```
from google.colab import drive
#drive.mount('/content/drive')
```

```
import pandas as pd
pd.options.display.max_columns = None
url = "https://drive.google.com/file/d/1-uCH-bTSSOaBkpk3pRYXfHWFzd7r6PG6/view?usp=sharing"
file_id = url.split('/')[-2]
dwn_url = "https://drive.google.com/uc?id=" + file_id
df = pd.read_csv(dwn_url)
```

```
df.head(10)
```

	ubigeo	renipress	diagnostic	ano	semana	tipo_dx	edad	tipo_edad	sexo	fecha_ini
0	21803	00001675	A97.0	2013	1	C	42	A	F	2013-01-02
1	21803	00001675	A97.0	2013	1	C	66	A	F	2013-01-01
2	21803	00001675	A97.0	2013	1	C	13	A	M	2013-01-02
3	21803	00001675	A97.0	2013	1	C	54	A	F	2013-01-03
4	21803	00009047	A97.0	2013	1	C	28	A	F	2013-01-05
5	21803	00001675	A97.0	2013	1	C	33	A	F	2013-01-01
6	21803	00001675	A97.0	2013	1	C	31	A	F	2013-01-01
7	21801	00001703	A97.1	2013	1	C	20	A	F	2013-01-05
8	21803	00001675	A97.0	2013	1	C	14	A	F	2012-12-31
9	21803	00001675	A97.0	2013	1	C	7	A	M	2012-12-31

```
df['diagnostic'].value_counts()
```

```
A97.0    524051
A97.1     76635
A97.2      2501
Name: diagnostic, dtype: int64
```

```
df_agrupado = df.groupby('fecha_ini', as_index=False).size()
df_agrupado
```

	fecha_ini	size
0	0000-00-00	9
1	2012-12-30	23
2	2012-12-31	31
3	2013-01-01	67
4	2013-01-02	55
...	...	...
3944	2023-10-17	349
3945	2023-10-18	313
3946	2023-10-19	248
3947	2023-10-20	241
3948	2023-10-21	149

3949 rows × 2 columns

```
df['tipo_dx'].value_counts()
```

```
C    489978
P    113209
Name: tipo_dx, dtype: int64
```

```
df = df[df['fecha_ini'] != '0000-00-00']
```

```
df['fecha_ini'] = pd.to_datetime(df['fecha_ini'])
```

```
df_agrupado = df.groupby('fecha_ini', as_index=False).size()
df_agrupado
```

	fecha_ini	size
0	2012-12-30	23
1	2012-12-31	31
2	2013-01-01	67
3	2013-01-02	55
4	2013-01-03	40
...	...	...
3943	2023-10-17	349
3944	2023-10-18	313
3945	2023-10-19	248
3946	2023-10-20	241
3947	2023-10-21	149

3948 rows × 2 columns

```
df_conf = df[df['tipo_dx'] == 'C']
```

```
df_agrupado = df_conf.groupby('fecha_ini', as_index=False).size()
df_agrupado
```

	fecha_ini	size
0	2012-12-30	23
1	2012-12-31	31
2	2013-01-01	67
3	2013-01-02	55
4	2013-01-03	40
...	...	...
3941	2023-10-17	102
3942	2023-10-18	73
3943	2023-10-19	57
3944	2023-10-20	70
3945	2023-10-21	23

3946 rows × 2 columns

```
df_agrupado.plot.scatter(x='fecha_ini', y = 'size')
```

```
<Axes: xlabel='fecha_ini', ylabel='size'>
```

```
from prophet import Prophet
df_pred = df_agrupado.head(3916)
df_test = df_agrupado.tail(30)

df_pred = df_pred.rename(columns={"fecha_ini": "ds", "size": "y"})
```

