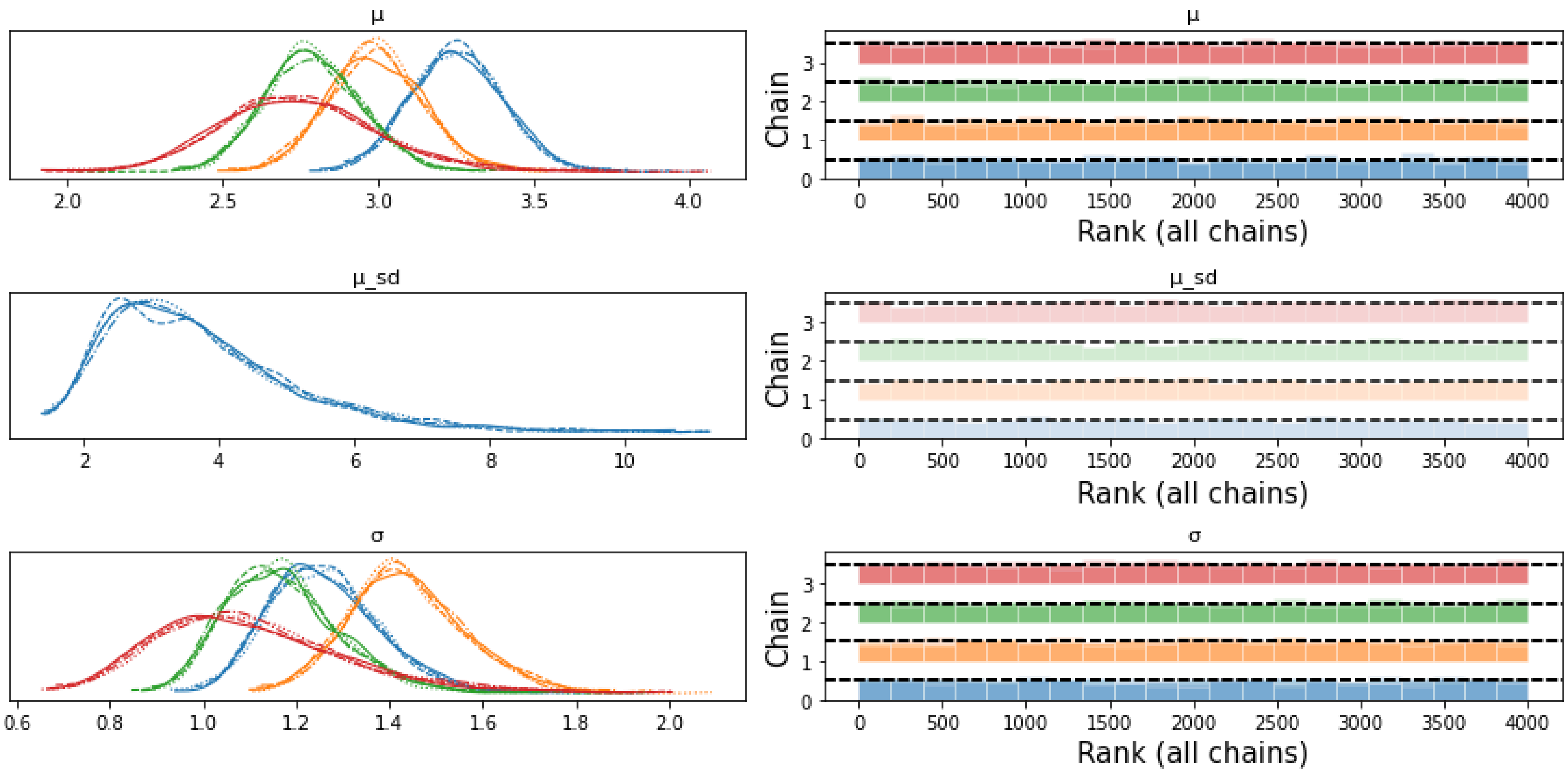
In [28]:
```
az.loo(idata_cd)
```

Out[28]:
```
Computed from 4000 posterior samples and 243 observations log-likelihood matrix.

         Estimate       SE
elpd_loo  -394.25    13.35
p_loo        7.57        -
------

Pareto k diagnostic values:
                         Count   Pct.
(-Inf, 0.70]   (good)      243  100.0%
   (0.70, 1]   (bad)         0    0.0%
   (1, Inf)   (very bad)    0    0.0%
```

- Multiple models are not being compared, so the only metrics of interest here are the Pareto k diagnostic values.

- These values indicate that no single data point has an excessive influence on the posterior, and there's no need for further adjustment of the data points.

