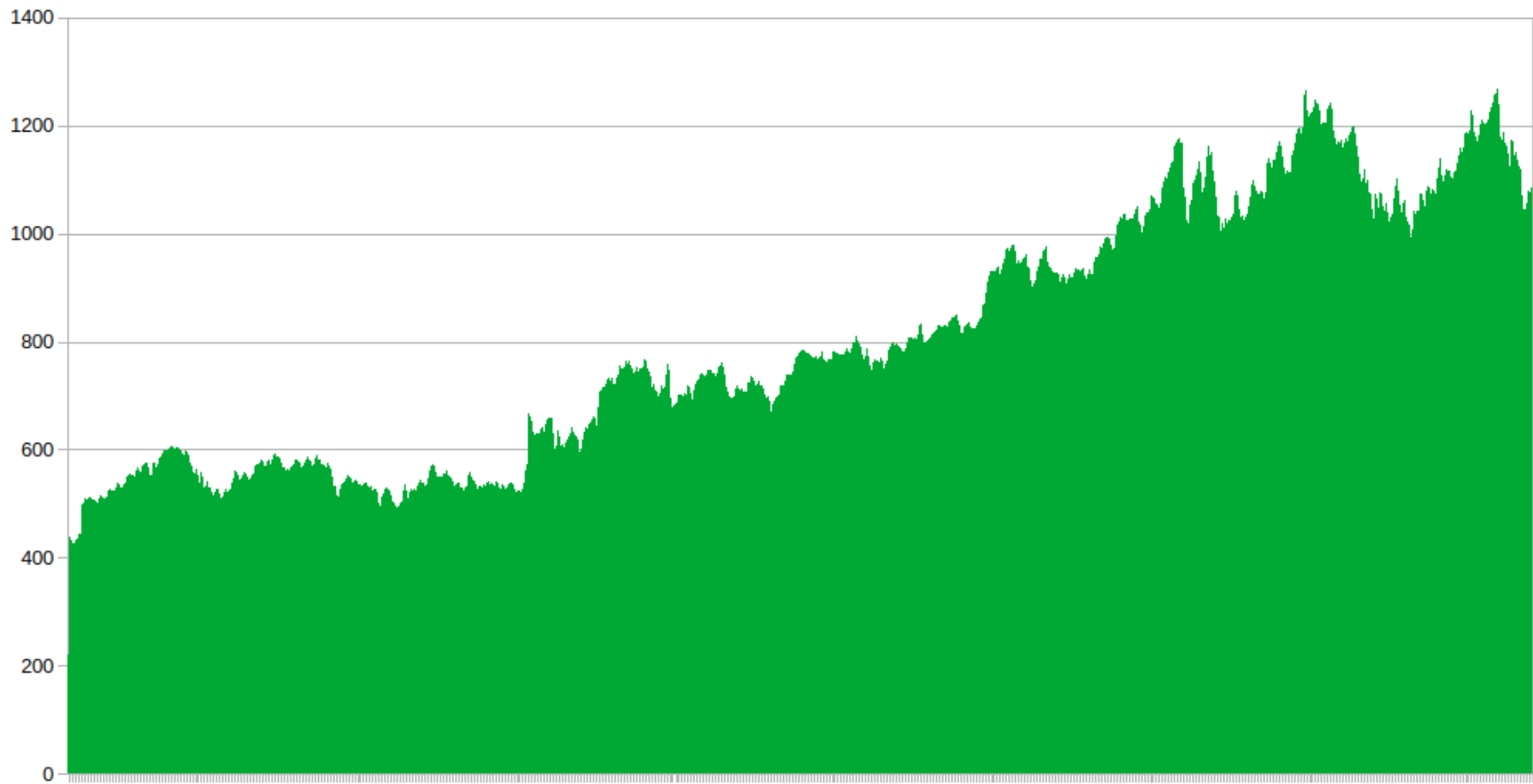


# Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



# Sammamish, WA

Monday 11:00 AM

Cloudy



61 °F | °C

Precipitation: 0%

Humidity: 79%

Wind: 5 mph