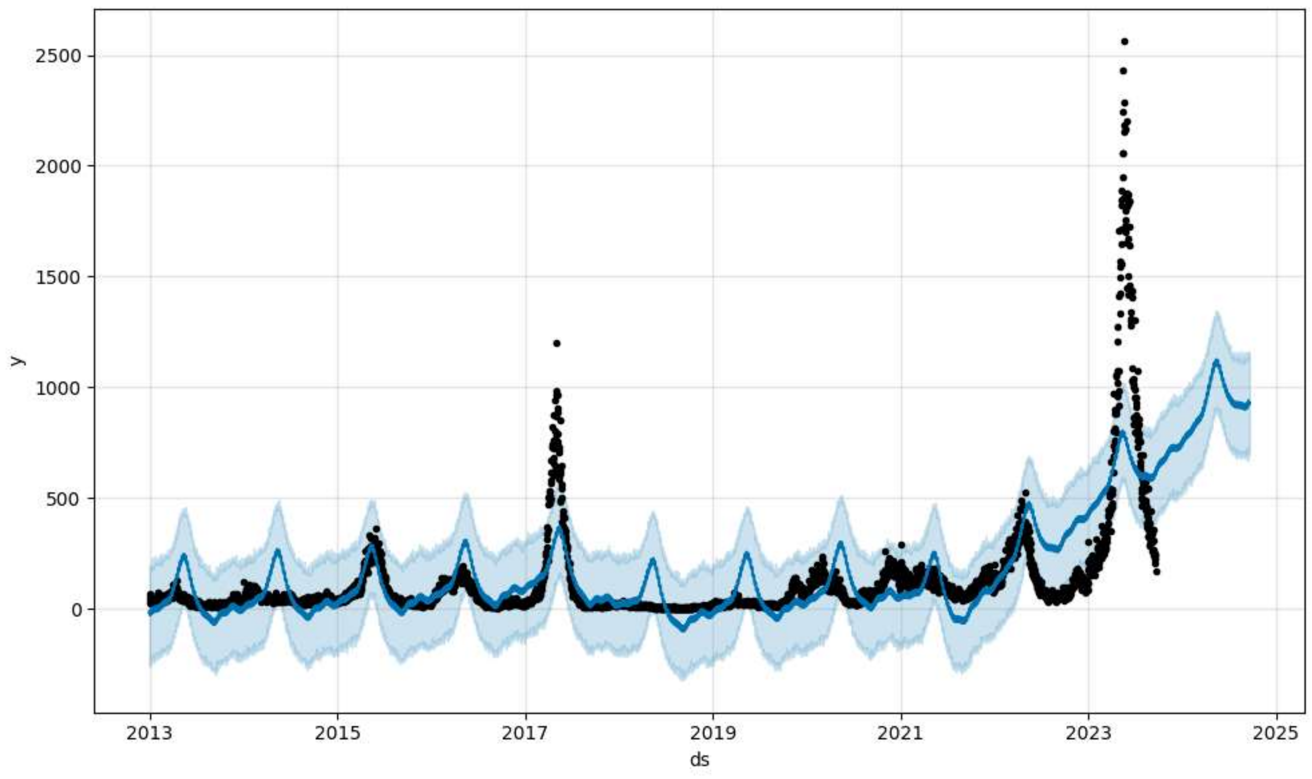
```
m = Prophet()
m.fit(df_pred)
future = m.make_future_dataframe(periods=365)
forecast = m.predict(future)
```

```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmp5m3ik_nn/jlhp9l7b.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmp5m3ik_nn/a7171vme.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.10/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=91420', '
16:42:22 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
16:42:22 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
```

```
forecast.tail(30)
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	
4251	2024-08-22	994.966893	681.972828	1107.778231	943.325130	1052.137332	-90.810531	-90.810531	-90.810531	·
4252	2024-08-23	995.849402	692.406541	1133.794107	943.977403	1053.235140	-89.305544	-89.305544	-89.305544	·
4253	2024-08-24	996.731911	711.564871	1134.683462	944.420876	1054.332948	-84.439351	-84.439351	-84.439351	
4254	2024-08-25	997.614420	694.612348	1149.719576	945.161358	1055.461301	-78.860608	-78.860608	-78.860608	
4255	2024-08-26	998.496929	703.796959	1151.501551	945.753335	1056.545360	-72.518619	-72.518619	-72.518619	1
4256	2024-08-27	999.379438	699.088665	1123.445104	946.345312	1057.644055	-91.659203	-91.659203	-91.659203	·
4257	2024-08-28	1000.261947	680.740400	1106.740592	946.996129	1058.753392	-98.636796	-98.636796	-98.636796	·
4258	2024-08-29	1001.144456	683.506889	1128.695303	947.696360	1059.869351	-100.594900	-100.594900	-100.594900	·
4259	2024-08-30	1002.026965	680.747767	1129.207093	948.396592	1060.985309	-98.946766	-98.946766	-98.946766	·
4260	2024-08-31	1002.909474	697.074801	1135.725654	949.096824	1062.101268	-93.839989	-93.839989	-93.839989	
4261	2024-09-01	1003.791983	710.424250	1140.690330	949.641439	1063.217226	-87.917846	-87.917846	-87.917846	
4262	2024-09-02	1004.674492	710.530151	1157.115174	950.171031	1064.333184	-81.125946	-81.125946	-81.125946	1
4263	2024-09-03	1005.557001	681.429617	1117.962428	950.700623	1065.445558	-99.708246	-99.708246	-99.708246	·
4264	2024-09-04	1006.439511	687.692281	1121.887127	951.230215	1066.695767	-106.019283	-106.019283	-106.019283	·

```
fig1 = m.plot(forecast)
```



09-18

```
df_test
```

	fecha_ini	size
3916	2023-09-22	209
3917	2023-09-23	164
3918	2023-09-24	188
3919	2023-09-25	237
3920	2023-09-26	155
3921	2023-09-27	131
3922	2023-09-28	187
3923	2023-09-29	168
3924	2023-09-30	180
3925	2023-10-01	233
3926	2023-10-02	193
3927	2023-10-03	157
3928	2023-10-04	159
3929	2023-10-05	136
3930	2023-10-06	159
3931	2023-10-07	118
3932	2023-10-08	156
3933	2023-10-09	148
3934	2023-10-10	160
3935	2023-10-11	104
3936	2023-10-12	105
3937	2023-10-13	104
3938	2023-10-14	110
3939	2023-10-15	91
3940	2023-10-16	117
3941	2023-10-17	102
3942	2023-10-18	73
3943	2023-10-19	57
3944	2023-10-20	70
3945	2023-10-21	23

Agrupamiento por lugar de establecimiento de salud

