In [1]:
```python
# Python
import itertools
import numpy as np
import pandas as pd
import pandas as pd
import numpy as np
from prophet import Prophet
from prophet.diagnostics import cross_validation
from prophet.diagnostics import performance_metrics
import matplotlib.pyplot as plt
from prophet.plot import plot_cross_validation_metric
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error,
import funciones
```

In [2]:
```python
df_main = pd.read_excel("https://raw.githubusercontent.com/carrenogf/MCD-Series-
df_main = df_main.sort_values("FECHA",ascending=True)
df_main.set_index("FECHA", inplace=True)
df_copa = df_main["CHU_COPA_AJUST"].dropna()
df_recprop = df_main["CHU_REC_PROPIOS_AJUST"].dropna()
df_regal = df_main["CHU_REGALIAS_AJUST"].dropna()
dataframes = [df_copa, df_recprop, df_regal]
for i in range(len(dataframes)):
  dataframes[i] = dataframes[i].reindex(pd.date_range(start=dataframes[i].index.
  dataframes[i] = dataframes[i].fillna(0)

titulos = ["CHU_COPA_AJUST", "CHU_REC_PROPIOS_AJUST", "CHU_REGALIAS_AJUST"]
```

In [3]:
```python
# TRAIN TEST
n_train = 0.9
train_copa = dataframes[0].iloc[:round(len(dataframes[0])*n_train)]
test_copa = dataframes[0].iloc[round(len(dataframes[0])*n_train):]
print(f"Coparticipacion: train({train_copa.shape}), test({test_copa.shape})")

train_recursos = dataframes[1].iloc[:round(len(dataframes[1])*n_train)]
test_recursos = dataframes[1].iloc[round(len(dataframes[1])*n_train):]
print(f"Recursos: train({train_recursos.shape}), test({test_recursos.shape})")

train_regalias = dataframes[2].iloc[:round(len(dataframes[2])*n_train)]
test_regalias = dataframes[2].iloc[round(len(dataframes[2])*n_train):]
print(f"Regalias: train({train_regalias.shape}), test({test_regalias.shape})")

dataframes_train = [ train_copa, train_recursos, train_regalias ]
dataframes_test = [ test_copa, test_recursos, test_regalias ]
```

```
Coparticipacion: train((1584,)), test((176,))
Recursos: train((1995,)), test((222,))
Regalias: train((1985,)), test((221,))
```

In [ ]:
```python
from prophet.diagnostics import cross_validation, performance_metrics

fourier_y_list = []
for df in dataframes_train:
    name = df.name
    df = df.to_frame()
    df = df.reset_index()
    df.columns = ['ds', 'y']
    for fourier in [3, 5, 7, 10]:
        model = Prophet(yearly_seasonality=False)
```

```
        model.add_seasonality(name='yearly', period=365.25, fourier_order=fourie
        model.fit(df)

        df_cv = cross_validation(model, initial='730 days', period='180 days', h
        df_p = performance_metrics(df_cv)
        print(f'Fourier Order: {fourier}, mae: {df_p["mae"].mean()}')
        fourier_y_list.append({
            'name': name,
            'fourier_order': fourier,
            'mae': df_p["mae"].mean(),
            'mse': df_p["mse"].mean(),
            'rmse': df_p["rmse"].mean(),

        })
```

In [ ]:
```
from prophet.diagnostics import cross_validation, performance_metrics
fourier_m_list = []
for df in dataframes_train:
    name = df.name
    df = df.to_frame()
    df = df.reset_index()
    df.columns = ['ds', 'y']
    for fourier in [3, 5, 7, 10]:
        model = Prophet(yearly_seasonality=False)
        model.add_seasonality(name='monthly', period=30.5, fourier_order=fourier
        model.fit(df)

        df_cv = cross_validation(model, initial='730 days', period='180 days', h
        df_p = performance_metrics(df_cv)
        print(f'Fourier Order: {fourier}, mae: {df_p["mae"].mean()}')
        fourier_m_list.append({
            'name': name,
            'fourier_order': fourier,
            'mae': df_p["mae"].mean(),
            'mse': df_p["mse"].mean(),
            'rmse': df_p["rmse"].mean(),

        })
```