Temperature

Precipitation

Wind

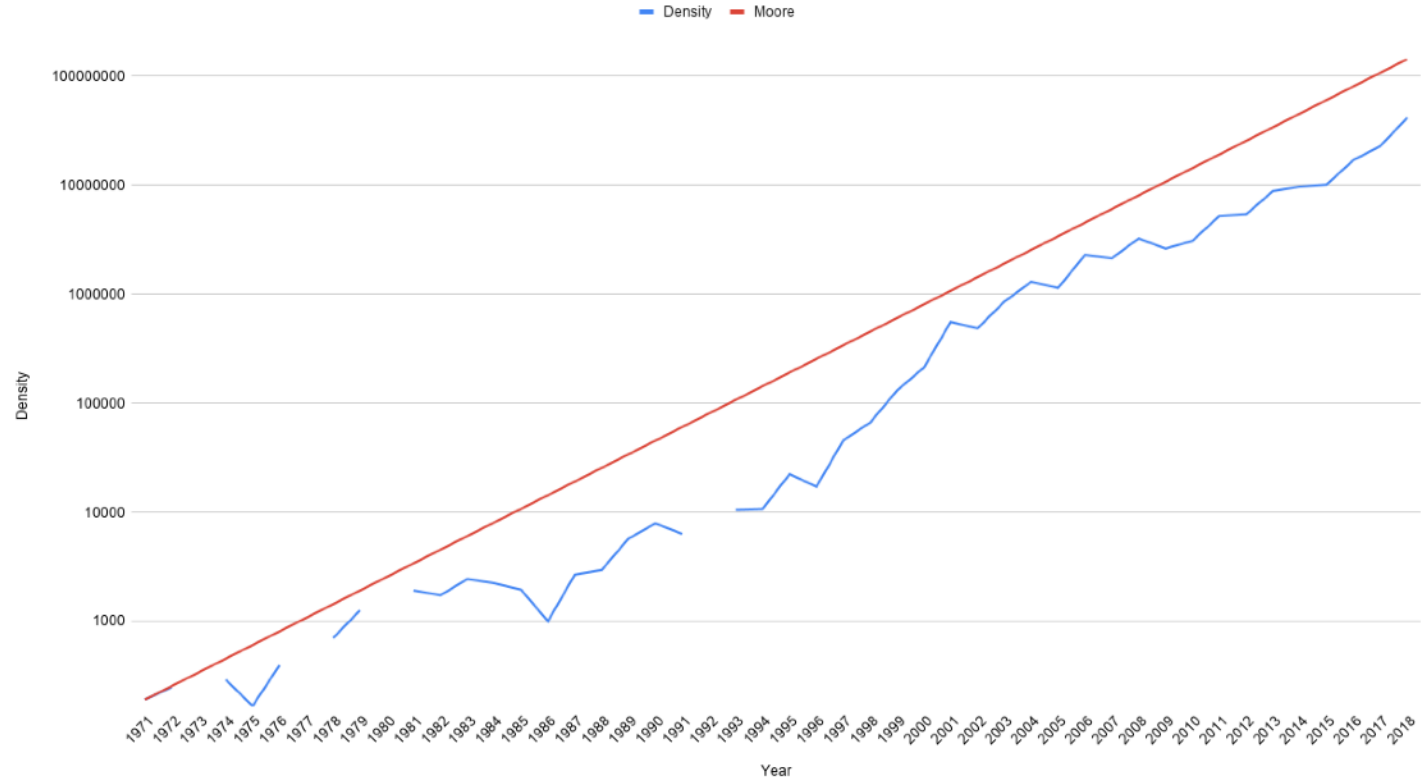


Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
75° 56°	67° 51°	64° 50°	66° 51°	73° 53°	72° 53°	65° 53°	65° 53°

