# Country-Specific Prophet

**This approach creates individual Prophet models for each country of residence:**

- **Each country gets its own forecasting model trained on its historical data**
- **Captures country-specific seasonality patterns and trends**
- **Predicts which countries will produce the most arrivals in future months**
- **Naturally handles the "Country of Residence" column by training separate models**

---

**This might be a bad solution knowing each country has less data when separated to its clusters and each country would need extra supervision just to forecast accurately.**

In [ ]:

```
import pandas as pd
from datetime import datetime, timedelta
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from prophet import Prophet
from prophet.diagnostics import cross_validation, performance_metrics
from prophet.plot import plot_cross_validation_metric
import warnings
import os
import plotly.graph_objects as go
import holidays
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, mean_absol
ute_percentage_error
warnings.filterwarnings('ignore')

plt.style.use('ggplot')
sns.set_palette("colorblind")
pd.set_option('display.float_format', '{:.10f}'.format)
```

# Load the CSV

In [ ]:

```
df = pd.read_csv("/content/drive/MyDrive/dataset/tourism/final-raw-dataset.csv")
df.head()
```

Out[ ]:

| | Year | Country of Residence | January | February | March | April | May | June | July | August | September | October | November | Decemb |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2008 | ARGENTINA | 60 | 41 | 54 | 38 | 58 | 38 | 29 | 46 | 59 | 72 | 55 | |
| 1 | 2008 | AUSTRALIA | 11218 | 8861 | 10857 | 9904 | 9630 | 8554 | 8412 | 7965 | 9077 | 9215 | 9542 | 182 |
| 2 | 2008 | AUSTRIA | 1145 | 1028 | 990 | 829 | 555 | 709 | 1178 | 570 | 451 | 610 | 769 | 12 |
| 3 | 2008 | BAHRAIN | 226 | 160 | 200 | 239 | 221 | 303 | 403 | 258 | 130 | 308 | 198 | 2 |
| 4 | 2008 | BANGLADESH | 114 | 128 | 162 | 151 | 202 | 125 | 144 | 199 | 186 | 223 | 179 | 1 |

# Data prep

## Reshape to long format

In [ ]:

```
df_long = pd.melt(
    df,
    id_vars=['Year', 'Country of Residence'],
    value_vars=['January', 'February', 'March', 'April', 'May', 'June',
                'July', 'August', 'September', 'October', 'November', 'December'],
    var_name='Month',
    value_name='Arrivals'
)

# Create a proper date column
month_map = {
    'January': 1, 'February': 2, 'March': 3, 'April': 4, 'May': 5, 'June': 6,
    'July': 7, 'August': 8, 'September': 9, 'October': 10, 'November': 11, 'December': 1
2
}

df_long['Month_Num'] = df_long['Month'].map(month_map)
df_long['Date'] = pd.to_datetime(df_long['Year'].astype(str) + '-' +
                                 df_long['Month_Num'].astype(str) + '-01')
```

## Remove future rows on current dataset

In [ ]:

```
cutoff_index = df_long[(df_long['Date'] == pd.to_datetime('2025-04-01'))].tail(1).index[
0]
df_long = df_long[df_long['Date'] <= '2025-04-01']
df_long = df_long.sort_values(['Date', 'Country of Residence'])
df_long = df_long.reset_index(drop=True)
df_long
```

Out[ ]:

| | Year | Country of Residence | Month | Arrivals | Month_Num | Date |
|---|---|---|---|---|---|---|
| **0** | 2008 | ARGENTINA | January | 60 | 1 | 2008-01-01 |
| **1** | 2008 | AUSTRALIA | January | 11218 | 1 | 2008-01-01 |
| **2** | 2008 | AUSTRIA | January | 1145 | 1 | 2008-01-01 |
| **3** | 2008 | BAHRAIN | January | 226 | 1 | 2008-01-01 |
| **4** | 2008 | BANGLADESH | January | 114 | 1 | 2008-01-01 |
| **...** | ... | ... | ... | ... | ... | ... |
| **20439** | 2025 | VIRGIN ISLANDS (U.S.) | April | 2 | 4 | 2025-04-01 |
| **20440** | 2025 | WALLIS AND FUTUNA | April | 0 | 4 | 2025-04-01 |
| **20441** | 2025 | YEMEN | April | 3 | 4 | 2025-04-01 |
| **20442** | 2025 | ZAMBIA | April | 12 | 4 | 2025-04-01 |
| **20443** | 2025 | ZIMBABWE | April | 29 | 4 | 2025-04-01 |

**20444 rows × 6 columns**

## Check for zero values

In [ ]:

```
data_quality = df_long.groupby('Country of Residence').agg(
        total_records=('Arrivals', 'count'),
        zero_values=('Arrivals', lambda x: (x == 0).sum()),
        missing_values=('Arrivals', lambda x: x.isna().sum()),
        percent_zeros=('Arrivals', lambda x: (x == 0).sum() / len(x) * 100),
        mean_arrivals=('Arrivals', 'mean'),
```

```
        max_arrivals=('Arrivals', 'max')
).sort_values('percent_zeros')
data_quality
```

Out[ ]:

| Country of Residence | total_records | zero_values | missing_values | percent_zeros | mean_arrivals | max_arrivals |
|---|---|---|---|---|---|---|
| ALAND ISLANDS | 28 | 0 | 0 | 0.0000000000 | 8.3928571429 | 24 |
| BRITISH INDIAN OCEAN TERRITORY | 28 | 0 | 0 | 0.0000000000 | 15.2857142857 | 30 |
| JERSEY | 28 | 0 | 0 | 0.0000000000 | 30.5000000000 | 60 |
| GUERNSEY | 28 | 0 | 0 | 0.0000000000 | 16.3214285714 | 33 |
| HONGKONG | 168 | 0 | 0 | 0.0000000000 | 8429.4702380952 | 14997 |
| ... | ... | ... | ... | ... | ... | ... |
| COCOS (KEELING) ISLANDS | 24 | 21 | 0 | 87.5000000000 | 0.2500000000 | 4 |
| MONTSERRAT | 24 | 22 | 0 | 91.6666666667 | 0.1250000000 | 2 |
| SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS | 24 | 22 | 0 | 91.6666666667 | 0.0833333333 | 1 |
| SAINT PIERRE AND MIQUELON | 28 | 26 | 0 | 92.8571428571 | 0.0714285714 | 1 |
| WESTERN SAHARA | 36 | 34 | 0 | 94.4444444444 | 0.0555555556 | 1 |

249 rows × 6 columns

# Identify countries with consistent data (less than 10% zeros)

In [ ]:

```
main_countries = data_quality[(data_quality['percent_zeros'] < 10) & (data_quality['tota
l_records'] == 208)].index.tolist()
main_countries
```

Out[ ]:

```
['ITALY',
 'JAPAN',
 'GERMANY',
 'IRELAND',
 'BRAZIL',
 'CANADA',
 'BAHRAIN',
 'AUSTRALIA',
 'NORWAY',
 'SOUTH AFRICA',
 'SWITZERLAND',
 'THAILAND',
 'VIETNAM',
 'USA',
 'UNITED KINGDOM',
 'SPAIN',
 'NETHERLANDS',
 'POLAND',
 'RUSSIA',
 'NEW ZEALAND',
 'MALAYSIA',
 'MEXICO',
 'KOREA',
 'CHINA',
 'FINLAND',
 'FRANCE',
 'INDONESIA',
 'INDIA',
 'BELGIUM',
```

```
 'PORTUGAL',
 'NEPAL',
 'PAKISTAN',
 'SWEDEN',
 'SAUDI ARABIA',
 'SINGAPORE',
 'UNITED ARAB EMIRATES',
 'PAPUA NEW GUINEA',
 'TAIWAN',
 'ARGENTINA',
 'MYANMAR',
 'PERU',
 'NIGERIA',
 'SRI LANKA',
 'GREECE',
 'EGYPT',
 'KUWAIT',
 'JORDAN',
 'DENMARK',
 'AUSTRIA',
 'ISRAEL',
 'GUAM',
 'COLOMBIA',
 'VENEZUELA',
 'CAMBODIA',
 'IRAN',
 'BANGLADESH',
 'BRUNEI',
 'LAOS',
 'LUXEMBOURG']
```

## Group inconsistent countries

In [ ]:

```
df_grouped = df_long.copy()
df_grouped['Country of Residence'] = df_grouped['Country of Residence'].apply(
    lambda x: x if x in main_countries else 'Others'
)
```

In [ ]:

```
# Aggregate arrivals for 'Others' by date to avoid duplicate rows
main_df = df_grouped[df_grouped['Country of Residence'].isin(main_countries)]
others_df = df_grouped[~df_grouped['Country of Residence'].isin(main_countries)]

# Aggregate Others by summing arrivals on the same date
others_agg = others_df.groupby(['Year', 'Country of Residence', 'Month', 'Month_Num', 'Date']).agg({
    'Arrivals': 'sum'
}).reset_index()

df_grouped = pd.concat([main_df, others_agg], ignore_index=True).sort_values(['Date', 'Country of Residence'])
df_grouped
```

Out[ ]:

| | Year | Country of Residence | Month | Arrivals | Month_Num | Date |
|---|---|---|---|---|---|---|
| 0 | 2008 | ARGENTINA | January | 60 | 1 | 2008-01-01 |
| 1 | 2008 | AUSTRALIA | January | 11218 | 1 | 2008-01-01 |
| 2 | 2008 | AUSTRIA | January | 1145 | 1 | 2008-01-01 |
| 3 | 2008 | BAHRAIN | January | 226 | 1 | 2008-01-01 |
| 4 | 2008 | BANGLADESH | January | 114 | 1 | 2008-01-01 |
| ... | ... | ... | ... | ... | ... | ... |

| | Year | Country of Residence | Month | Arrivals | Month_Num | Date |
|---|---|---|---|---|---|---|
| 12267 | 2025 | UNITED ARAB EMIRATES | April | 2804 | 4 | 2025-04-01 |
| 12268 | 2025 | UNITED KINGDOM | April | 18310 | 4 | 2025-04-01 |
| 12269 | 2025 | USA | April | 75202 | 4 | 2025-04-01 |
| 12270 | 2025 | VENEZUELA | April | 11 | 4 | 2025-04-01 |
| 12271 | 2025 | VIETNAM | April | 2573 | 4 | 2025-04-01 |

**12480 rows × 6 columns**

## Handle January 2022 missing values

**A temporary solution**

In [ ]:

```
df_fixed = df_grouped.copy()
```

In [ ]:

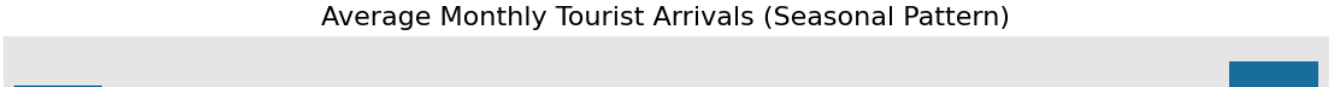
```
jan_2022_mask = (df_fixed['Year'] == 2022) & (df_fixed['Month'] == 'January') & (df_fixe
d['Arrivals'] == 0)
jan_2022_indices = df_fixed[jan_2022_mask].index

for idx in jan_2022_indices:
    country = df_fixed.loc[idx, 'Country of Residence']

    dec_2021_value = df_fixed[(df_fixed['Year'] == 2021) &
                              (df_fixed['Month'] == 'December') &
                              (df_fixed['Country of Residence'] == country)]['Arrivals']
.values

    feb_2022_value = df_fixed[(df_fixed['Year'] == 2022) &
                              (df_fixed['Month'] == 'February') &
                              (df_fixed['Country of Residence'] == country)]['Arrivals']
.values

    if len(dec_2021_value) > 0 and len(feb_2022_value) > 0:
        interpolated_value = (dec_2021_value[0] + feb_2022_value[0]) / 2
        df_fixed.loc[idx, 'Arrivals'] = interpolated_value
```

