■■■■■ ■■■■ ■■■■■■■ ■■■■■■ ■■■ ■■ Polars

■■■ 9: Time Series Analysis

■■■■■: 02/11/2025

■■■■ ■■■■■■■■

1. [redacted]

[redacted] [redacted] [redacted] [redacted] [redacted] [redacted] (Time Series Analysis) [redacted] [redacted] Polars - [redacted] [redacted] [redacted] [redacted] [redacted] [redacted].

[redacted]:

Python 3.8+	[redacted]
Polars	[redacted]
Datetime	[redacted] ([redacted])
[redacted]	toronto_weather.csv

[redacted]:

```
import polars as pl
lf = pl.scan_csv('../data/toronto_weather.csv')
lf = lf.with_columns(pl.col('temperature')-273.15)
```

2. ■■■■■■ ■■ ■■■■■■■■■ ■■■■■■

■■■■■■ ■■■■■■ ■■ ■■■■■■■■ ■■■■ ■■■■■■ ■■■■■■■ ■■■■■■ ■■■■. Polars ■■■■■ ■■■■■ ■■■■■■ ■■■■■■, ■■■■■■ ■■■■■■■ ■■ ■■■■■■■■.

2.3 ■■■■■■■■■ ■■■■■■ ■■■■■■■■:

■■■■■■■■	■■■■■■	■■■■■■ ■■■■
.dt.year()	■■■■■ ■■■■	2024
.dt.month()	■■■■■ ■■■■■	1-12
.dt.day()	■■■■■ ■■■■	1-31
.dt.hour()	■■■■■ ■■■■	0-23
.dt.weekday()	■■■■ ■■■■■■	0-6
.dt.date()	■■■■■■ ■■■■■	2024-01-15
.dt.time()	■■■■ ■■■■■	14:30:00

```
lf.select(
'datetime',
pl.col('datetime').dt.year().alias('year'),
pl.col('datetime').dt.month().alias('month'),
pl.col('datetime').dt.day().alias('day')
)
```

3. ■■■■■■ ■■■■■■■■■ (Rolling Windows)

■■■■ ■■■■■■■ ■■■ ■■■■■■■ ■■■■■■■ ■■■■■■ ■■ ■■■■■ ■■■■■■ ■■■ ■■ ■■■■■■■. ■■■ ■■■ ■■■■■■ ■■■■■■■ ■■■■■■■, ■■■■■■ ■■■■■■■, ■■■■■■■ ■■■■■■■■■■■■■ ■■■■■■■.

■■■■■■■■	■■■■■■	■■■■■■ ■■■■■
rolling_mean()	■■■■■■ ■■	■■■■■■ ■■■■■■■
rolling_min()	■■■■■■■■ ■■	■■■■■■ ■■■■■■■
rolling_max()	■■■■■■■■ ■■	■■■■■■ ■■■■■■
rolling_sum()	■■■■■ ■■	■■■■■■■ ■■■■■■■■
rolling_std()	■■■■■■ ■■■■ ■■■■	■■■■■■ ■■■■■■■■■
rolling_map()	■■■■■■■■ ■■■■■■■	■■■■■■■■ ■■■■■■■■

```
# ■■■■■■ ■■ ■■■■■
pl.col('temperature').rolling_mean(3)

# ■■ min_periods ■■■■■■■ nulls
pl.col('temperature').rolling_mean(
window_size=7,
min_periods=1
)
```

4. ■■■■■■■ ■■■■■■ ■■■■ (Resampling)

Resampling ■■■■■■ ■■■■■■ ■■ ■■■■■■■ ■■■■■■■■ - ■■■■■■■ ■■ ■■■■■■■ ■■ ■■■■■■■■■■ ■■■■■■■. ■■■■ ■■■■ ■■■■■■ ■■■■■■■ ■■■■■■■ ■■■■■■ ■■■■■■.