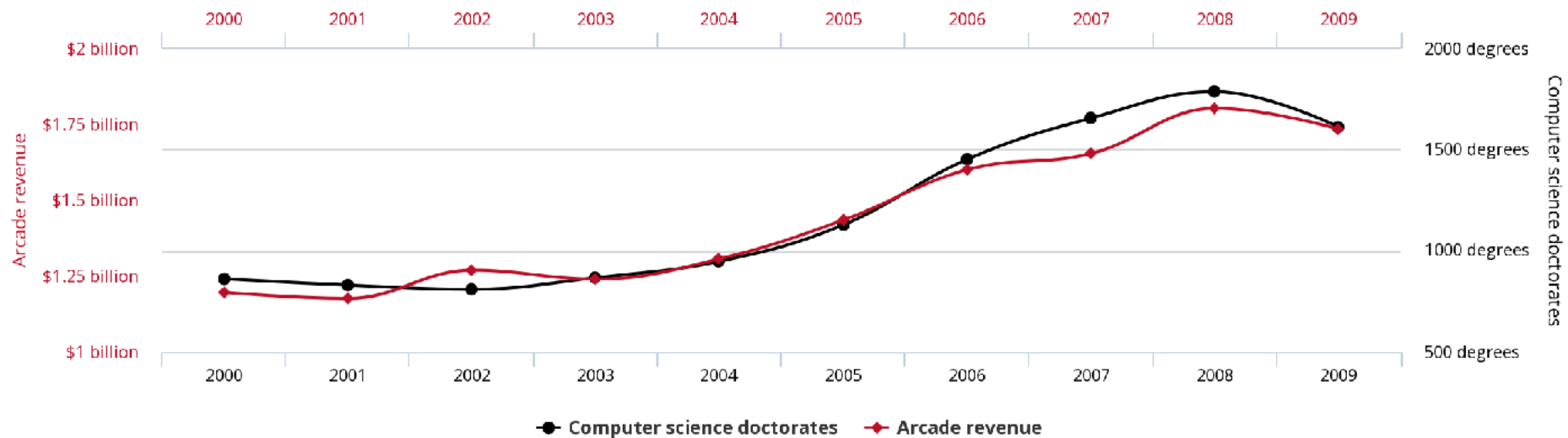
[More on weather.com](#)

[Feedback](#)