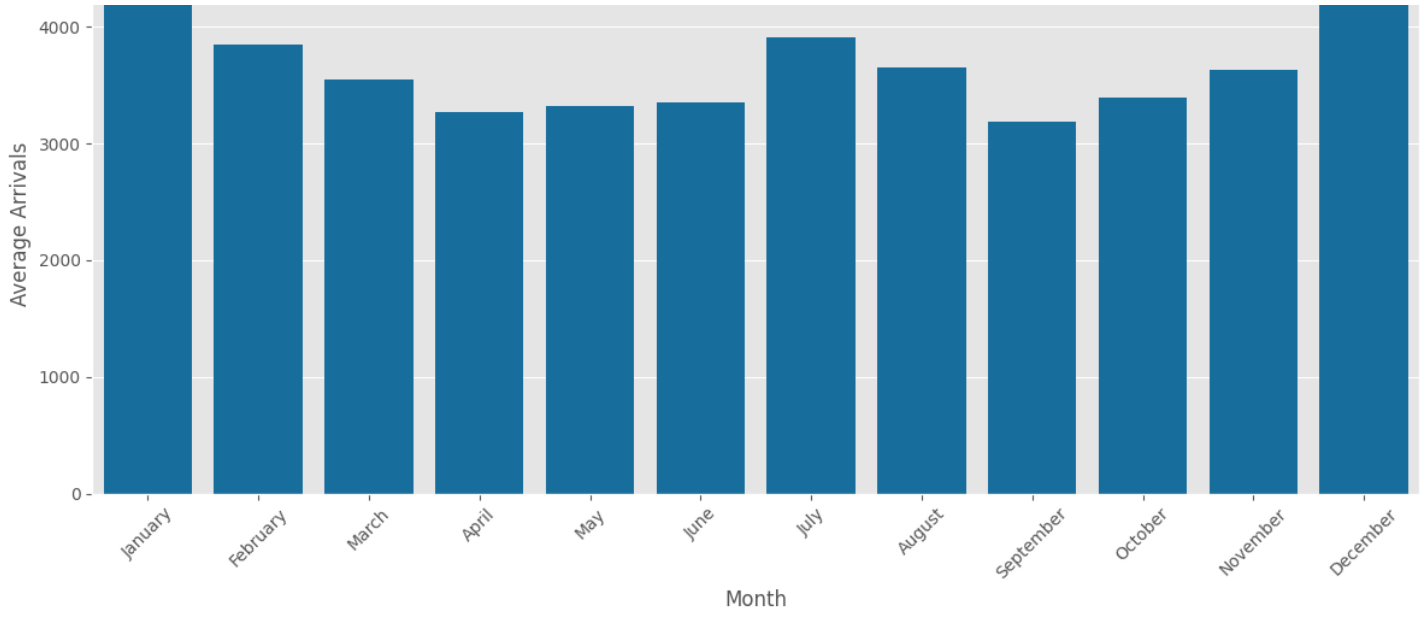
# Data visualizations

## Visualize seasonal patterns

In [ ]:

```
monthly_avg = df_long.groupby(['Month_Num', 'Month'])['Arrivals'].mean().reset_index()
monthly_avg = monthly_avg.sort_values('Month_Num')

plt.figure(figsize=(12, 6))
sns.barplot(x='Month', y='Arrivals', data=monthly_avg, order=[
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
])
plt.title('Average Monthly Tourist Arrivals (Seasonal Pattern)', fontsize=16)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Average Arrivals', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('seasonal_patterns.png')
plt.show()
```






# Prophet model

## Adjust regular holidays

`holidays` **package in pip is so helpful here to obtain list of country holidays from 2008 - 2025**

In [ ]:

```
data_years = list(range(df_long['Date'].dt.year.min(), df_long['Date'].dt.year.max() + 1
))

country_code = 'PH'
country_holidays = holidays.country_holidays(country_code, years=data_years)

holiday_df = pd.DataFrame(
    [(name, date) for date, name in country_holidays.items()],
    columns=['holiday', 'ds']
)

lower_window = 0
upper_window = 0

holiday_df['lower_window'] = lower_window
holiday_df['upper_window'] = upper_window
holiday_df = holiday_df.sort_values(by='ds').reset_index(drop=True)
holiday_df
```

Out[ ]:

| | holiday | ds | lower_window | upper_window |
|---|---|---|---|---|
| 0 | New Year's Day | 2008-01-01 | 0 | 0 |
| 1 | Maundy Thursday | 2008-03-20 | 0 | 0 |
| 2 | Good Friday | 2008-03-21 | 0 | 0 |
| 3 | Day of Valor | 2008-04-07 | 0 | 0 |
| 4 | Labor Day | 2008-05-01 | 0 | 0 |
| ... | ... | ... | ... | ... |
| 336 | Immaculate Conception | 2025-12-08 | 0 | 0 |
| 337 | Christmas Eve | 2025-12-24 | 0 | 0 |
| 338 | Christmas Day | 2025-12-25 | 0 | 0 |
| 339 | Rizal Day | 2025-12-30 | 0 | 0 |

| | holiday | ds | lower_window | upper_window |
|---|---|---|---|---|
| 340 | New Year's Eve | 2025-12-31 | 0 | 0 |

341 rows × 4 columns

In [ ]:

```
holiday_adjusted = holiday_df.copy()
```

In [ ]:

```
holiday_adjusted['ds'] = pd.to_datetime(holiday_adjusted['ds'])
holiday_adjusted['ds'] = holiday_adjusted['ds'].dt.strftime('%Y-%m-01')
holiday_adjusted['ds'] = pd.to_datetime(holiday_adjusted['ds'])
holiday_adjusted = holiday_adjusted.drop_duplicates(subset=['holiday', 'ds'])
holiday_adjusted
```

Out[ ]:

| | holiday | ds | lower_window | upper_window |
|---|---|---|---|---|
| 0 | New Year's Day | 2008-01-01 | 0 | 0 |
| 1 | Maundy Thursday | 2008-03-01 | 0 | 0 |
| 2 | Good Friday | 2008-03-01 | 0 | 0 |
| 3 | Day of Valor | 2008-04-01 | 0 | 0 |
| 4 | Labor Day | 2008-05-01 | 0 | 0 |
| ... | ... | ... | ... | ... |
| 336 | Immaculate Conception | 2025-12-01 | 0 | 0 |
| 337 | Christmas Eve | 2025-12-01 | 0 | 0 |
| 338 | Christmas Day | 2025-12-01 | 0 | 0 |
| 339 | Rizal Day | 2025-12-01 | 0 | 0 |
| 340 | New Year's Eve | 2025-12-01 | 0 | 0 |

339 rows × 4 columns

# Handling COVID shocks

> **To prevent large dips and spikes from being captured by the trend component, we can treat the days impacted by COVID-19 as holidays that will not repeat again in the future.**

**Based on plotted data, these are likely the possible cases. These dates will be adjusted to first of the month since we have regular data gaps**

---

**Reference:**

- https://facebook.github.io/prophet/docs/handling_shocks.html

## Aggregate arrivals column by dates

**Sum all arrivals by date so we can visualize what is needed to be used as a lockdown holiday for Prophet.**

In [ ]:

```
df_agg_date = df_long[['Date', 'Arrivals']].groupby('Date').sum().reset_index()
df_2019_2023 = df_agg_date[(df_agg_date['Date'].dt.year >= 2019) & (df_agg_date['Date'].
dt.year <= 2023)]
```

**Use `plotly.graph_objects.go` for plotting so we can use cursor to determine which needs to be logged as lockdown holidays**

**lockdown holidays**

In [ ]:

```
fig = go.Figure()

fig.add_trace(
    go.Scatter(
        x=df_2019_2023['Date'],
        y=df_2019_2023['Arrivals'],
        mode='lines',
        name='Tourist Arrivals',
        line=dict(color='royalblue')
    )
)

fig.update_layout(
    title='Total Monthly Tourist Arrivals (2019-2023)',
    title_font=dict(size=16),
    xaxis_title='Date',
    yaxis_title='Total Arrivals',
    xaxis_title_font=dict(size=12),
    yaxis_title_font=dict(size=12),
    width=1000,
    height=500,
    template='plotly_white'
)

fig.show()
```

In [ ]:

```
lockdowns = pd.DataFrame([
    {
        'holiday': 'covid_impact_1',  # First major drop
        'ds': '2020-02-01',  # Starting with February 2020 when arrivals began dropping
        'lower_window': 0,
        'ds_upper': '2020-12-01'  # Through the end of 2020
    },
    {
```

```
        'holiday': 'covid_impact_2',   # Continued low levels
        'ds': '2021-01-01',
        'lower_window': 0,
        'ds_upper': '2021-12-01'   # Through the end of 2021
    },
    {
        'holiday': 'covid_recovery',   # Recovery period
        'ds': '2022-01-01',
        'lower_window': 0,
        'ds_upper': '2022-07-01'   # First half of 2022 when recovery was still ongoing
    }
])

for t_col in ['ds', 'ds_upper']:
    lockdowns[t_col] = pd.to_datetime(lockdowns[t_col])

lockdowns['upper_window'] = (lockdowns['ds_upper'] - lockdowns['ds']).dt.days
lockdowns
```

Out[ ]:

| | holiday | ds | lower_window | ds_upper | upper_window |
|---|---|---|---|---|---|
| 0 | covid_impact_1 | 2020-02-01 | 0 | 2020-12-01 | 304 |
| 1 | covid_impact_2 | 2021-01-01 | 0 | 2021-12-01 | 334 |
| 2 | covid_recovery | 2022-01-01 | 0 | 2022-07-01 | 181 |

# Building Prophet

***The following recommendations are tested on aggregated data. Not country-specific data. Here is the notebook for aggregated forecasting: link***

**For training:**

- **We'll be using `multiplicative` seasonality here since that fits our problem. Tho this might cause a problem for calculating MAPE**
- **PH Holidays are added but adjusted to first day of month since our aggregated (monthly) data has regular gaps.**
- **COVID19 phase are treated as holidays also**

**Performance metrics ($R^2$ score) of model if it implements the ff.:**

| Configuration | $R^2$ Score | Improvement over Base |
|---|---|---|
| Base Model (No Binary Month Regressors & No Pre/Post COVID Seasonality) | 0.915 | - |
| With Binary Month Regressors Only | 0.914 | -0.001 |
| With Pre/Post COVID Seasonality Only | 0.902 | -0.013 |
| With Both Features | **0.935** | +0.020 |

**The synergistic effect of both features together yields better results than either feature individually.**

**For forecasting:**

- **We should use `MS` since Prophet will only be able to see the first day of a month. MS stands for Month Start**

---

**References:**


- https://facebook.github.io/prophet/docs/non-daily_data.html#monthly-data
- https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#fourier-order-for-seasonalities
- https://facebook.github.io/prophet/docs/handling_shocks.html#changes-in-seasonality-between-pre--and-post-covid
- https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#built-in-country-

holidays
- https://facebook.github.io/prophet/docs/non-daily_data.html#holidays-with-aggregated-data

