```python
# 1. Import libraries, download and prepare stock data
import yfinance as yf
import pandas as pd
import numpy as np
from yahoo_fin import stock_info as si
from datetime import datetime, timedelta
from scipy.optimize import minimize
from itertools import product
import dash_bootstrap_components as dbc
from dash import html, dcc, dash_table
from pmdarima import auto_arima
from statsmodels.tsa.stattools import adfuller
import matplotlib.pyplot as plt
import plotly.graph_objs as go
import dash
from dash.dependencies import Input, Output, State, ALL
import requests
from bs4 import BeautifulSoup
from dash_bootstrap_templates import load_figure_template
import json
from io import StringIO
from dash.exceptions import PreventUpdate


combined_data = None


# Function to download data, check if it's older than 1 year, and fill NaN values
def download_data_fillna(tickers, start_date, end_date):
    valid_tickers = []
    young_tickers = []
    invalid_tickers = []

    for ticker in tickers:
        try:
            # Download data
            data = yf.download(tickers=[ticker], start=start_date, end=end_date)['Close']
            print(f"Downloaded data for {ticker}: {data}")

            # Check if data is empty
            if data.empty:
                print(f"{ticker} has no data, categorizing as invalid.")
                invalid_tickers.append(ticker)
                continue  # Move to the next ticker

            # Check if the ticker is young (doesn't exist on or before 2023-01-03)
            if data.index[0] > pd.to_datetime("2023-01-03"):
                print(f"{ticker} is young, categorizing as young.")
                young_tickers.append(ticker)
            else:
                # Ensure data is a DataFrame and rename the column to ticker name
                data = pd.DataFrame(data.rename(ticker))
                valid_tickers.append(data)
                print(f"{ticker} is valid and added to the list of valid tickers.")

        except Exception as e:
            print(f"Error downloading data for {ticker}: {e}")
            invalid_tickers.append(ticker)

    if valid_tickers:
        # Concatenate valid data into a single DataFrame
        all_data = pd.concat(valid_tickers, axis=1)
        all_data = all_data.asfreq('B')  # Ensure data is business day frequency

        # Fill NaN values with previous day's value
        data = all_data.ffill()
```

```
        return data, young_tickers, invalid_tickers
    else:
        return pd.DataFrame(), young_tickers, invalid_tickers
```