Density vs. Year

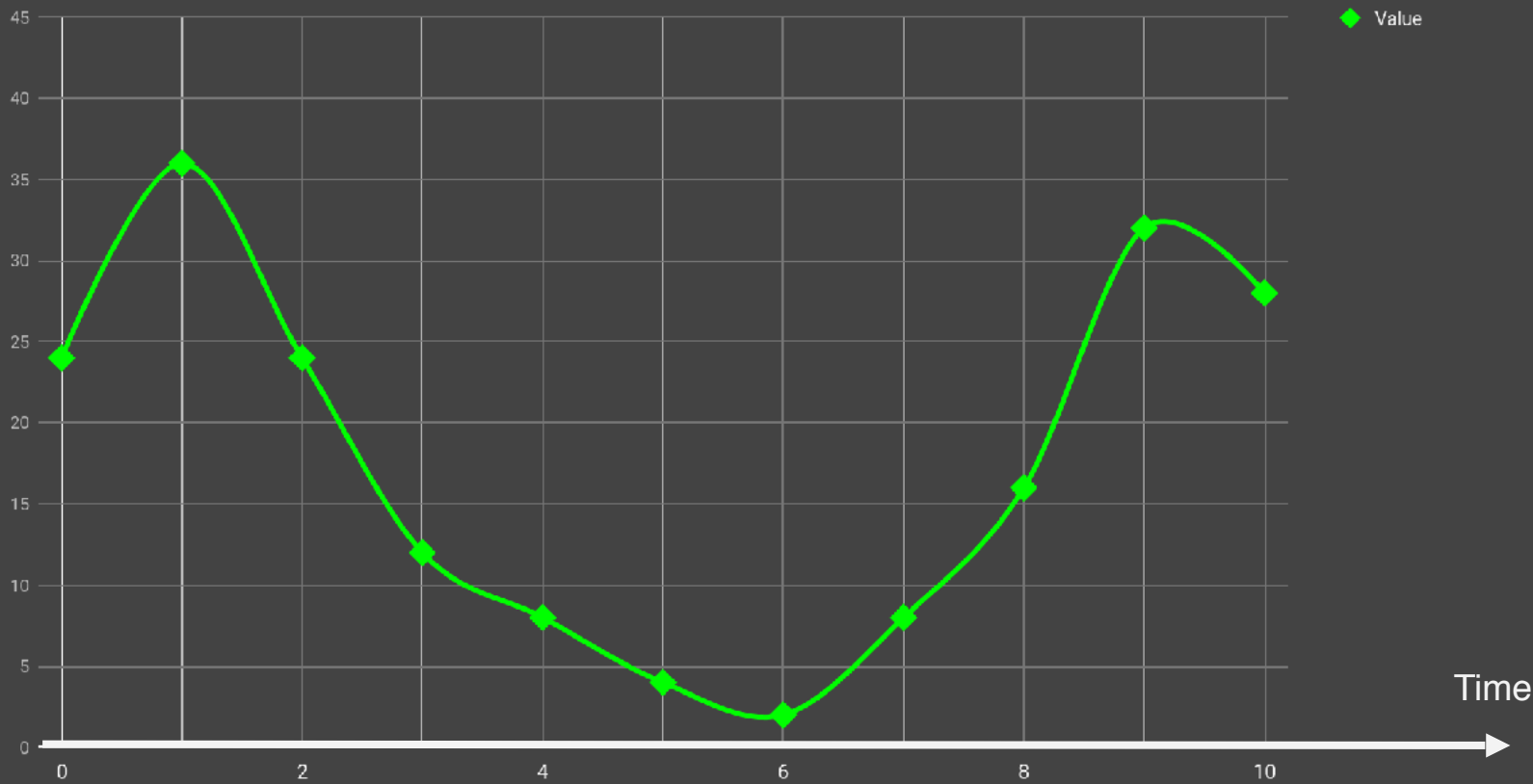


**Total revenue generated by arcades**  
correlates with  