```
pd.options.display.max_columns = None
url = "https://drive.google.com/file/d/1UDdVebcfZdIlq8-b_WVPs9Jp09VyrmRw/view?usp=sharing"
file_id = url.split('/')[-2]
dwn_url = "https://drive.google.com/uc?id=" + file_id
df_tabla = pd.read_csv(dwn_url)
df_tabla
```

	id_eess	codigo_renaes	categoria	nombre	diresa	red	direccion	longitud	latitud	id_ubigeo
0	1	1	II-2	HOSPITAL IQUITOS "CESAR GARAYAR GARCIA"	LORETO	NO PERTENECE A NINGUNA RED	CALLE CORNEJO PORTUGAL 1710 IQUITOS MAYNAS LORETO	-73.253653	-3.762818	1453
1	2	2	SD	CENTRO REHABILITACION ENFERMO MENTAL	LORETO	NO PERTENECE A NINGUNA RED	CALLE 3 DE MAYO S/N CPM QUILCATACTA	NaN	NaN	1463
2	3	3	III-1	HOSPITAL REGIONAL DE LORETO "FELIPE SANTIAGO A...	LORETO	NO PERTENECE A NINGUNA RED	AVENIDA AVENIDA 28 DE JULIO S/N S/N AVENIDA 28...	-73.253417	-3.726960	1460
				C.S. I-3 SANTA		MAYNAS	OTROS CASERIO DE SANTA MARIA			

```
df_tabla.isnull().sum()
```

```
id_eess             0
codigo_renaes       0
categoria           0
nombre              0
diresa              0
red                 0
direccion         167
longitud        13597
latitud         13597
id_ubigeo           0
dtype: int64
```

CENTRO DE SALUD NO

```
provincia = df_tabla['diresa'].value_counts().index.to_list()
provincia
```

```
['LIMA DIRIS CENTRO',
 'LIMA DIRIS NORTE',
 'LIMA DIRIS SUR',
 'LIMA DIRIS ESTE',
 'PIURA',
 'CAJAMARCA',
 'AREQUIPA',
 'CUSCO',
 'CALLAO',
 'JUNIN',
 'LIMA PROVINCIAS',
 'LORETO',
 'LA LIBERTAD',
 'LAMBAYEQUE',
 'PUNO',
 'SAN MARTIN',
 'AMAZONAS',
 'ANCASH',
 'AYACUCHO',
 'TACNA',
 'APURIMAC',
 'ICA',
 'HUANUCO',
 'HUANCAVELICA',
 'PASCO',
 'UCAYALI',
 'MADRE DE DIOS',
 'MOQUEGUA',
 'TUMBES']
```

```
d = {name: pd.DataFrame() for name in provincia}
for name, df_2 in d.items(): d[name] = df_tabla[df_tabla['diresa'] == name]
```

```
id = d['LIMA DIRIS ESTE']['id_eess'].value_counts().index.to_list()
```

```
df['renipress'] = df['renipress'].astype('str').str.extractall('(\d+)').unstack().fillna('').sum(axis=1).astype(int)
#df['renipress'] = df['renipress'].str.lstrip('0').replace('.', '')
df
```

	ubigeo	renipress	diagnostic	ano	semana	tipo_dx	edad	tipo_edad	sexo	fecha_ini
0	21803	1675.0	A97.0	2013	1	C	42	A	F	2013-01-02
1	21803	1675.0	A97.0	2013	1	C	66	A	F	2013-01-01
2	21803	1675.0	A97.0	2013	1	C	13	A	M	2013-01-02
3	21803	1675.0	A97.0	2013	1	C	54	A	F	2013-01-03
4	21803	9047.0	A97.0	2013	1	C	28	A	F	2013-01-05
...	...	...	...	...	...	...	...	...	...	...
603182	110303	3443.0	A97.0	2023	42	P	21	A	M	2023-10-21
603183	60811	4256.0	A97.0	2023	42	P	12	A	F	2023-10-20
603184	240101	1882.0	A97.0	2023	42	C	15	A	F	2023-10-16
603185	240102	1866.0	A97.0	2023	42	P	26	A	M	2023-10-18

```
#df.dropna(subset=['renipress'], inplace=True)
```

```
#df_obj = df.select_dtypes(['object'])
#df_obj
```

```
df['renipress']
#df['renipress'] = df['renipress'].map(float)
```

```
0         1675.0
1         1675.0
2         1675.0
3         1675.0
4         9047.0
           ...
603182    3443.0
603183    4256.0
603184    1882.0
603185    1866.0
603186    5299.0
Name: renipress, Length: 603178, dtype: float64
```