## Define functions

In [ ]:

```
def train_prophet_model(df_prophet, periods, country_name=None):
    model = Prophet(
        yearly_seasonality=False,
        seasonality_mode='multiplicative',
        holidays=pd.concat([lockdowns, holiday_adjusted]),
    )

    months = ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov
', 'dec']

    for i, month in enumerate(months, 1):
        df_prophet[f'is_{month}'] = (df_prophet['ds'].dt.month == i).astype(int)

    for month in months:
        model.add_regressor(f'is_{month}')

    covid_outbreak_date = '2020-02-01'
    covid_end_recovery_date = '2023-07-01'

    df_prophet['pre_covid'] = pd.to_datetime(df_prophet['ds']) < pd.to_datetime(covid_ou
tbreak_date)
    df_prophet['has_covid'] = (
        (pd.to_datetime(df_prophet['ds']) > pd.to_datetime(covid_outbreak_date)) &
        (pd.to_datetime(df_prophet['ds']) < pd.to_datetime(covid_end_recovery_date))
    )

    monthly_period = 365.5
    fourier_order = 10
    model.add_seasonality(name='monthly_pre_covid', period=monthly_period, fourier_order=
fourier_order, condition_name='pre_covid')
    model.add_seasonality(name='monthly_has_covid', period=monthly_period, fourier_order=
fourier_order, condition_name='has_covid')

    model.fit(df_prophet)

    future = model.make_future_dataframe(periods=periods, freq='MS')

    future['pre_covid'] = pd.to_datetime(future['ds']) < pd.to_datetime(covid_outbreak_d
ate)
    future['has_covid'] = (
        (pd.to_datetime(future['ds']) > pd.to_datetime(covid_outbreak_date)) &
        (pd.to_datetime(future['ds']) < pd.to_datetime(covid_end_recovery_date))
    )

    for i, month in enumerate(months, 1):
        future[f'is_{month}'] = (future['ds'].dt.month == i).astype(int)

    forecast = model.predict(future)

    return model, forecast
```

In [ ]:

```
def prepare_prophet_data(df_long, country=None):
    df_prophet = df_long[df_long['Country of Residence'] == country].copy()
    df_prophet = df_prophet.reset_index().rename(columns={'Date': 'ds', 'Arrivals': 'y'}
)

    return df_prophet
```

## Start training

In [ ]:

```
all_countries = df_fixed['Country of Residence'].unique().tolist()

years_to_predict = 6
months_to_predict = 12 * years_to_predict

country_models = {}
country_forecasts = {}

for country in all_countries:
  print(f"\nModeling {country} tourist arrivals...")

  df_prophet = prepare_prophet_data(df_fixed, country)
  model, forecast = train_prophet_model(df_prophet, months_to_predict)

  country_models[country] = model
  country_forecasts[country] = forecast
```

```
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling ARGENTINA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/n7feio45.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/51vwze10.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=37950', 'data', 'file=/tmp/tmpunk9xwl5/n7feio45.jso
n', 'init=/tmp/tmpunk9xwl5/51vwze10.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
kr5e86o4/prophet_model-20250519093612.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:12 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:13 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling AUSTRALIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/knp9y8yx.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ed6m6e9w.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=67302', 'data', 'file=/tmp/tmpunk9xwl5/knp9y8yx.jso
n', 'init=/tmp/tmpunk9xwl5/ed6m6e9w.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
gdyj9ng2/prophet_model-20250519093614.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:14 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:14 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling AUSTRIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/halakbj3.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/s8itxg35.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=71115', 'data', 'file=/tmp/tmpunk9xwl5/halakbj3.jso
n', 'init=/tmp/tmpunk9xwl5/s8itxg35.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
```

```
u93q32m1/prophet_model-20250519093615.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:15 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:16 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling BAHRAIN tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/g4g7mt2e.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/qp7i650e.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=12032', 'data', 'file=/tmp/tmpunk9xwl5/g4g7mt2e.jso
n', 'init=/tmp/tmpunk9xwl5/qp7i650e.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
1p695l66/prophet_model-20250519093617.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:17 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:17 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling BANGLADESH tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/pzd59x43.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/01xb44q6.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=98175', 'data', 'file=/tmp/tmpunk9xwl5/pzd59x43.jso
n', 'init=/tmp/tmpunk9xwl5/01xb44q6.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
icur7teo/prophet_model-20250519093618.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:18 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:18 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling BELGIUM tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/0711c7r7.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ux0fv37a.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=52426', 'data', 'file=/tmp/tmpunk9xwl5/0711c7r7.jso
n', 'init=/tmp/tmpunk9xwl5/ux0fv37a.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
8fpx7rid/prophet_model-20250519093620.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:20 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:20 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling BRAZIL tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/zhrnmowy.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/opw28nk3.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=54011', 'data', 'file=/tmp/tmpunk9xwl5/zhrnmowy.jso
n', 'init=/tmp/tmpunk9xwl5/opw28nk3.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
fjp62p3_/prophet_model-20250519093621.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:21 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:22 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling BRUNEI tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/b9v4605w.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ozs4el70.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=97603', 'data', 'file=/tmp/tmpunk9xwl5/b9v4605w.jso
n', 'init=/tmp/tmpunk9xwl5/ozs4el70.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
7_ucfxk6/prophet_model-20250519093623.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:23 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:24 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling CAMBODIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/38nklshb.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/u_9vyqo8.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=7520', 'data', 'file=/tmp/tmpunk9xwl5/38nklshb.json
', 'init=/tmp/tmpunk9xwl5/u_9vyqo8.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelo
ta15qn9/prophet_model-20250519093626.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10
000']
09:36:26 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:26 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling CANADA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/jdle02h4.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/btx6274z.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=46608', 'data', 'file=/tmp/tmpunk9xwl5/jdle02h4.jso
n', 'init=/tmp/tmpunk9xwl5/btx6274z.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
iw1nu3tf/prophet_model-20250519093627.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:27 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:28 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
```

```
INFO:propnet:Disabling weekly seasonality. Run propnet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling CHINA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/x7l90uav.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/zbr10ozv.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=32116', 'data', 'file=/tmp/tmpunk9xwl5/x7l90uav.jso
n', 'init=/tmp/tmpunk9xwl5/zbr10ozv.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
6wiokg2u/prophet_model-20250519093629.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:29 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:30 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling COLOMBIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/zrm4t0st.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/j9sjx_bv.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=56710', 'data', 'file=/tmp/tmpunk9xwl5/zrm4t0st.jso
n', 'init=/tmp/tmpunk9xwl5/j9sjx_bv.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
7xv5pfei/prophet_model-20250519093631.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:31 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:32 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling DENMARK tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/m69rmqv5.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/9y6kobws.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=18328', 'data', 'file=/tmp/tmpunk9xwl5/m69rmqv5.jso
n', 'init=/tmp/tmpunk9xwl5/9y6kobws.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
cqiujhi4/prophet_model-20250519093633.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:33 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:33 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/m_8itmq_.json
```

```
Modeling EGYPT tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/gm6yddht.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=39811', 'data', 'file=/tmp/tmpunk9xwl5/m_8itmq_.jso
n', 'init=/tmp/tmpunk9xwl5/gm6yddht.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
```

```
q90q5atc/prophet_model-20250519093634.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:34 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:34 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling FINLAND tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/5wb_qb3t.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/nwlo_182.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=70353', 'data', 'file=/tmp/tmpunk9xwl5/5wb_qb3t.jso
n', 'init=/tmp/tmpunk9xwl5/nwlo_182.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
8ifswco8/prophet_model-20250519093635.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:35 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:36 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling FRANCE tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/crv7pzdb.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/i6kq4a7b.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=18179', 'data', 'file=/tmp/tmpunk9xwl5/crv7pzdb.jso
n', 'init=/tmp/tmpunk9xwl5/i6kq4a7b.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
l4znyylx/prophet_model-20250519093637.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:37 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:38 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling GERMANY tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/rqves5ir.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/nw6499h3.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=40628', 'data', 'file=/tmp/tmpunk9xwl5/rqves5ir.jso
n', 'init=/tmp/tmpunk9xwl5/nw6499h3.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
ihbug1ae/prophet_model-20250519093639.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:39 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:40 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling GREECE tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/505ybwqi.json
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/505ybwgi.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/f77ekufb.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=33019', 'data', 'file=/tmp/tmpunk9xwl5/505ybwgi.json', 'init=/tmp/tmpunk9xwl5/f77ekufb.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelddy8mcwy/prophet_model-20250519093641.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:36:41 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:41 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling GUAM tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/dy5762hl.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/xpyg6h0w.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=48511', 'data', 'file=/tmp/tmpunk9xwl5/dy5762hl.json', 'init=/tmp/tmpunk9xwl5/xpyg6h0w.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelgm7rv3zb/prophet_model-20250519093642.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:36:42 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:43 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling INDIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/z410dogh.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/1lqi9_23.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=81426', 'data', 'file=/tmp/tmpunk9xwl5/z410dogh.json', 'init=/tmp/tmpunk9xwl5/1lqi9_23.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model7wkacvvg/prophet_model-20250519093644.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:36:44 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:44 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling INDONESIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/r47s6jzm.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/donuua8j.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=92894', 'data', 'file=/tmp/tmpunk9xwl5/r47s6jzm.json', 'init=/tmp/tmpunk9xwl5/donuua8j.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model627aisvx/prophet_model-20250519093645.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:36:45 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:46 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
```

```
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling IRAN tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/24lpt_zs.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/7kb5swv6.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=27194', 'data', 'file=/tmp/tmpunk9xwl5/24lpt_zs.jso
n', 'init=/tmp/tmpunk9xwl5/7kb5swv6.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
lm5__4ml/prophet_model-20250519093646.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:46 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:47 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ft7fexo4.json
```

Modeling IRELAND tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/gk2f_m4b.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=83123', 'data', 'file=/tmp/tmpunk9xwl5/ft7fexo4.jso
n', 'init=/tmp/tmpunk9xwl5/gk2f_m4b.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
fjyc0cfn/prophet_model-20250519093648.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:48 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:48 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling ISRAEL tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/5avjdhhh.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/9o8kxa31.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=25840', 'data', 'file=/tmp/tmpunk9xwl5/5avjdhhh.jso
n', 'init=/tmp/tmpunk9xwl5/9o8kxa31.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
gj73bxf7/prophet_model-20250519093649.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:49 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:50 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/vjwiuxqb.json
```

Modeling ITALY tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/lqrqa02u.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=7216', 'data', 'file=/tmp/tmpunk9xwl5/vjwiuxqb.json
', 'init=/tmp/tmpunk9xwl5/lqrqa02u.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model2
```

```
b6jrip7/prophet_model-20250519093651.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10
000']
09:36:51 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:52 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling JAPAN tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/laffj6do.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/s8ioald7.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=38397', 'data', 'file=/tmp/tmpunk9xwl5/laffj6do.jso
n', 'init=/tmp/tmpunk9xwl5/s8ioald7.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
vft9idys/prophet_model-20250519093653.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:53 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:54 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling JORDAN tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/4ch2qdop.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/wlg6d5xb.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=17538', 'data', 'file=/tmp/tmpunk9xwl5/4ch2qdop.jso
n', 'init=/tmp/tmpunk9xwl5/wlg6d5xb.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
jn51ge8b/prophet_model-20250519093655.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:55 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:56 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling KOREA tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/e0plyqhi.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/snk7395j.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=85597', 'data', 'file=/tmp/tmpunk9xwl5/e0plyqhi.jso
n', 'init=/tmp/tmpunk9xwl5/snk7395j.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
mb1ugw3f/prophet_model-20250519093657.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:36:57 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:57 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling KUWAIT tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/5inl89jb.json
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/c0ejewtu.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=95706', 'data', 'file=/tmp/tmpunk9xwl5/5inl89jb.json', 'init=/tmp/tmpunk9xwl5/c0ejewtu.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelf7msts7d/prophet_model-20250519093658.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:36:58 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:36:58 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling LAOS tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ypl5rjf7.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/zny82cds.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=85576', 'data', 'file=/tmp/tmpunk9xwl5/ypl5rjf7.json', 'init=/tmp/tmpunk9xwl5/zny82cds.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelodragtlp/prophet_model-20250519093659.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:36:59 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:00 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/42l7ilcc.json
```

```
Modeling LUXEMBOURG tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/xgy43y4_.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=85139', 'data', 'file=/tmp/tmpunk9xwl5/42l7ilcc.json', 'init=/tmp/tmpunk9xwl5/xgy43y4_.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model_6e6mr2l/prophet_model-20250519093701.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:37:01 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:01 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling MALAYSIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/n8pjk7cm.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/xw7v5083.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=85022', 'data', 'file=/tmp/tmpunk9xwl5/n8pjk7cm.json', 'init=/tmp/tmpunk9xwl5/xw7v5083.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelsqxl09in/prophet_model-20250519093702.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:37:02 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:03 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
```

```
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/k47jxbln.json
```