**Computer science doctorates awarded in the US**

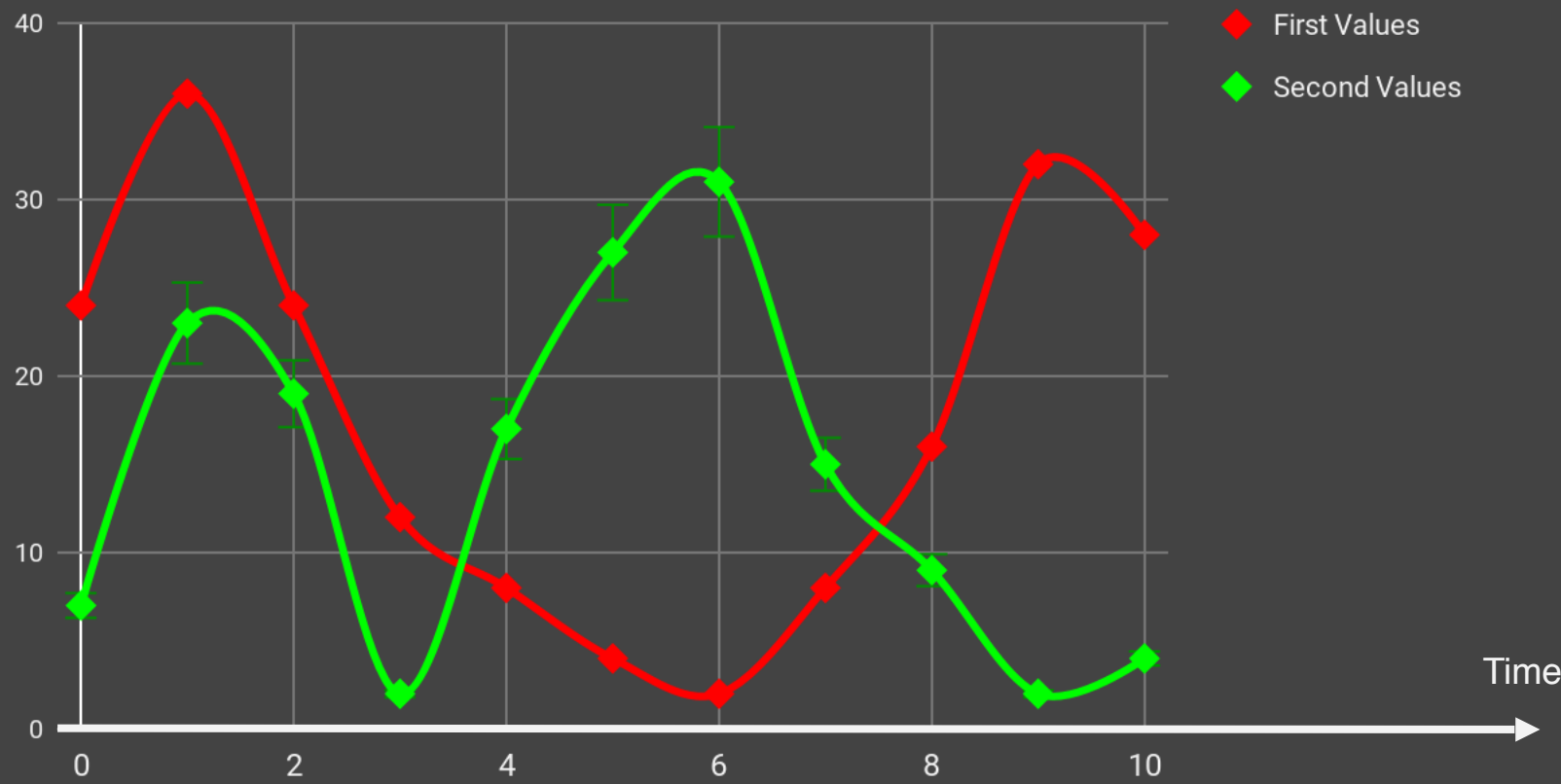


tylervigen.com

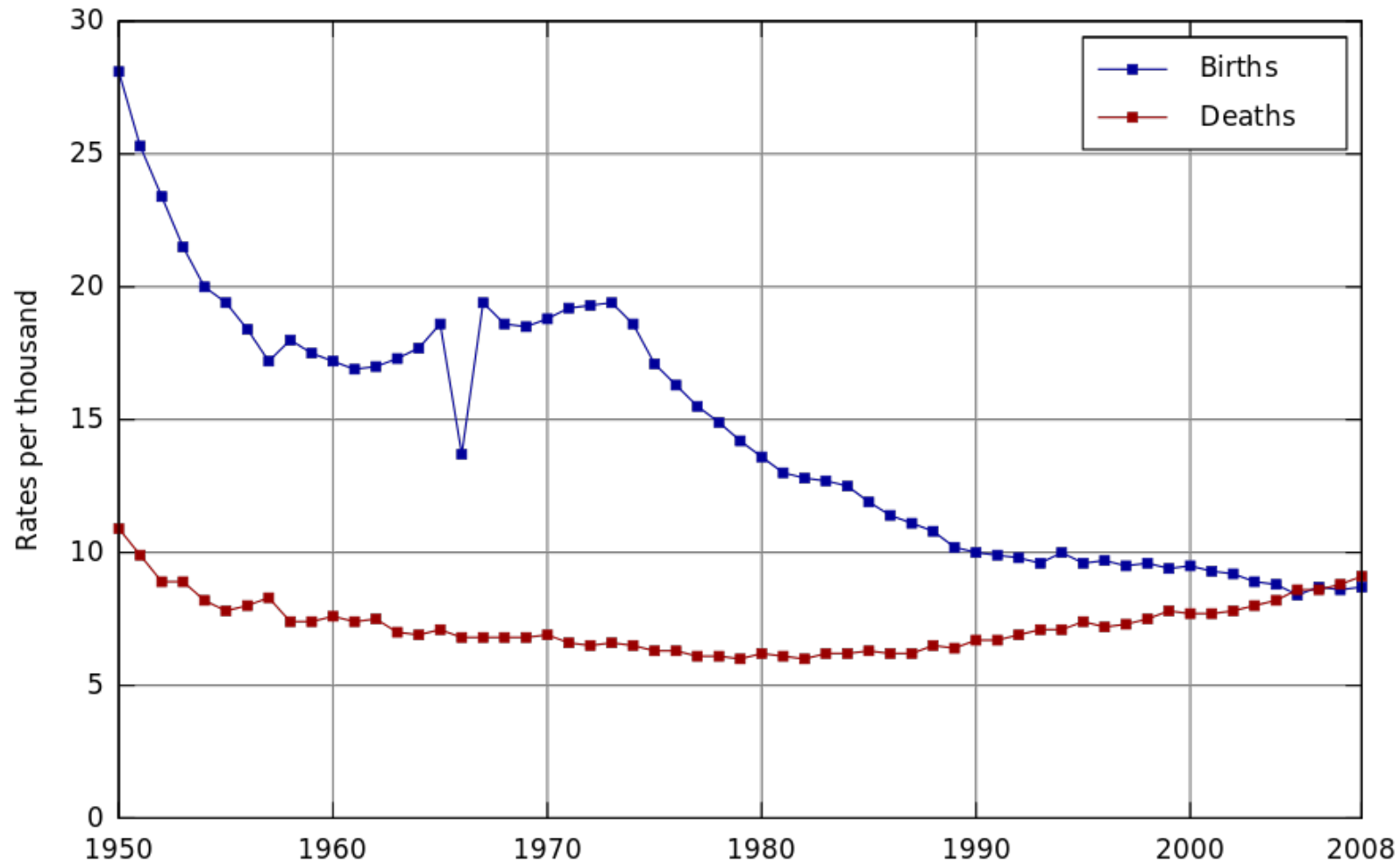