```
df_3 = df.loc[df['renipress'].isin(id)]
df_3
```

	ubigeo	renipress	diagnostic	ano	semana	tipo_dx	edad	tipo_edad	sexo	fecha_ini
4	21803	9047.0	A97.0	2013	1	C	28	A	F	2013-01-05
10	21809	9047.0	A97.0	2013	1	C	77	A	M	2013-01-03
173	220101	6733.0	A97.0	2013	1	C	19	A	M	2013-01-05
192	220104	6733.0	A97.0	2013	1	C	45	A	F	2013-01-05
310	220101	6733.0	A97.0	2013	2	C	12	A	F	2013-01-10
...	...	...	...	...	...	...	...	...	...	...
603056	21809	9047.0	A97.1	2023	42	C	62	A	F	2023-10-16
603065	21801	9047.0	A97.1	2023	42	C	58	A	M	2023-10-16
603098	220101	6733.0	A97.1	2023	42	P	31	A	M	2023-10-20
603109	220101	6733.0	A97.0	2023	42	P	4	A	M	2023-10-20
603170	220101	6733.0	A97.0	2023	42	P	15	A	M	2023-10-20

16908 rows × 10 columns

```
pd.options.display.max_columns = None
url = "https://drive.google.com/file/d/1cn3KgOgUxSd14Mo4q-d6uFGWeHYrWr2s/view?usp=sharing"
file_id = url.split('/')[-2]
dwn_url = "https://drive.google.com/uc?id=" + file_id
lima_este = pd.read_csv(dwn_url)
lima_este
```

	YEAR	DOY	T2M_MAX	T2M_MIN	TS	RH2M	PRECTOTCORR	FechaMeteor
0	2012	1	23.26	17.15	22.17	75.56	0.00	20120101
1	2012	2	24.35	16.88	22.96	73.50	0.00	20120102
2	2012	3	23.73	16.37	22.58	76.25	0.03	20120103
3	2012	4	24.05	16.23	22.44	74.94	0.02	20120104
4	2012	5	20.95	16.08	20.64	79.75	0.02	20120105
...	...	...	...	...	...	...	...	...
4313	2023	296	21.71	16.28	20.06	74.81	1.23	20231023
4314	2023	297	22.30	16.10	20.17	75.69	3.20	20231024
4315	2023	298	20.56	15.46	19.08	82.56	13.28	20231025
4316	2023	299	20.75	14.41	19.18	81.75	1.34	20231026
4317	2023	300	20.66	14.08	18.83	83.06	0.92	20231027

```
lima_este['FechaMeteor'] = pd.to_datetime(lima_este['FechaMeteor'], format="%Y%m%d")
lima_este
```

	YEAR	DOY	T2M_MAX	T2M_MIN	TS	RH2M	PRECTOTCORR	FechaMeteor
0	2012	1	23.26	17.15	22.17	75.56	0.00	2012-01-01
1	2012	2	24.35	16.88	22.96	73.50	0.00	2012-01-02
2	2012	3	23.73	16.37	22.58	76.25	0.03	2012-01-03
3	2012	4	24.05	16.23	22.44	74.94	0.02	2012-01-04
4	2012	5	20.95	16.08	20.64	79.75	0.02	2012-01-05
...	...	...	...	...	...	...	...	...
4313	2023	296	21.71	16.28	20.06	74.81	1.23	2023-10-23
4314	2023	297	22.30	16.10	20.17	75.69	3.20	2023-10-24
4315	2023	298	20.56	15.46	19.08	82.56	13.28	2023-10-25
4316	2023	299	20.75	14.41	19.18	81.75	1.34	2023-10-26
4317	2023	300	20.66	14.08	18.83	83.06	0.92	2023-10-27

4318 rows × 8 columns

```
lima_este = lima_este.drop(labels=['YEAR', 'DOY'], axis=1)
lima_este
```

	T2M_MAX	T2M_MIN	TS	RH2M	PRECTOTCORR	FechaMeteor
0	23.26	17.15	22.17	75.56	0.00	2012-01-01
1	24.35	16.88	22.96	73.50	0.00	2012-01-02
2	23.73	16.37	22.58	76.25	0.03	2012-01-03
3	24.05	16.23	22.44	74.94	0.02	2012-01-04
4	20.95	16.08	20.64	79.75	0.02	2012-01-05
...	...	...	...	...	...	...
4313	21.71	16.28	20.06	74.81	1.23	2023-10-23
4314	22.30	16.10	20.17	75.69	3.20	2023-10-24
4315	20.56	15.46	19.08	82.56	13.28	2023-10-25
4316	20.75	14.41	19.18	81.75	1.34	2023-10-26
4317	20.66	14.08	18.83	83.06	0.92	2023-10-27

4318 rows × 6 columns