Modeling MEXICO tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/15tokxs7.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=19024', 'data', 'file=/tmp/tmpunk9xwl5/k47jxbln.jso
n', 'init=/tmp/tmpunk9xwl5/15tokxs7.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
vdgxf36e/prophet_model-20250519093704.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:04 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:04 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling MYANMAR tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/es6dgnsa.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/xphpo4fq.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=5306', 'data', 'file=/tmp/tmpunk9xwl5/es6dgnsa.json
', 'init=/tmp/tmpunk9xwl5/xphpo4fq.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modela
9vo6dgd/prophet_model-20250519093705.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10
000']
09:37:05 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:06 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling NEPAL tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/rspah42m.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/vwgvuuio.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=10670', 'data', 'file=/tmp/tmpunk9xwl5/rspah42m.jso
n', 'init=/tmp/tmpunk9xwl5/vwgvuuio.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
sle5728q/prophet_model-20250519093708.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:08 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:08 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/efp3hksu.json
```

Modeling NETHERLANDS tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/iu1hgvnv.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=73281', 'data', 'file=/tmp/tmpunk9xwl5/efp3hksu.jso
n', 'init=/tmp/tmpunk9xwl5/iu1hgvnv.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
70oklcre/prophet_model-20250519093709.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
```

```
0000']
09:37:09 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:10 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling NEW ZEALAND tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ekwpkaro.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/0wt4ns_e.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=25625', 'data', 'file=/tmp/tmpunk9xwl5/ekwpkaro.jso
n', 'init=/tmp/tmpunk9xwl5/0wt4ns_e.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
xwq0nwxc/prophet_model-20250519093711.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:11 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:11 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling NIGERIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/mc0obbdt.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/jw09_52l.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=60752', 'data', 'file=/tmp/tmpunk9xwl5/mc0obbdt.jso
n', 'init=/tmp/tmpunk9xwl5/jw09_52l.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
4cclbx27/prophet_model-20250519093712.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:12 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:13 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling NORWAY tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/ex8lmsjg.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/bw7kjtnk.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=8882', 'data', 'file=/tmp/tmpunk9xwl5/ex8lmsjg.json
', 'init=/tmp/tmpunk9xwl5/bw7kjtnk.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model9
rn4bynm/prophet_model-20250519093714.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10
000']
09:37:14 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:14 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling Others tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/_j0ezxf0.json
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/00ky0tax.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=53045', 'data', 'file=/tmp/tmpunk9xwl5/_j0ezxf0.json', 'init=/tmp/tmpunk9xwl5/00ky0tax.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modeldkh1ey7k/prophet_model-20250519093715.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:37:15 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:16 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling PAKISTAN tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/dxhh4iyp.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/2xgetiu1.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=63003', 'data', 'file=/tmp/tmpunk9xwl5/dxhh4iyp.json', 'init=/tmp/tmpunk9xwl5/2xgetiu1.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelrwb9nvn3/prophet_model-20250519093717.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:37:17 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:17 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/8o131nez.json
```

```
Modeling PAPUA NEW GUINEA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/xtdf01tb.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=69633', 'data', 'file=/tmp/tmpunk9xwl5/8o131nez.json', 'init=/tmp/tmpunk9xwl5/xtdf01tb.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelm995rmv5/prophet_model-20250519093719.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:37:19 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:19 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Modeling PERU tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/7txnoo9f.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/wwd950s6.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=38335', 'data', 'file=/tmp/tmpunk9xwl5/7txnoo9f.json', 'init=/tmp/tmpunk9xwl5/wwd950s6.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_modelup6li40f/prophet_model-20250519093720.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=10000']
09:37:21 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:21 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
```

```
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling POLAND tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/0vlmfee6.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/up6r2m0y.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=51431', 'data', 'file=/tmp/tmpunk9xwl5/0vlmfee6.jso
n', 'init=/tmp/tmpunk9xwl5/up6r2m0y.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
4icjvk8k/prophet_model-20250519093722.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:22 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:23 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling PORTUGAL tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/_488xjro.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/kc9hqt4n.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=96160', 'data', 'file=/tmp/tmpunk9xwl5/_488xjro.jso
n', 'init=/tmp/tmpunk9xwl5/kc9hqt4n.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
cakgbp0e/prophet_model-20250519093724.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:24 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:25 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling RUSSIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/aduhyq4s.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/3c2ifzmg.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=49877', 'data', 'file=/tmp/tmpunk9xwl5/aduhyq4s.jso
n', 'init=/tmp/tmpunk9xwl5/3c2ifzmg.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
p3arjpjg/prophet_model-20250519093726.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:26 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:26 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling SAUDI ARABIA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/y45m7839.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/fb23ojig.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=18366', 'data', 'file=/tmp/tmpunk9xwl5/y45m7839.jso
n', 'init=/tmp/tmpunk9xwl5/fb23ojig.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
q_cmtlkq/prophet_model-20250519093727.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
```

```
0000']
09:37:27 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:28 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling SINGAPORE tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/tw9lufqt.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/muxkprjo.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=40017', 'data', 'file=/tmp/tmpunk9xwl5/tw9lufqt.jso
n', 'init=/tmp/tmpunk9xwl5/muxkprjo.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
my0xmc7e/prophet_model-20250519093729.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:29 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:29 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/wjavw8bz.json
```

Modeling SOUTH AFRICA tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/rjeb_l2g.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=74055', 'data', 'file=/tmp/tmpunk9xwl5/wjavw8bz.jso
n', 'init=/tmp/tmpunk9xwl5/rjeb_l2g.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
5gmd293j/prophet_model-20250519093730.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:30 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:31 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling SPAIN tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/vjvf4zzn.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/kpgv2ucb.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=42547', 'data', 'file=/tmp/tmpunk9xwl5/vjvf4zzn.jso
n', 'init=/tmp/tmpunk9xwl5/kpgv2ucb.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
zr96gnjz/prophet_model-20250519093732.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:32 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:33 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling SRI LANKA tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/o6ej0p7c.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/p6cpwaak.json
```

```
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=27701', 'data', 'file=/tmp/tmpunk9xwl5/o6ej0p7c.jso
n', 'init=/tmp/tmpunk9xwl5/p6cpwaak.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
i7w4imtn/prophet_model-20250519093734.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:34 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:34 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling SWEDEN tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/pvf_krgg.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/be43_j9q.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=13951', 'data', 'file=/tmp/tmpunk9xwl5/pvf_krgg.jso
n', 'init=/tmp/tmpunk9xwl5/be43_j9q.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
rupy3jjd/prophet_model-20250519093736.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:36 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:36 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/qgqmn1rr.json
```

```
Modeling SWITZERLAND tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/bnjti4hm.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=16665', 'data', 'file=/tmp/tmpunk9xwl5/qgqmn1rr.jso
n', 'init=/tmp/tmpunk9xwl5/bnjti4hm.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
ahp97azn/prophet_model-20250519093737.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:37 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:38 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling TAIWAN tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/tnh7r4y7.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/a4rs2w46.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=19216', 'data', 'file=/tmp/tmpunk9xwl5/tnh7r4y7.jso
n', 'init=/tmp/tmpunk9xwl5/a4rs2w46.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
ccik2ftn/prophet_model-20250519093739.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:39 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:39 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
```

```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/vb5uiazd.json
```

Modeling THAILAND tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/d_vv5kxq.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=81304', 'data', 'file=/tmp/tmpunk9xwl5/vb5uiazd.jso
n', 'init=/tmp/tmpunk9xwl5/d_vv5kxq.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
3rx7u2z5/prophet_model-20250519093740.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:40 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:41 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/43v_0742.json
```

Modeling UNITED ARAB EMIRATES tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/g97fbyct.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=55181', 'data', 'file=/tmp/tmpunk9xwl5/43v_0742.jso
n', 'init=/tmp/tmpunk9xwl5/g97fbyct.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
16ig3zi8/prophet_model-20250519093742.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:42 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:42 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

Modeling UNITED KINGDOM tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/piqrphl4.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/6hc8wovu.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=65394', 'data', 'file=/tmp/tmpunk9xwl5/piqrphl4.jso
n', 'init=/tmp/tmpunk9xwl5/6hc8wovu.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
rnmj0anu/prophet_model-20250519093743.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:43 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:43 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/z552116r.json
```

Modeling USA tourist arrivals...

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/joeo70pi.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=77192', 'data', 'file=/tmp/tmpunk9xwl5/z552116r.jso
n', 'init=/tmp/tmpunk9xwl5/joeo70pi.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
u5pwfbu2/prophet_model-20250519093744.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
```

```
09:37:44 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:45 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling VENEZUELA tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/655rbtpt.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/tletd6pi.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=62398', 'data', 'file=/tmp/tmpunk9xwl5/655rbtpt.jso
n', 'init=/tmp/tmpunk9xwl5/tletd6pi.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
me8r7ne0/prophet_model-20250519093746.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:46 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:47 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to ov
erride this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to over
ride this.
```

```
Modeling VIETNAM tourist arrivals...
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/75fdrky4.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpunk9xwl5/iafxlj3y.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.11/dist-packages/prophet/stan_mode
l/prophet_model.bin', 'random', 'seed=56222', 'data', 'file=/tmp/tmpunk9xwl5/75fdrky4.jso
n', 'init=/tmp/tmpunk9xwl5/iafxlj3y.json', 'output', 'file=/tmp/tmpunk9xwl5/prophet_model
rpwa0k5o/prophet_model-20250519093748.csv', 'method=optimize', 'algorithm=lbfgs', 'iter=1
0000']
09:37:48 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
09:37:50 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
```

## Compare forecasts

In [ ]:

```
last_date = max([forecast['ds'].max() for forecast in country_forecasts.values()])

forecast_totals = {}
for country, forecast in country_forecasts.items():
    future_forecast = forecast[forecast['ds'] > (last_date - pd.Timedelta(days=months_to
_predict*30))]
    forecast_totals[country] = future_forecast['yhat'].sum()

forecast_df = pd.DataFrame(
    {'Total Forecasted Arrivals': forecast_totals}
).sort_values('Total Forecasted Arrivals', ascending=False)
forecast_df
```