# Univariate Time Series



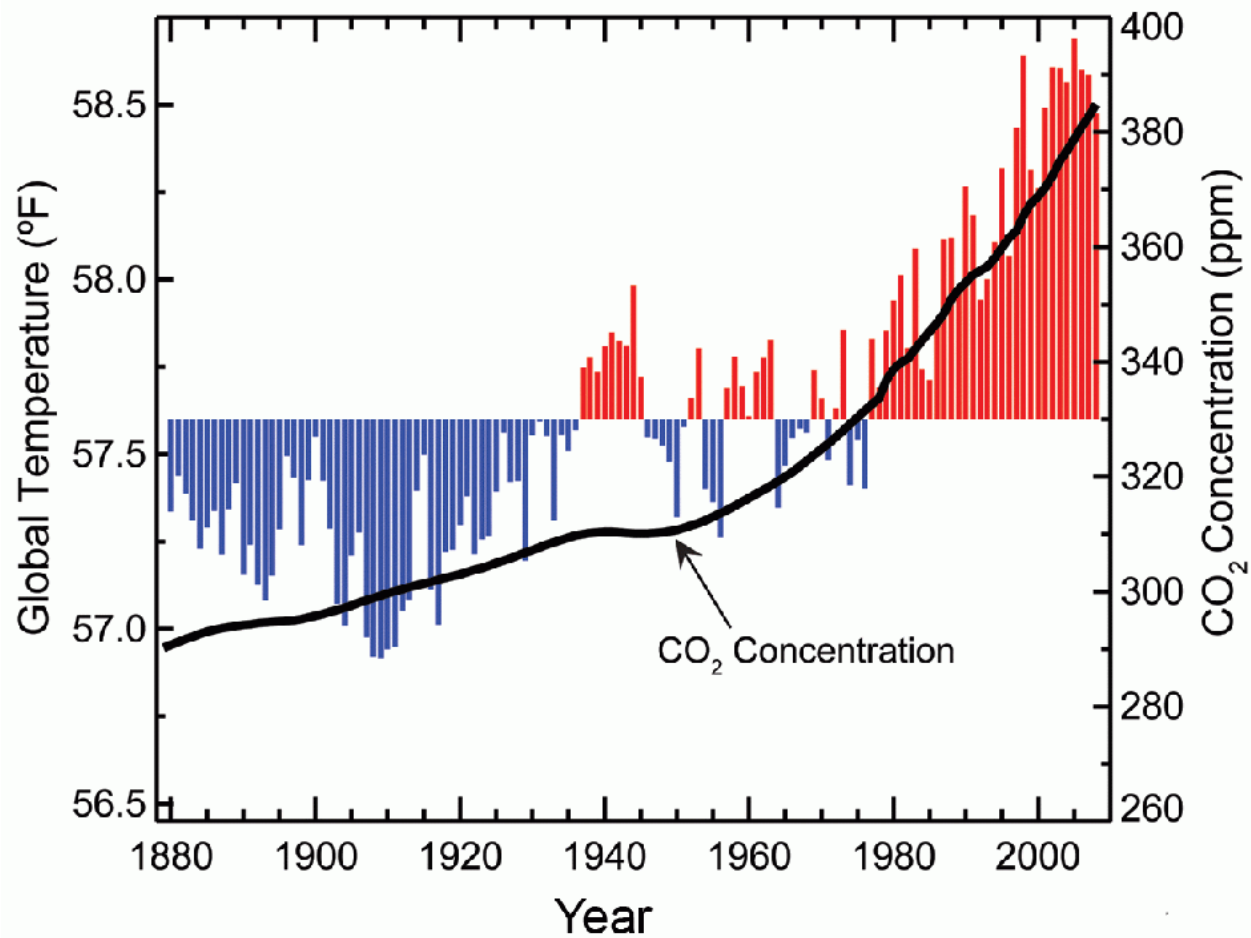
# Multivariate Time Series