```
lima_este = lima_este.rename(columns={"FechaMeteor": "fecha_ini"})
```

```
df_conf = df_3[df_3['tipo_dx'] == 'C']
```

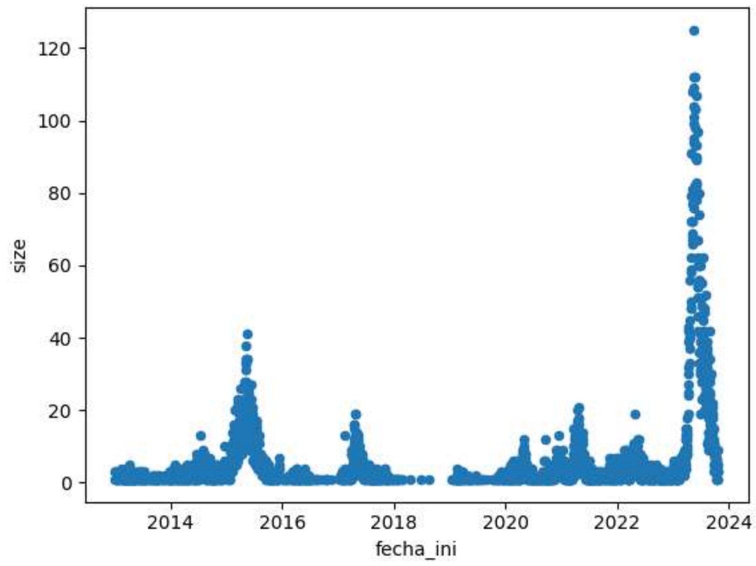
```
df_agrupado = df_3.groupby('fecha_ini', as_index=False).size()
df_agrupado
```

	fecha_ini	size
0	2013-01-03	1
1	2013-01-05	3
2	2013-01-06	1
3	2013-01-07	3
4	2013-01-10	1
...	...	...
2377	2023-10-16	5
2378	2023-10-17	3
2379	2023-10-18	3
2380	2023-10-19	3
2381	2023-10-20	9

2382 rows × 2 columns

```
df_agrupado.plot.scatter(x='fecha_ini', y = 'size')
```

```
<Axes: xlabel='fecha_ini', ylabel='size'>
```

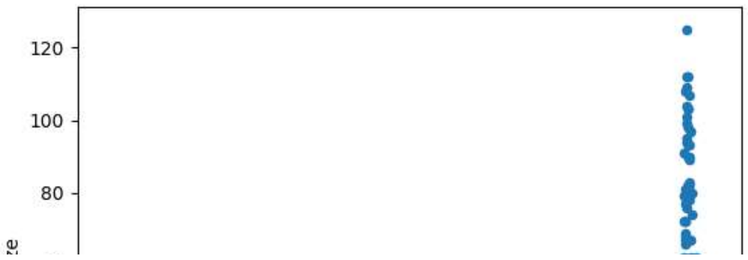


```
df_lima_este = df_agrupado.set_index('fecha_ini').join(lima_este.set_index('fecha_ini'))
```

```
df_lima_este = df_lima_este.reset_index()
df_lima_este = df_lima_este.dropna()
```

```
df_lima_este.plot.scatter(x='fecha_ini', y = 'size')
```

```
<Axes: xlabel='fecha_ini', ylabel='size'>
```



```
df_pred = df_lima_este
```

```
df_pred = df_pred.rename(columns={"fecha_ini": "ds", "size": "y"})
df_pred
```

	ds	y	T2M_MAX	T2M_MIN	TS	RH2M	PRECTOTCORR
0	2013-01-03	1	23.54	15.94	22.33	71.69	0.00
1	2013-01-05	3	24.55	16.30	22.42	70.25	0.00
2	2013-01-06	1	23.01	16.31	22.13	70.44	0.03
3	2013-01-07	3	23.57	15.95	22.48	69.62	0.02
4	2013-01-10	1	24.17	16.85	23.37	68.56	0.01
...	...	...	...	...	...	...	...
2377	2023-10-16	5	23.81	15.32	21.07	70.69	1.03
2378	2023-10-17	3	22.28	15.84	20.30	77.69	4.60
2379	2023-10-18	3	22.62	16.04	20.19	77.38	6.34
2380	2023-10-19	3	23.33	16.48	20.89	73.50	1.54
2381	2023-10-20	9	22.75	15.60	20.61	74.44	1.08

2382 rows × 7 columns

```
df_pred.corrwith(df_pred["y"])
```

```
<ipython-input-92-776c8a4a01bd>:1: FutureWarning: The default value of numeric_only in DataFrame.corrwith is deprecated. In a future ver
  df_pred.corrwith(df_pred["y"])
y              1.000000
T2M_MAX        0.248146
T2M_MIN        0.101274
TS             0.023776
RH2M           0.097446
PRECTOTCORR   -0.024248
dtype: float64
```

```
train_end_date = '2023-09-20'