Out[ ]:

| | Total Forecasted Arrivals |
|---|---|
| **KOREA** | 6429236.5497027198 |
| **USA** | 4599657.8814374721 |
| **CHINA** | 1556804.3524883299 |

| AUSTRALIA | Total Forecasted Arrivals |
|---|---|
| CANADA | 1198065.0618112839 |
| Others | 1130820.7526055875 |
| JAPAN | 991689.3458624858 |
| SINGAPORE | 789801.4408618143 |
| TAIWAN | 739440.6423494936 |
| UNITED KINGDOM | 627849.9821772992 |
| GERMANY | 349180.2828313585 |
| MALAYSIA | 328067.0111746844 |
| UNITED ARAB EMIRATES | 327891.4711485712 |
| FRANCE | 259923.8265465413 |
| GUAM | 246974.2925709430 |
| INDONESIA | 239673.5948953746 |
| SPAIN | 205825.7618496258 |
| THAILAND | 182387.2277126144 |
| NETHERLANDS | 181194.2125027147 |
| INDIA | 165449.7425099512 |
| ITALY | 164391.1813874570 |
| SWITZERLAND | 138458.6740711306 |
| POLAND | 115484.7610548342 |
| NEW ZEALAND | 107629.1223743102 |
| NORWAY | 93480.9256390962 |
| ISRAEL | 87956.3971539322 |
| AUSTRIA | 82256.4699975108 |
| RUSSIA | 71609.0426058577 |
| BELGIUM | 71557.6990603095 |
| IRELAND | 64302.3488742307 |
| SWEDEN | 59593.1884997445 |
| DENMARK | 59447.9404145526 |
| PAPUA NEW GUINEA | 44211.1774661618 |
| BAHRAIN | 41250.9902601642 |
| KUWAIT | 38988.7593868969 |
| BANGLADESH | 28074.9452602761 |
| BRUNEI | 21630.4280522546 |
| NIGERIA | 20671.5119373302 |
| SAUDI ARABIA | 19863.2662757666 |
| GREECE | 17249.1778803646 |
| PAKISTAN | 16710.6809113393 |
| MEXICO | 15371.3633410005 |
| FINLAND | 14811.0107763713 |
| PORTUGAL | 13974.8851584007 |
| NEPAL | 12910.5197485235 |
| CAMBODIA | 12584.7802227809 |
| ARGENTINA | 10294.6098232029 |
| VIETNAM | 9633.9733150339 |

| | |
|---|---|
| **SOUTH AFRICA** | **Total Forecasted Arrivals** |
| **BRAZIL** | 7912.1960846467 |
| **LUXEMBOURG** | 7067.7057725286 |
| **MYANMAR** | 5438.8382550391 |
| **COLOMBIA** | 4253.0681800427 |
| **LAOS** | 4044.2504106912 |
| **IRAN** | 3902.0532892831 |
| **EGYPT** | 3810.4668054313 |
| **PERU** | 3194.8280660571 |
| **SRI LANKA** | 3171.1626454736 |
| **VENEZUELA** | -222.2284730448 |
| **JORDAN** | -279.9530915016 |

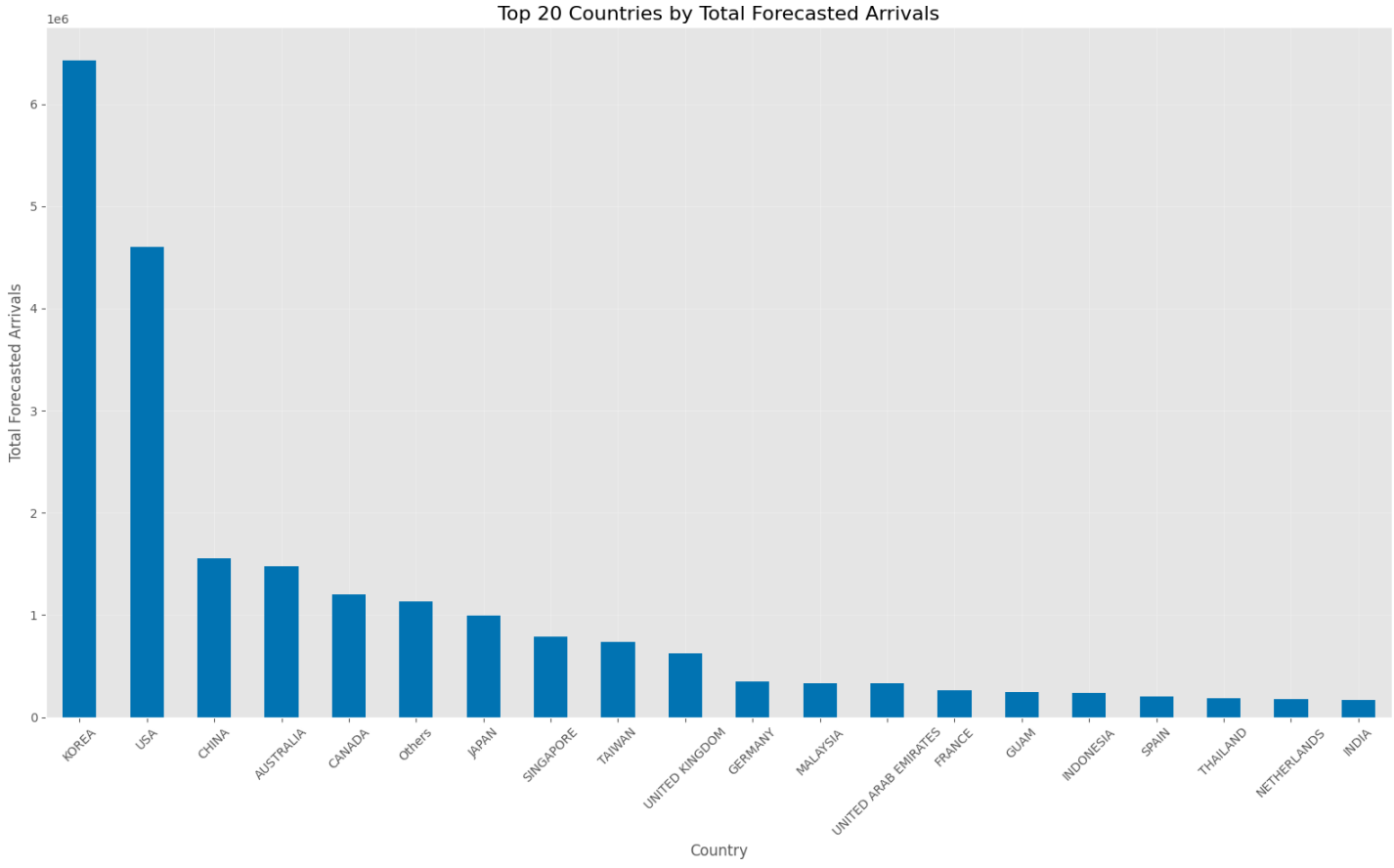
In [ ]:

```
top10_countries = forecast_df.head(10).index.tolist()
```

# Visualize the top 20 countries

In [ ]:

```
plt.figure(figsize=(16, 10))
forecast_df.head(20)['Total Forecasted Arrivals'].plot(kind='bar')
plt.title('Top 20 Countries by Total Forecasted Arrivals', fontsize=16)
plt.xlabel('Country', fontsize=12)
plt.ylabel('Total Forecasted Arrivals', fontsize=12)
plt.xticks(rotation=45)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig('country_forecast_totals.png')
plt.show()
```



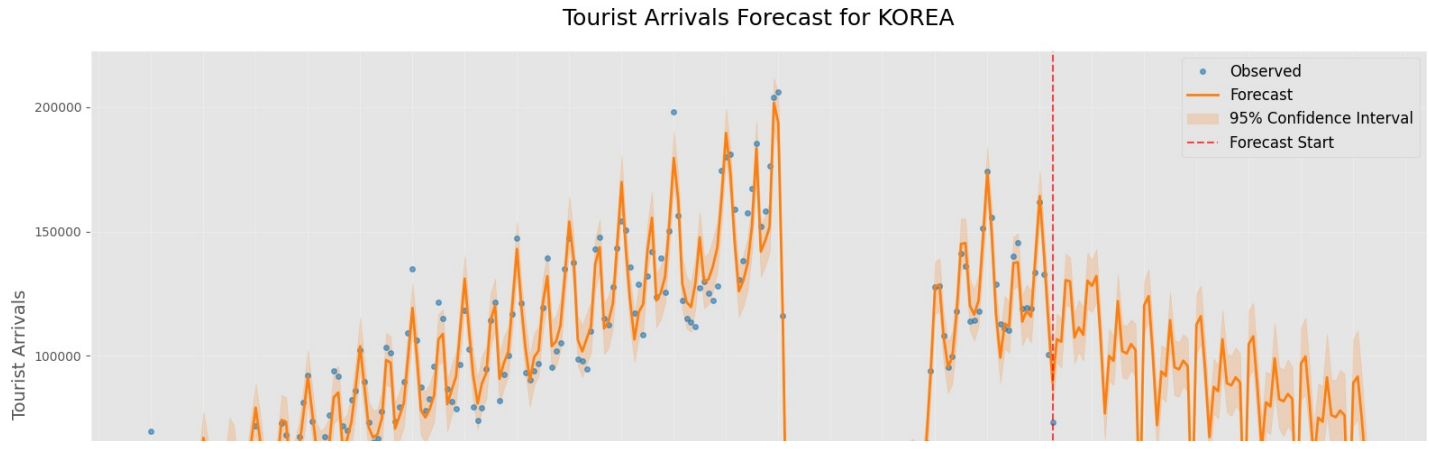
# Visualize top 10 country forecasts

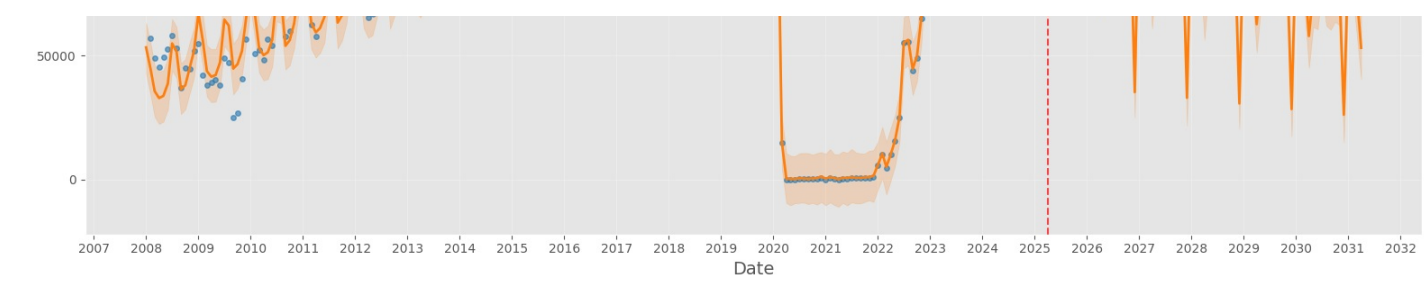
In [ ]:

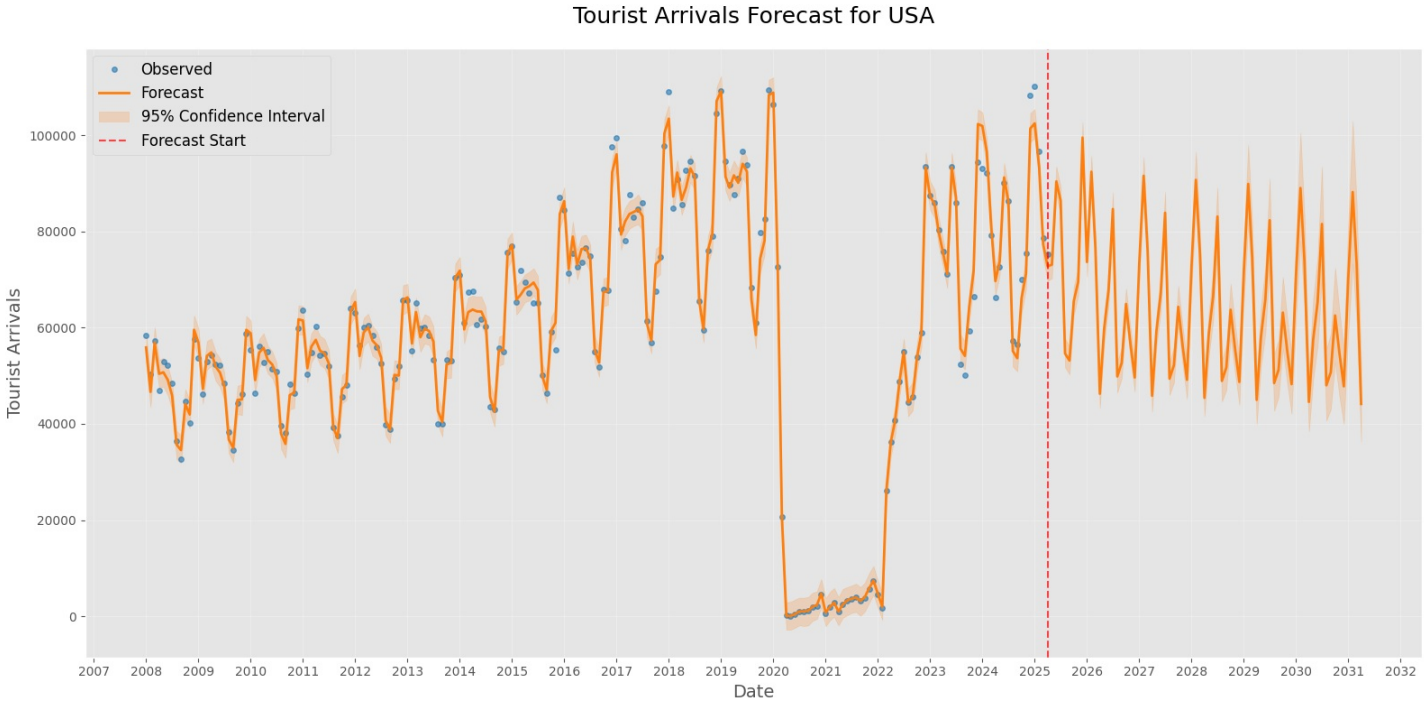
```
def create_better_plot(forecast, df, title, future_periods=0):
    plt.figure(figsize=(16, 8))

    # Plot actual data
    plt.plot(df['ds'], df['y'], 'o', color='#1f77b4', alpha=0.6, markersize=4, label='Observed')

    # Plot forecast
    plt.plot(forecast['ds'], forecast['yhat'], '-', color='#ff7f0e', linewidth=2, label='Forecast')

    # Plot confidence interval
    plt.fill_between(
        forecast['ds'],
        forecast['yhat_lower'],
        forecast['yhat_upper'],
        color='#ff7f0e',
        alpha=0.2,
        label='95% Confidence Interval'
    )

    if future_periods > 0:
        forecast_start = df['ds'].iloc[-1]
        plt.axvline(x=forecast_start, color='red', linestyle='--', alpha=0.7, label='Forecast Start')

    plt.title(title, fontsize=18, pad=20)
    plt.xlabel('Date', fontsize=14)
    plt.ylabel('Tourist Arrivals', fontsize=14)
    plt.legend(fontsize=12)
    plt.grid(True, alpha=0.3)

    plt.gca().xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y'))
    plt.gca().xaxis.set_major_locator(plt.matplotlib.dates.YearLocator())

    plt.tight_layout()
    return plt
```

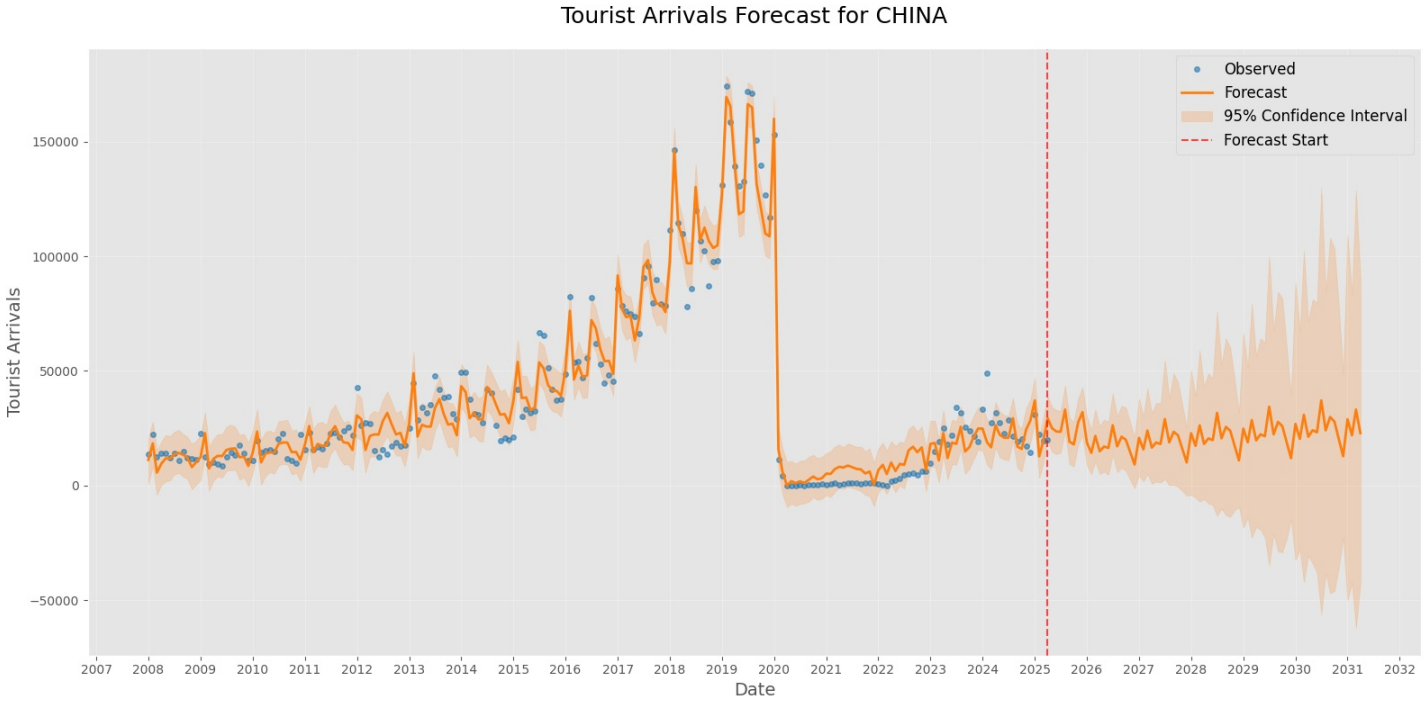
In [ ]:

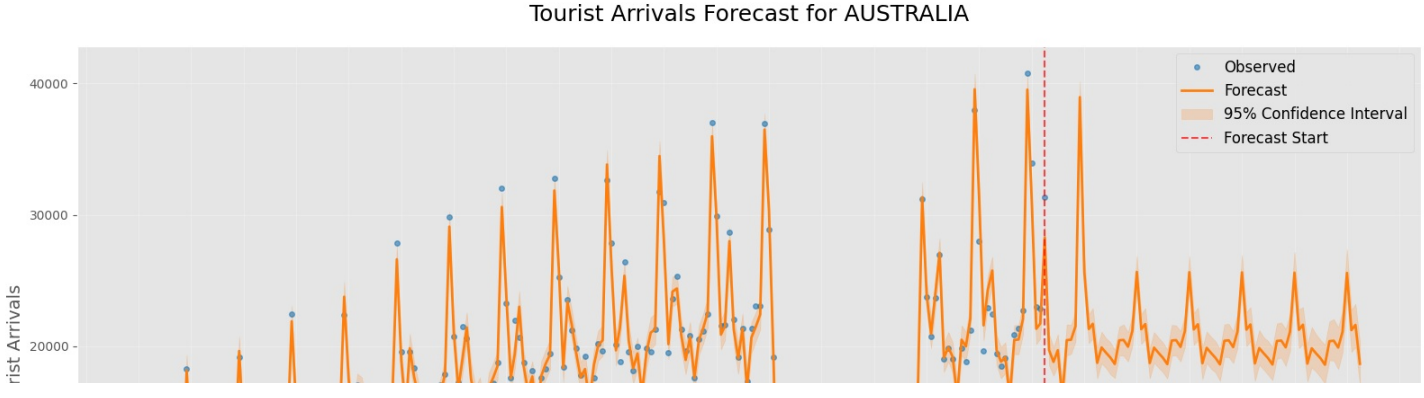
```
for country in top10_countries:
    forecast = country_forecasts[country]
    prophet_df = prepare_prophet_data(df_fixed, country)

    create_better_plot(
        forecast,
        prophet_df,
        f'Tourist Arrivals Forecast for {country}',
        future_periods=1
    ).show()

    print()
```