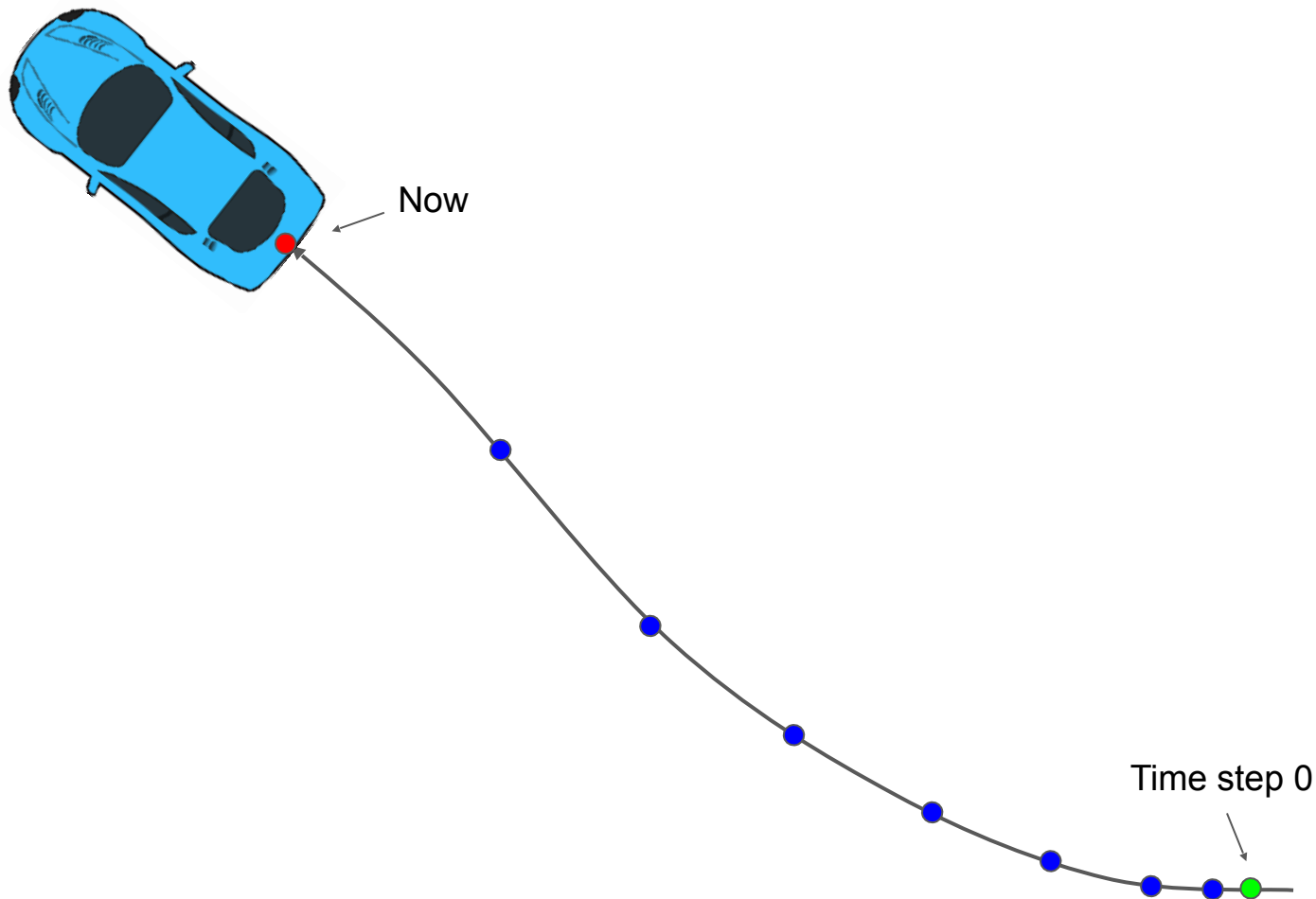


# Birth and Death Rate in Japan

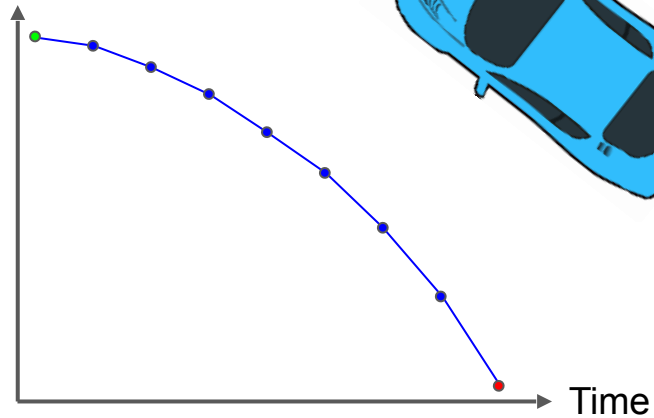




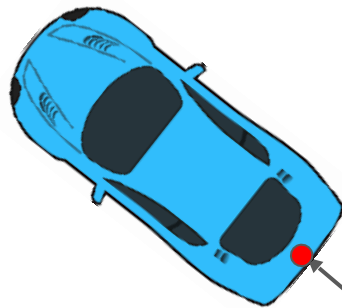