# Train test split
train = df_pred[df_pred['ds'] <= train_end_date]
test = df_pred[df_pred['ds'] > train_end_date]

print(train.shape)
print(test.shape)
```

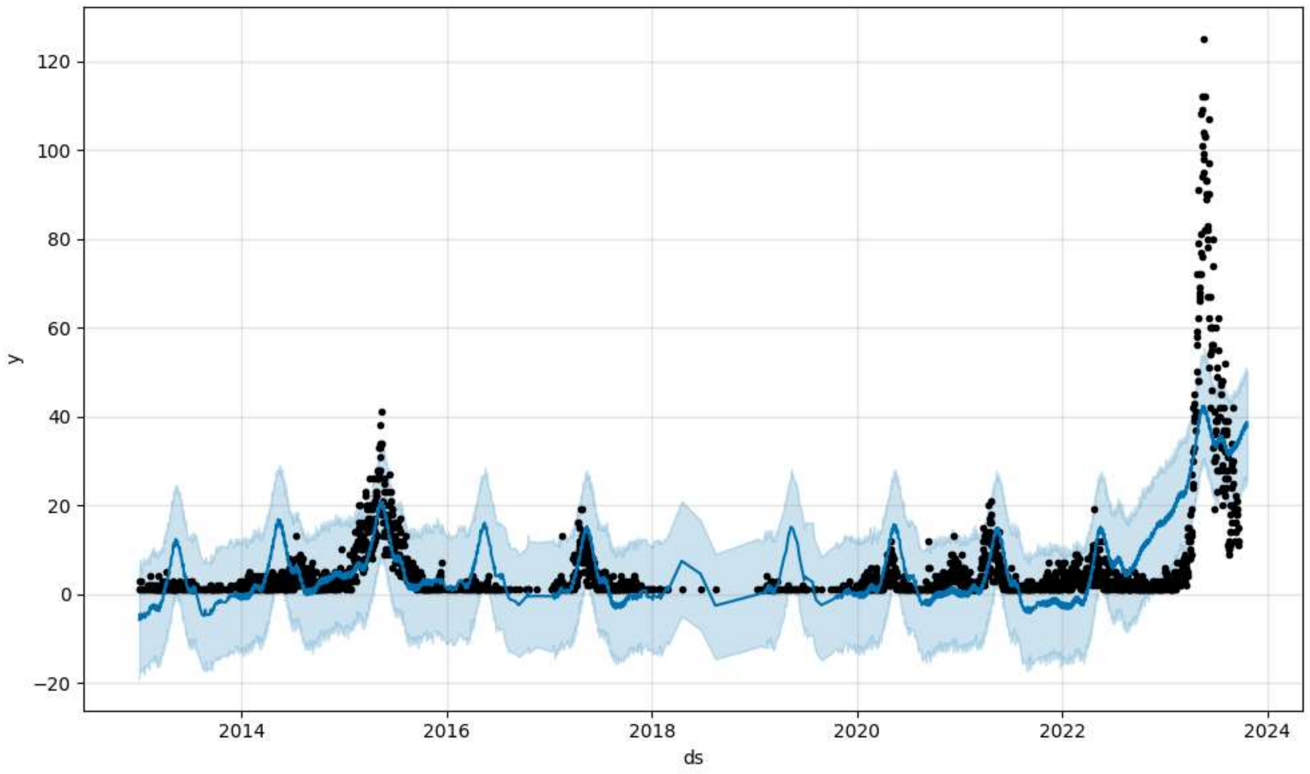
```
(2352, 7)
(30, 7)
```

```
model_baseline = Prophet(yearly_seasonality=True)

# Fit the model on the training dataset
model_baseline.fit(train)
```

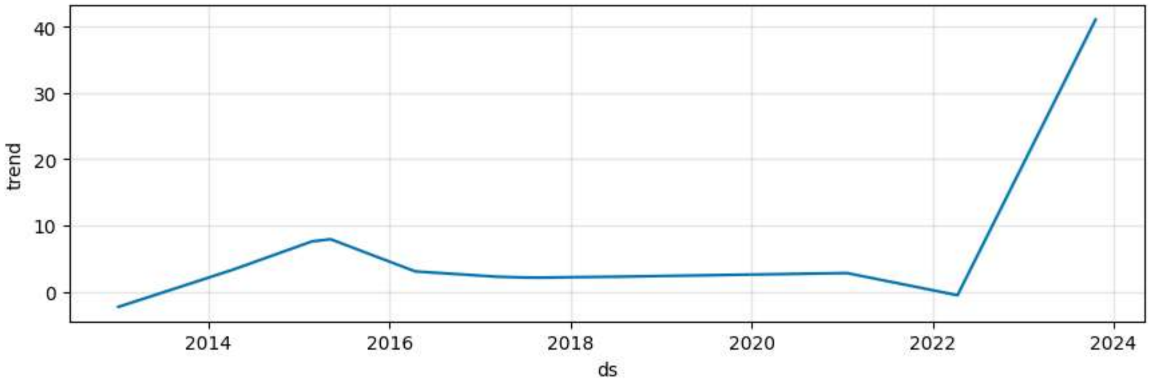
```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmp5m3ik_nn/9_asx45b.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmp5m3ik_nn/dc4zn9dd.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.10/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=11075', '
```

```
16:51:36 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
16:51:36 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
<prophet.forecaster.Prophet at 0x7952e40ab160>
```

```
# Create the time range for the forecast
future_baseline = model_baseline.make_future_dataframe(periods=30)