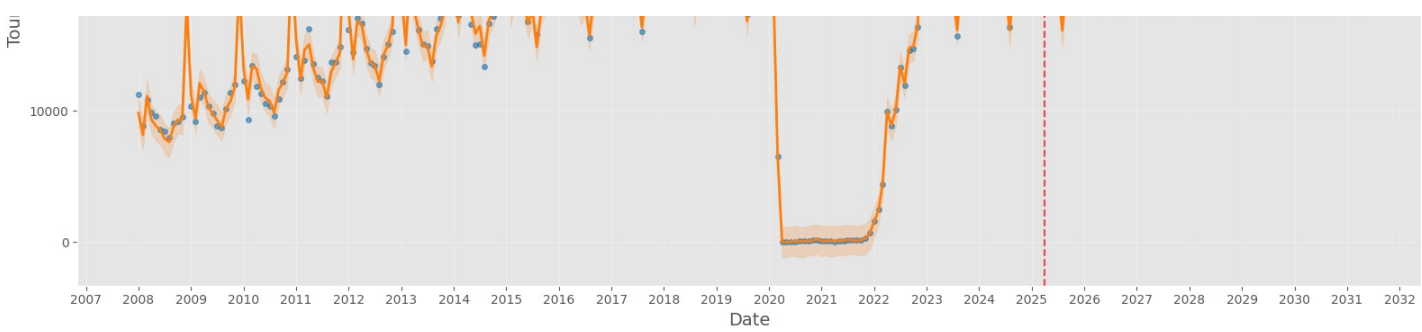


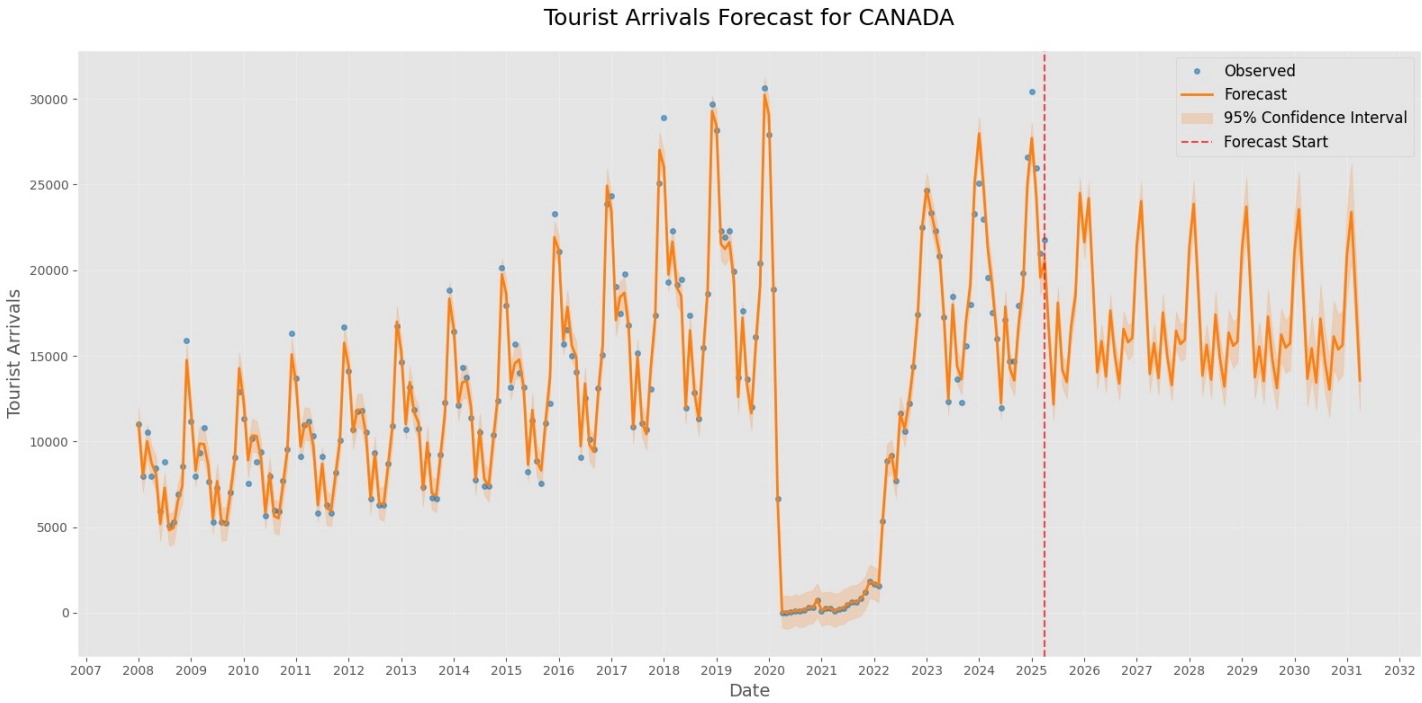

50000
0
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
Date

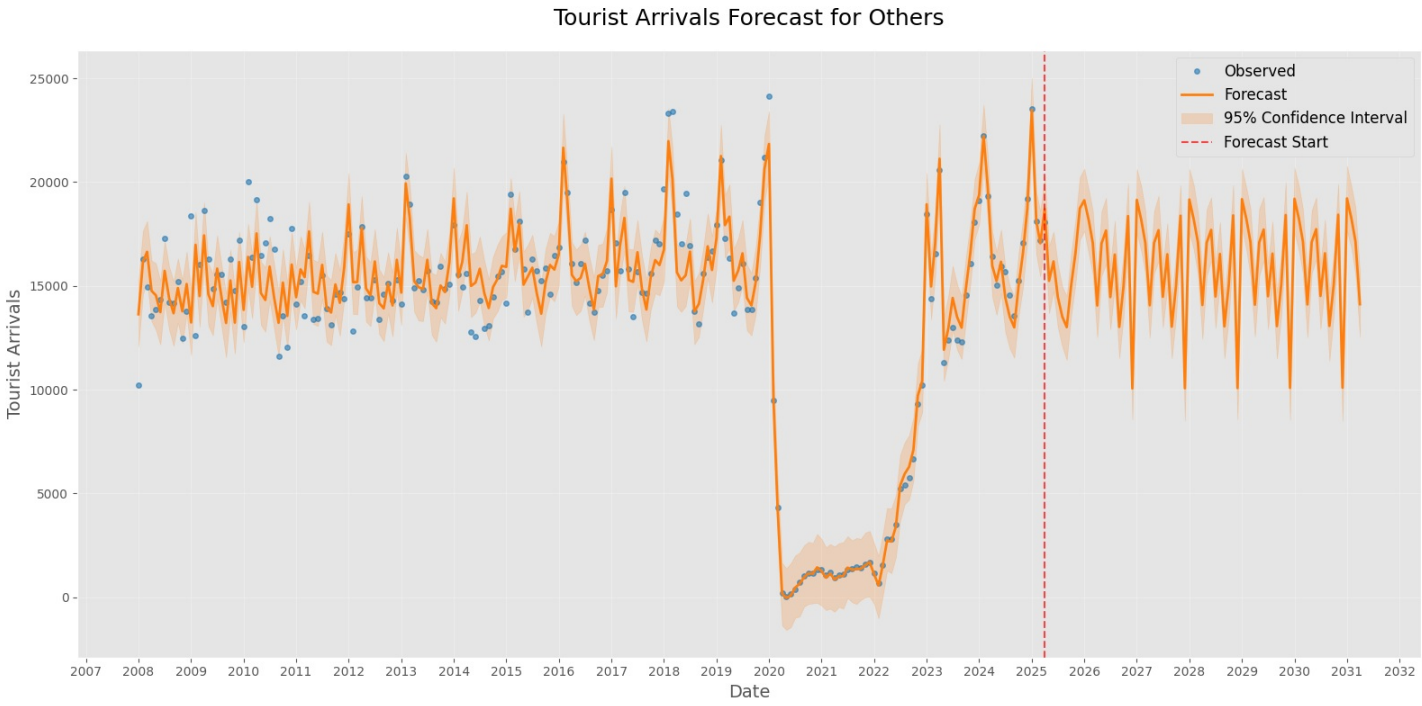

Tourist Arrivals Forecast for USA
Observed
Forecast
95% Confidence Interval
Forecast Start
Tourist Arrivals
100000
80000
60000
40000
20000
0
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
Date

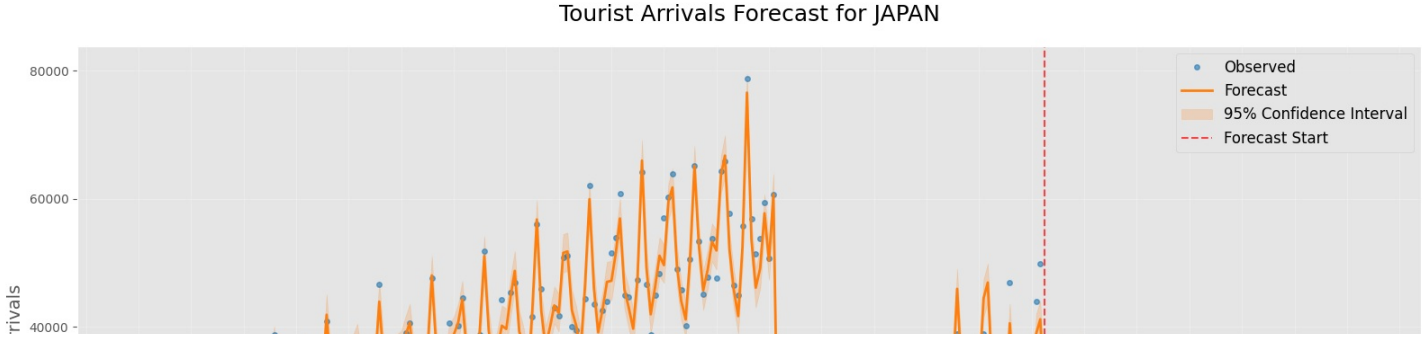

Tourist Arrivals Forecast for CHINA
Observed
Forecast
95% Confidence Interval
Forecast Start
Tourist Arrivals
150000
100000
50000
0
−50000
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
Date

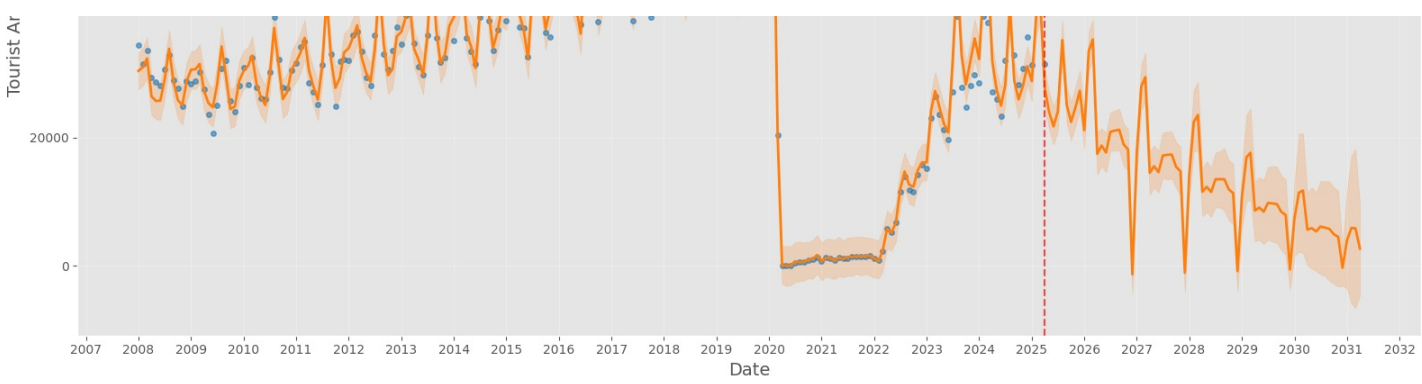

Tourist Arrivals Forecast for AUSTRALIA
Observed
Forecast
95% Confidence Interval
Forecast Start
40000
30000
20000