■■■■■■	■■■■■	■■■■■■■	■■■■■
Downsampling	■■■■■	group_by_dynamic	■■■■ → ■■■■
Upsampling	■■■■■	upsample	■■■■ → ■■■■
Interpolation	■■■■■	interpolate	■■■■■ nulls

■■■■■ - Downsampling:

```
lf.set_sorted('datetime').group_by_dynamic(
'datetime',
every='1w' # ■■ ■■■■
).agg(
pl.col('humidity').mean()
)
```

■■■■■ - Upsampling + Interpolation:

```
df.upsample(
time_column='datetime',
every='30m'
).with_columns(
pl.col('humidity').interpolate()
)
```

5. ■■■■■ ■■■■■■ ■■■■ ■■ Functime

■■■■■■ ■■■■■■ ■■■■ ■■■■ ■■■■■■ ■■ ■■■■■■ ■■■■■■ ■■■■■■■■■ ■■ ■■■■■ ■■■■■■■■ ■■■■■■■■■■. Functime ■■■■ ■■■■■■■ ■■■■■■■ ■■■■■■■ ■■■■■■■ ■■■■■■ ■■ Polars.

■■■■■■■ ■■■■■■■:

■■■■■	■■■■■■	■■■■■■ ■
linear_model	■■■■■■■ ■■■■■■■■	■■■■■■
lightgbm	LightGBM	■■■■■■ ■■■■■
xgboost	XGBoost	■■■■■ ■■■■■
knn	K-Nearest Neighbors	■■■■■■■ ■■■■■■■
auto_lightgbm	AutoML	■■■■■■■■■■■■ ■■■■■■■■■

```
from functime.forecasting import linear_model
from functime.metrics import mase

# ■■■■■■ ■■■■■
forecaster = linear_model(lags=24, freq='1mo')
forecaster.fit(y=y_train)

# ■■■■■■
y_pred = forecaster.predict(fh=3)

# ■■■■■■
scores = mase(y_true=y_test, y_pred=y_pred, y_train=y_train)
```

■■■■■ ■■■■■■:

■■■■	■■■■■■	■■■■ ■■■■■■■
MASE	Mean Absolute Scaled Error	■■■■■ (■■■■■■)
SMAPE	Symmetric MAPE	■■■■■■■ ■■ 0
MAE	Mean Absolute Error	■■■■■
RMSE	Root Mean Squared Error	■■■■■■

6. ■■■■■ ■■■■■■■ ■■■■■■■

■■■■■■■ ■■ ■■■■■■ ■■ ■■■■■■■■■ ■■■■■■■■■■ ■■■■■■■■■ ■■■■■■■ ■■■■■■ ■■■■ ■■ Polars. ■■■■■ ■■■■■■ ■■■■■■■■ ■■■■■■■■ ■■■■■■:

■■■■	■■■	■■■■■■■
1	■■■■■ ■■■■■■ ■-set_sorted() ■■■■■ ■■■■■■■ ■■■■	■■■■■■
2	■■■■■ ■■ head() ■■■■■ collect()	■■■■■ ■■■■■
3	■■■■■ ■■■■■■■■■ ■■■■■■■ ■■ rolling_map	■■■■■■
4	upsample ■■■■■ DataFrame (■■ LazyFrame)	■■■■■
5	■■■■■■ ■-min_periods=1 ■■■■■■■ nulls	■■■■■■■
6	■■■■ ■■ ■■■■■ ■■■■	■■■■■■
7	■■■■ ■■■■■■■■■■■■ ■■■■■■■ ■■■■■■■	■■■■■
8	■■■■■ ■■ maintain_order=True ■group_by	■■■■■■■

■■■■■■■ ■■■■■■■ (Pitfalls):

■■■■■	■■■■■
datetime is not sorted	■■■■■ set_sorted('datetime')
upsample requires DataFrame	■■■■■■ ■-collect() ■■■■■
■■■■■ ■■■■ nulls ■■■■■■■■	■■■■■ min_periods=1
rolling_map ■■■■■	■■■■■ ■■■■■■■■■ ■■■■■■■
■■■■■ ■■■■ ■■ ■■■■■	■■■■■ replace vs convert
■■■■■■ ■-String	■■■■ ■-Datetime

■■■■■■■ ■■■■■■■:

• ■■■■■■ ■■■■■: https://pola-rs.github.io/polars/
• API Reference: https://pola-rs.github.io/polars/py-polars/html/reference/
• Discord ■■■■■■: https://discord.gg/4UfP5cfBE7
• GitHub: https://github.com/pola-rs/polars

■■■■■■■ ■■■■■■■ ■■■■■■ ■■■■■ ■■■■■! ■■■■■ ■■■■■■ ■■ ■■■■■■■ ■■■■■■■■ ■■■■■■■■ ■■■■■■■■■ ■■■■■■■■ ■■■■■■■ ■■■■■■ ■■■■■■■■■■ ■■■■■.