# Make prediction
forecast_baseline = model_baseline.predict(future_baseline)

# Visualize the forecast
model_baseline.plot(forecast_baseline);
```



```
model_baseline.plot_components(forecast_baseline);
```



```
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error

# Merge actual and predicted values
performance_baseline = pd.merge(test, forecast_baseline[['ds', 'yhat', 'yhat_lower', 'yhat_upper']][-30:], on='ds')

# Check MAE value
performance_baseline_MAE = mean_absolute_error(performance_baseline['y'], performance_baseline['yhat'])
print(f'The MAE for the baseline model is {performance_baseline_MAE}')

# Check MAPE value
performance_baseline_MAPE = mean_absolute_percentage_error(performance_baseline['y'], performance_baseline['yhat'])
print(f'The MAPE for the baseline model is {performance_baseline_MAPE}')
```

```
The MAE for the baseline model is 30.626409928056123
The MAPE for the baseline model is 8.961592211598449
```

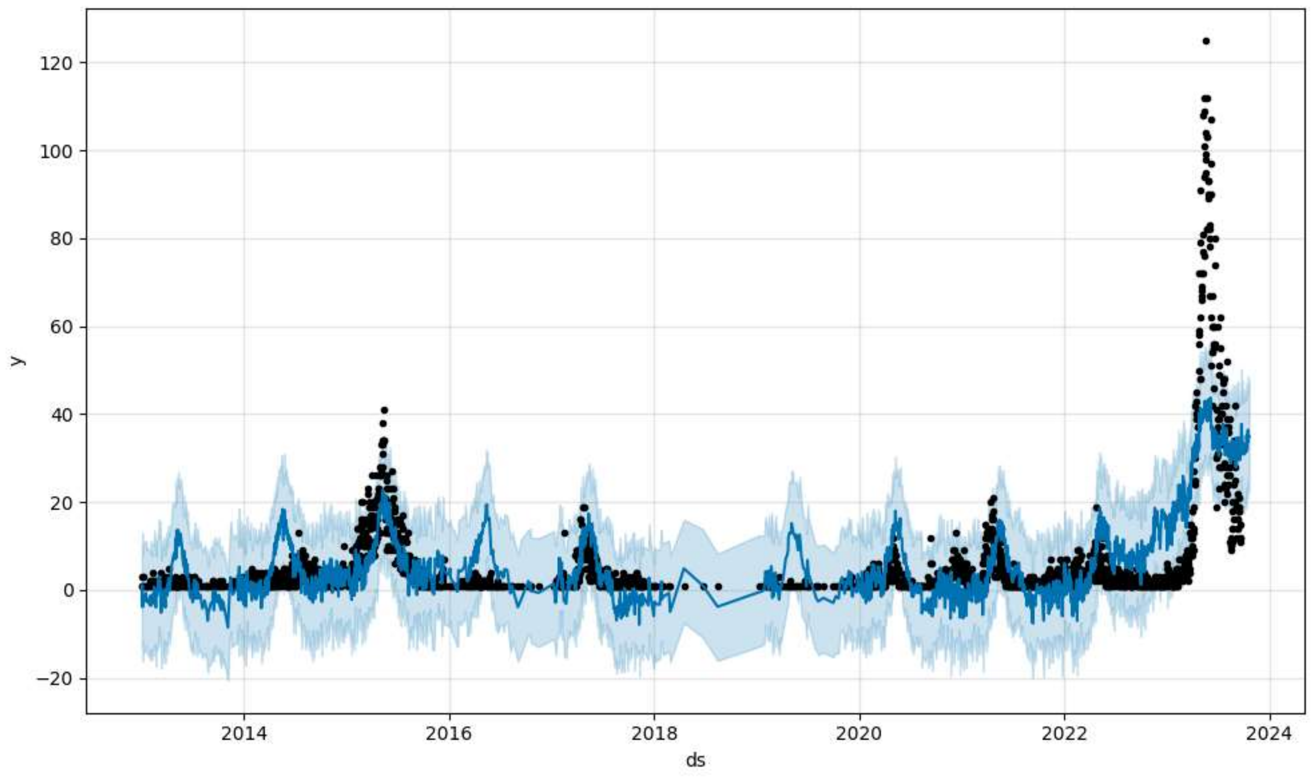
```
# Add seasonality
model_multivariate = Prophet(yearly_seasonality=True)

# Add regressors
model_multivariate.add_regressor('T2M_MAX', standardize=False)
#model_multivariate.add_regressor('T2M_MIN', standardize=False)
model_multivariate.add_regressor('RH2M', standardize=False)