Tour
10000
0
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
Date

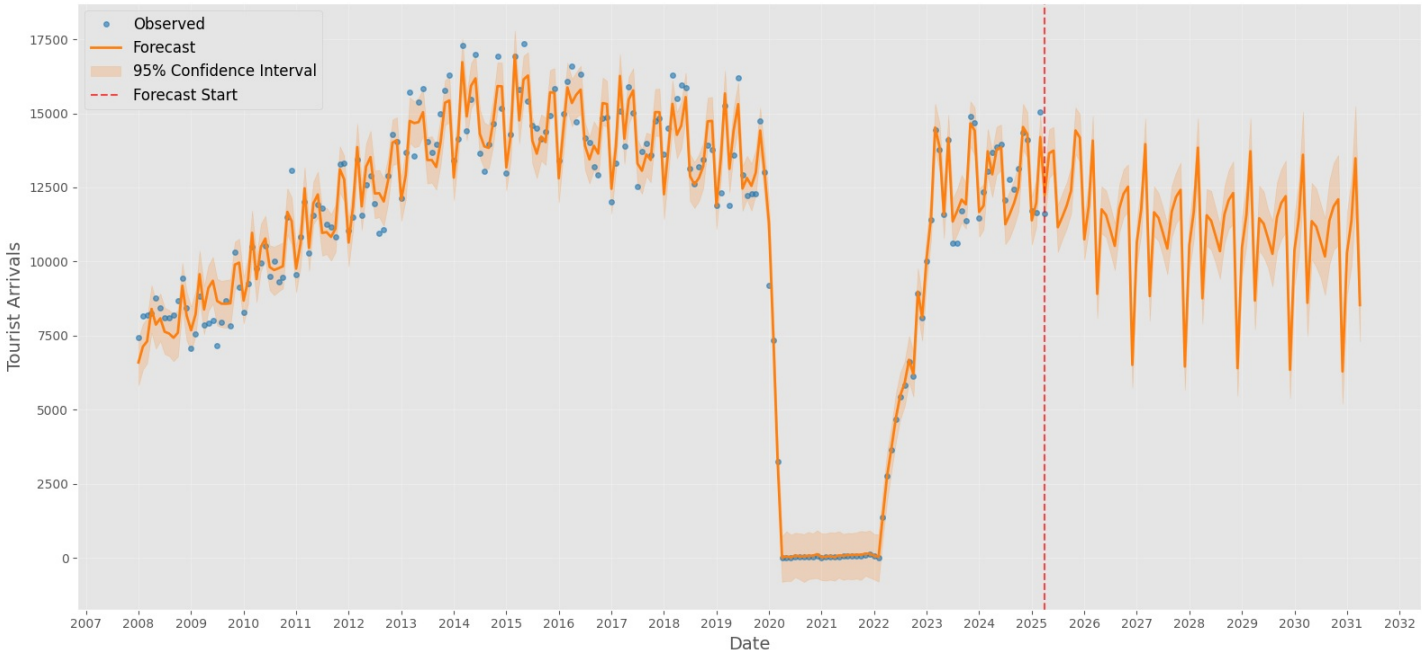

Tourist Arrivals Forecast for CANADA
Observed
Forecast
95% Confidence Interval
Forecast Start
Tourist Arrivals
30000
25000
20000
15000
10000
5000
0
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
Date


Tourist Arrivals Forecast for Others
Observed
Forecast
95% Confidence Interval
Forecast Start
Tourist Arrivals
25000
20000
15000
10000
5000
0
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032
Date


Tourist Arrivals Forecast for JAPAN
Observed
Forecast
95% Confidence Interval
Forecast Start
80000
60000
40000

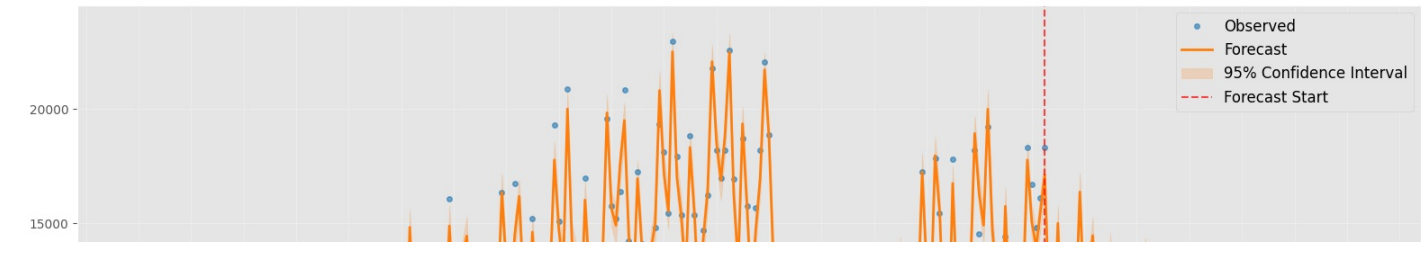


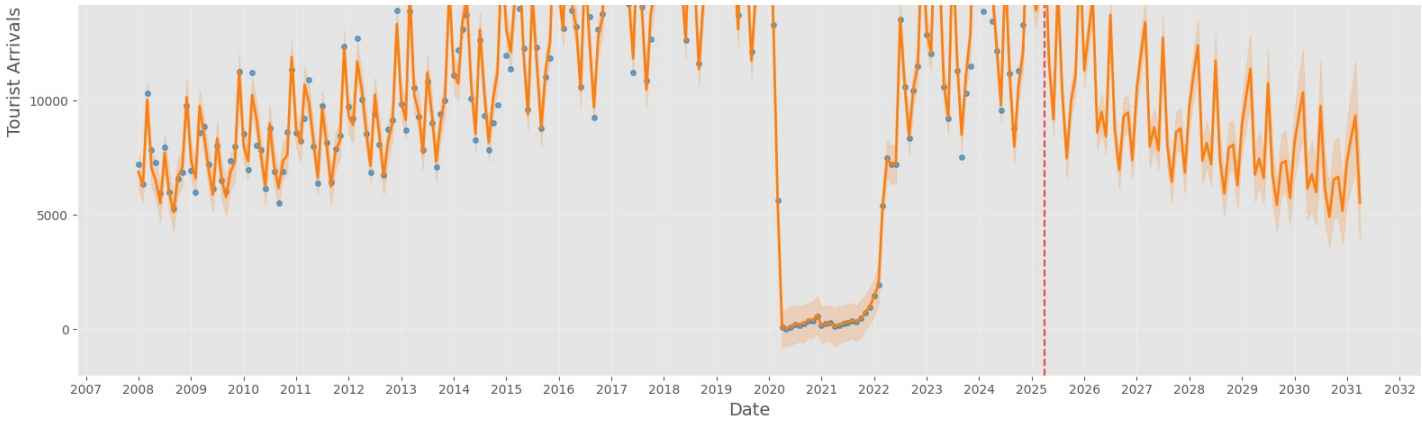

Tourist Arrivals Forecast for SINGAPORE




Tourist Arrivals Forecast for TAIWAN


Tourist Arrivals Forecast for UNITED KINGDOM






# Evaluation

## Regression report

In [ ]:

```
def regression_report(y_true, y_pred, normalize=True):
    y_true = np.array(y_true)
    y_pred = np.array(y_pred)

    # Metrics
    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_true, y_pred)
    mape = mean_absolute_percentage_error(y_true, y_pred) * 100

    # Normalization factor
    norm_factor = np.ptp(y_true) if normalize else 1.0  # ptp = max - min

    report = pd.DataFrame({
        'Metric': ['MAE', 'MSE', 'RMSE', 'MAPE'],
        'Value': [mae / norm_factor,
                  mse / (norm_factor ** 2),
                  rmse / norm_factor,
                  mape]
    })

    return report
```

In [ ]:

```
country_metrics = {}

for country in all_countries:
  model = country_models[country]
  forecast = country_forecasts[country]
  data = prepare_prophet_data(df_fixed, country)
  df_cv = regression_report(data['y'], forecast['yhat'][:-months_to_predict])

  country_metrics[country] = df_cv
```

In [ ]:

```
model_comparison = {}

for country, metrics in country_metrics.items():
    model_comparison[country] = metrics.loc[metrics['Metric'] == 'MAPE', 'Value'].values
[0]

comparison_df = pd.DataFrame(model_comparison.items(), columns=['Country', 'MAPE'])
comparison_df = comparison_df.set_index('Country')
top_20_countries = comparison_df.sort_values('MAPE', ascending=True).head(20)
top_20_countries
```

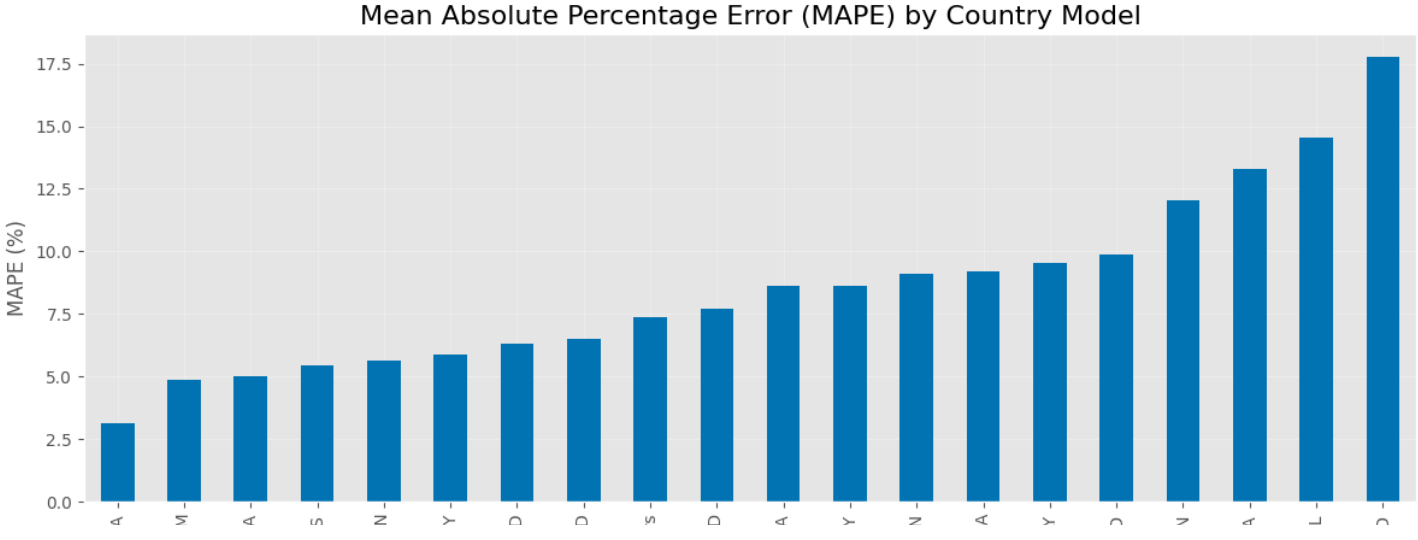
Out[ ]:

| Country | MAPE |
|---|---|
| USA | 3.1137469172 |
| UNITED KINGDOM | 4.8823332673 |
| CANADA | 5.0028082676 |
| NETHERLANDS | 5.4396488740 |
| JAPAN | 5.6380063392 |
| GERMANY | 5.8771139415 |
| SWITZERLAND | 6.3073856921 |
| NEW ZEALAND | 6.5216265400 |
| Others | 7.3716546573 |
| THAILAND | 7.7203101453 |
| AUSTRALIA | 8.6279426227 |
| NORWAY | 8.6376662874 |
| SPAIN | 9.0908690274 |
| SOUTH AFRICA | 9.1979822151 |
| ITALY | 9.5529584034 |
| IRELAND | 9.8525945281 |
| BAHRAIN | 12.0356144681 |
| RUSSIA | 13.3004063113 |
| BRAZIL | 14.5246163745 |
| POLAND | 17.7809927217 |

## Bar chart

In [ ]:

```
plt.figure(figsize=(12, 6))
(top_20_countries['MAPE']).sort_values().plot(kind='bar')
plt.title('Mean Absolute Percentage Error (MAPE) by Country Model', fontsize=16)
plt.xlabel('Country', fontsize=12)
plt.ylabel('MAPE (%)', fontsize=12)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig('model_comparison_mape.png')
plt.show()
```



US
UNITED KINGDOM
CANADA
NETHERLANDS
JAPAN
GERMANY
SWITZERLAND
NEW ZEALAND
Other
THAILAND
AUSTRALIA
NORWAY
SPAIN
SOUTH AFRICA
ITALY
IRELAND
BAHRAIN
RUSSIA
BRAZIL
POLAND
Country