In [24]:
```
pd.DataFrame(fourier_m_list)
```

Out[24]:

	name	fourier_order	mae	mse	rm
0	CHU_COPA_AJUST	3	1,295,550.58	3,368,417,717,852.50	1,829,278.
1	CHU_COPA_AJUST	5	1,292,136.70	3,382,102,160,158.99	1,833,572.
2	CHU_COPA_AJUST	7	1,289,267.78	3,403,619,136,925.37	1,839,180.
3	CHU_COPA_AJUST	10	1,292,343.45	3,435,217,624,484.17	1,847,424.
4	CHU_REC_PROPIOS_AJUST	3	875,832.51	1,354,696,658,529.47	1,160,714.
5	CHU_REC_PROPIOS_AJUST	5	876,294.82	1,360,751,514,817.23	1,163,577.
6	CHU_REC_PROPIOS_AJUST	7	884,354.91	1,384,814,129,416.05	1,173,991.
7	CHU_REC_PROPIOS_AJUST	10	886,719.39	1,389,622,924,478.05	1,175,983.
8	CHU_REGALIAS_AJUST	3	2,981,291.15	33,617,382,219,021.43	5,786,977.
9	CHU_REGALIAS_AJUST	5	2,881,989.42	34,018,833,939,631.12	5,820,355.
10	CHU_REGALIAS_AJUST	7	2,887,115.68	34,332,503,235,722.89	5,847,218.
11	CHU_REGALIAS_AJUST	10	2,896,604.65	34,623,962,264,283.60	5,872,018.

In [28]:
```
pd.options.display.float_format = '{:,.2f}'.format
pd.DataFrame(fourier_y_list)
```

Out[28]:

	name	fourier_order	mae	mse	rm
0	CHU_COPA_AJUST	3	1,659,037.26	4,493,518,123,189.03	2,114,628.
1	CHU_COPA_AJUST	5	1,659,235.07	4,492,317,272,786.19	2,114,568.
2	CHU_COPA_AJUST	7	1,662,586.82	4,503,776,502,703.05	2,117,144.
3	CHU_COPA_AJUST	10	1,662,410.64	4,524,420,159,270.77	2,120,638.
4	CHU_REC_PROPIOS_AJUST	3	1,007,027.71	1,554,772,294,789.62	1,244,465.
5	CHU_REC_PROPIOS_AJUST	5	1,006,999.60	1,550,073,308,813.71	1,242,539.
6	CHU_REC_PROPIOS_AJUST	7	1,005,906.47	1,551,414,785,218.95	1,242,870.
7	CHU_REC_PROPIOS_AJUST	10	1,004,277.36	1,548,804,571,628.49	1,241,865.
8	CHU_REGALIAS_AJUST	3	2,975,730.22	36,804,980,074,407.97	6,054,561.
9	CHU_REGALIAS_AJUST	5	2,981,596.93	36,810,240,630,045.30	6,054,990.
10	CHU_REGALIAS_AJUST	7	2,961,931.77	36,818,174,730,371.95	6,055,578.
11	CHU_REGALIAS_AJUST	10	2,970,283.20	36,865,017,859,487.95	6,059,508.

In [35]:
```
def get_best_fourier_orders(fourier_list):
  best_orders = {}
  for item in fourier_list:
    name = item['name']
    if name not in best_orders or item['rmse'] < best_orders[name]['rmse']:
      best_orders[name] = {
```

```
        'fourier_order': item['fourier_order'],
        'rmse': item['rmse']
      }
  return best_orders

best_fourier_orders_y = get_best_fourier_orders(fourier_y_list)
best_fourier_orders_m = get_best_fourier_orders(fourier_m_list)

print("Best Fourier Orders (Yearly):", best_fourier_orders_y)
print("Best Fourier Orders (Monthly):", best_fourier_orders_m)
result = pd.DataFrame([best_fourier_orders_y, best_fourier_orders_m]).T
result.columns = ['Fourier_yearly', 'Fourier_monthly']
result
```

```
Best Fourier Orders (Yearly): {'CHU_COPA_AJUST': {'fourier_order': 5, 'rmse': 211
4568.5994672766}, 'CHU_REC_PROPIOS_AJUST': {'fourier_order': 10, 'rmse': 1241865.
2497155124}, 'CHU_REGALIAS_AJUST': {'fourier_order': 3, 'rmse': 6054561.27930790
9}}
Best Fourier Orders (Monthly): {'CHU_COPA_AJUST': {'fourier_order': 3, 'rmse': 18
29278.0979960766}, 'CHU_REC_PROPIOS_AJUST': {'fourier_order': 3, 'rmse': 1160714.
3127307887}, 'CHU_REGALIAS_AJUST': {'fourier_order': 3, 'rmse': 5786977.88651003
3}}
```

Out[35]:

	Fourier_yearly	Fourier_monthly
CHU_COPA_AJUST	{'fourier_order': 5, 'rmse': 2114568.5994672766}	{'fourier_order': 3, 'rmse': 1829278.0979960766}
CHU_REC_PROPIOS_AJUST	{'fourier_order': 10, 'rmse': 1241865.2497155124}	{'fourier_order': 3, 'rmse': 1160714.3127307887}
CHU_REGALIAS_AJUST	{'fourier_order': 3, 'rmse': 6054561.279307909}	{'fourier_order': 3, 'rmse': 5786977.886510033}

In [36]:
```
result.to_csv("best_fourier_orders.csv")
```

In [ ]:
```

```