# Fit the model on the training dataset
model_multivariate.fit(train)
```

```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmp5m3ik_nn/m16ih8y3.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmp5m3ik_nn/j0i3jpxy.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.10/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=14850', '
17:19:19 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
17:19:20 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
<prophet.forecaster.Prophet at 0x7952e16b5b10>
```

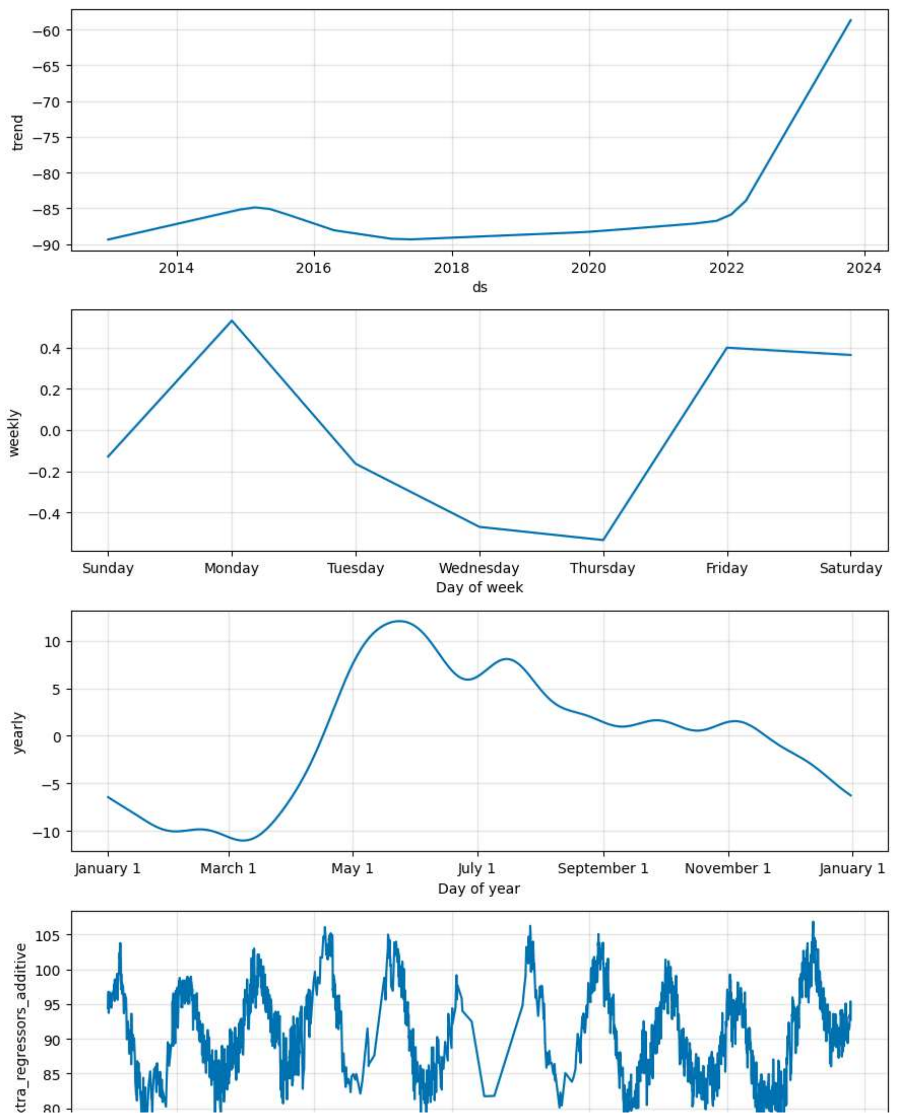
```
# Create the time range for the forecast
future_multivariate = model_multivariate.make_future_dataframe(periods=30)

# Append the regressor values
future_multivariate = pd.merge(future_multivariate, df_pred[['ds', 'T2M_MAX', 'RH2M']], on='ds', how='inner')

# Fill the missing values with the previous value
future_multivariate = future_multivariate.fillna(method='ffill')

# Check the data
future_multivariate.tail(10)
```

	ds	T2M_MAX	RH2M
2372	2023-10-11	24.00	68.81
2373	2023-10-12	23.55	69.81
2374	2023-10-13	22.19	71.06
2375	2023-10-14	22.33	73.38
2376	2023-10-15	23.47	69.88
2377	2023-10-16	23.81	70.69
2378	2023-10-17	22.28	77.69
2379	2023-10-18	22.62	77.38

```
# Make prediction
forecast_multivariate = model_multivariate.predict(future_multivariate)

# Visualize the forecast
model_multivariate.plot(forecast_multivariate);
```



```
# Visualize the forecast components
model_multivariate.plot_components(forecast_multivariate);
```



```
# Merge actual and predicted values
performance_multivariate = pd.merge(test, forecast_multivariate[['ds', 'yhat', 'yhat_lower', 'yhat_upper']][-30:], on='ds')

# Check MAE value
performance_multivariate_MAE = mean_absolute_error(performance_multivariate['y'], performance_multivariate['yhat'])
print(f'The MAE for the multivariate model is {performance_multivariate_MAE}')

# Check MAPE value
performance_multivariate_MAPE = mean_absolute_percentage_error(performance_multivariate['y'], performance_multivariate['yhat'])
print(f'The MAPE for the multivariate model is {performance_multivariate_MAPE}')

     The MAE for the multivariate model is 27.56990391720814
     The MAPE for the multivariate model is 8.040513170672945
```