In [29]:
```
az.plot_trace(idata_cd, kind='rank_bars', combined=False)
plt.tight_layout();
```



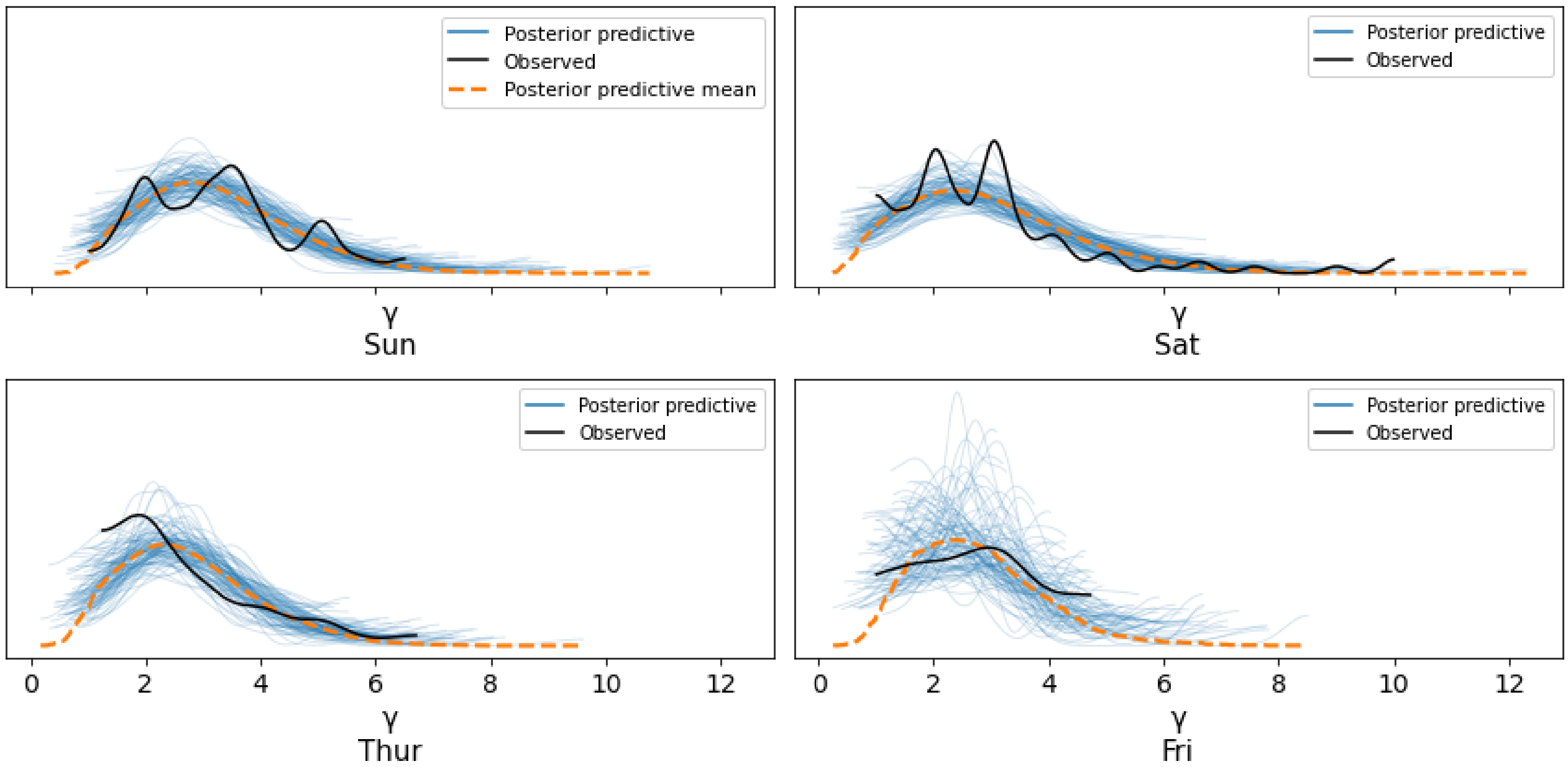
- On the left hand side, the KDE plots for all chains mostly overlap, meaning that all chains are sampling from the same posterior distribution.

- Likewise, on the right hand side, the color bands are fairly uniform across the chains, indicating that the sampling is well-mixed and balanced.

In [30]:

```
_,axes = plt.subplots(2,2, figsize=(12,6), sharex=True, sharey=True)

az.plot_ppc(idata_cd, num_pp_samples=100,
            coords={'days_flat':[categories]}, flatten=[], ax=axes)
```

```
plt.tight_layout()
plt.show()
```



- Some details are not being captured which could be because of the fairly small sample size, factors others than day influencing the tips, or a combination of the two.

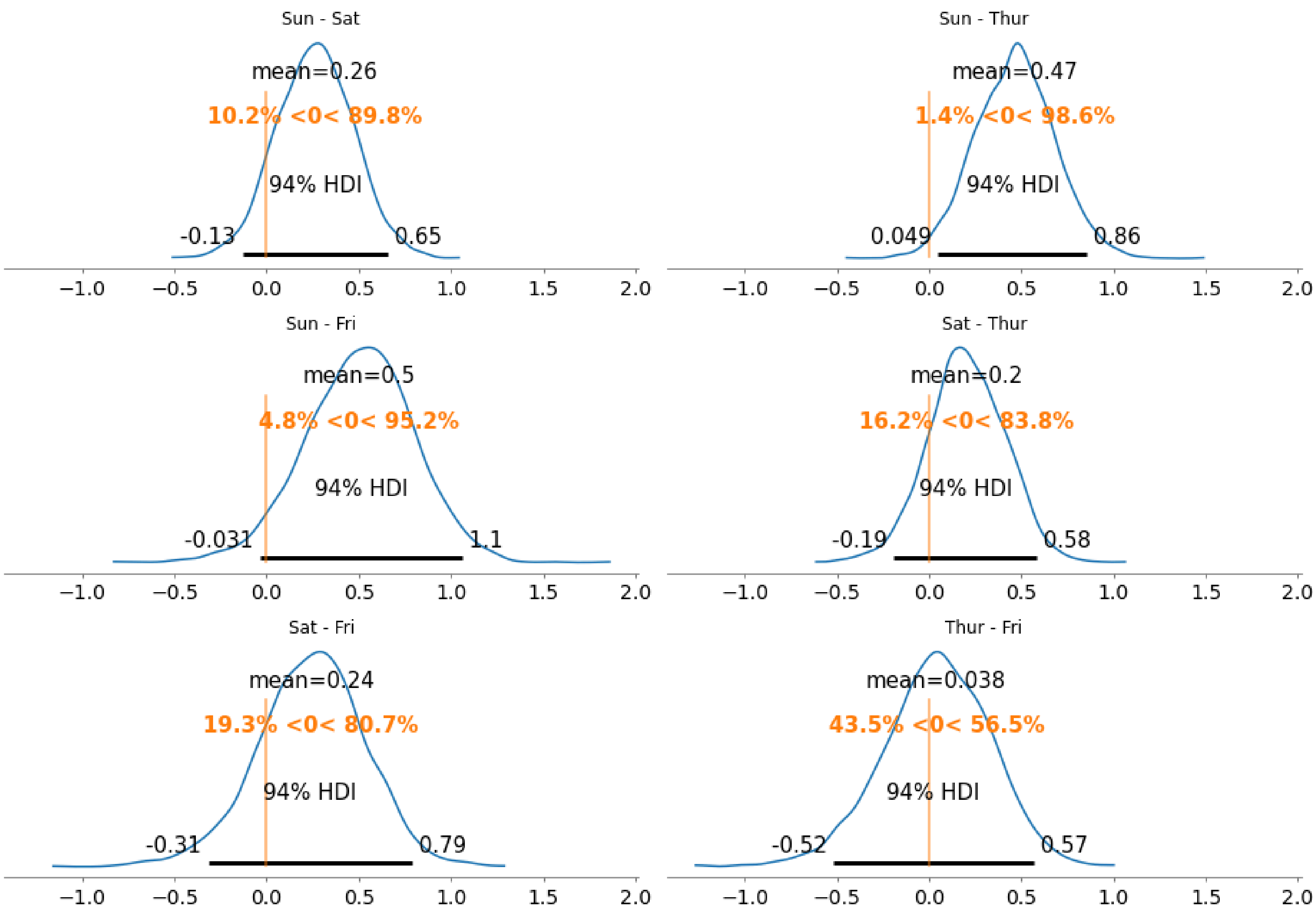
- However,the models do capture the general shape of the distributions, and these are being considered good enough to proceed.

Results

- To make a decision on which day to work, the results are expressed in terms of differences in posterior means.

In [31]:

```
cd_posterior = az.extract(idata_cd)

dist = pz.Normal(0, 1)

comparisons = [(categories[i], categories[j]) for i in range(4) for j in range(i+1, 4)]

_, axes = plt.subplots(3, 2, figsize=(13, 9), sharex=True)

for (i, j), ax in zip(comparisons, axes.ravel()):
    means_diff = cd_posterior["μ"].sel(days=i) - cd_posterior['μ'].sel(days=j)
    az.plot_posterior(means_diff.values, ref_val=0, ax=ax)
    ax.set_title(f"{i} - {j}")
    plt.tight_layout()
```



Taking the top left plot as an example:

- The blue curve is the posterior distribution of the difference in mean tips between Sunday and Saturday. The mean difference per tip is 0.26.

- For the orange text, there is a 10.2% chance that Saturday has higher mean tips than Sunday, and a 89.8% chance that Sunday has higher mean tips than Saturday.

- The 94% HDI (-0.13 to 0.65) represents the range in which 94% of the posterior distribution lies.

- The vertical line is a reference line for 0.

- For all of the differences except Sun-Thur, 0 is included in the HDI. This means the possibility of no difference cannot be ruled out. That being said, Sunday consistently outperforms all other days in terms of mean tip amounts. Saturday is second-best, but its advantage over other days is smaller and less certain.

Conclusion

Given these results, I will choose Sunday as my day to work.

This project was inspired by an example from Bayesian Analysis with Python Third Edition by Osvaldo Martin, but has been significantly changed and expanded upon.