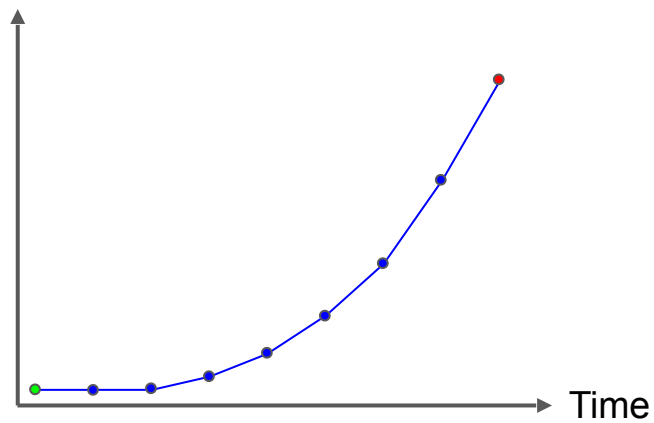




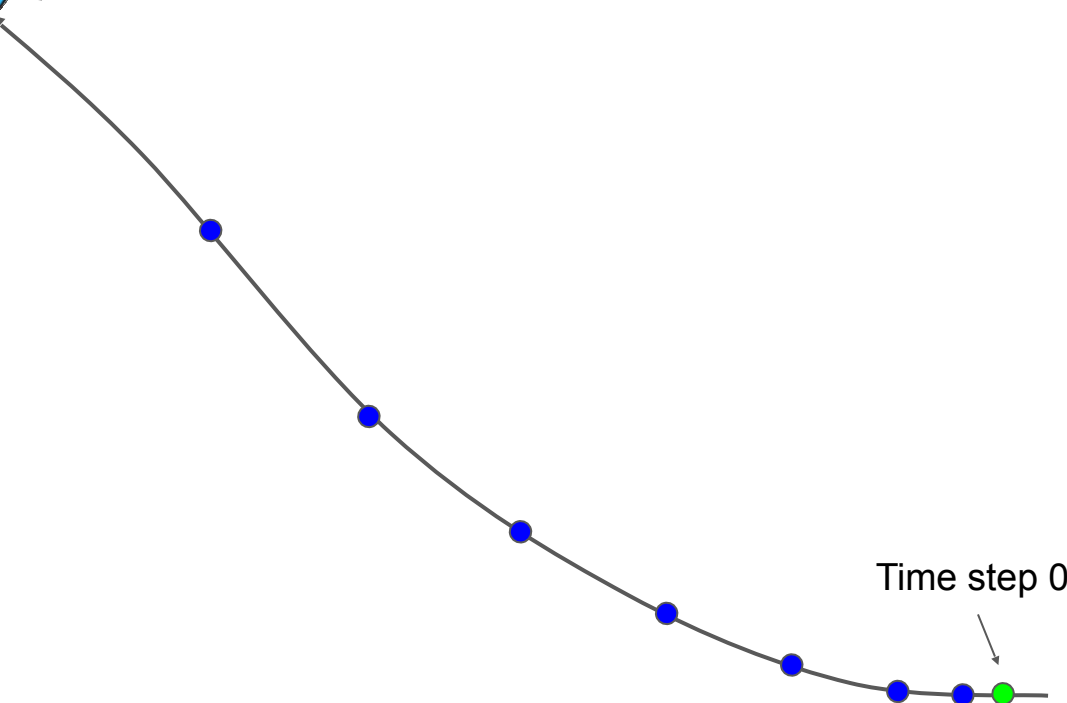
Longitude



Latitude

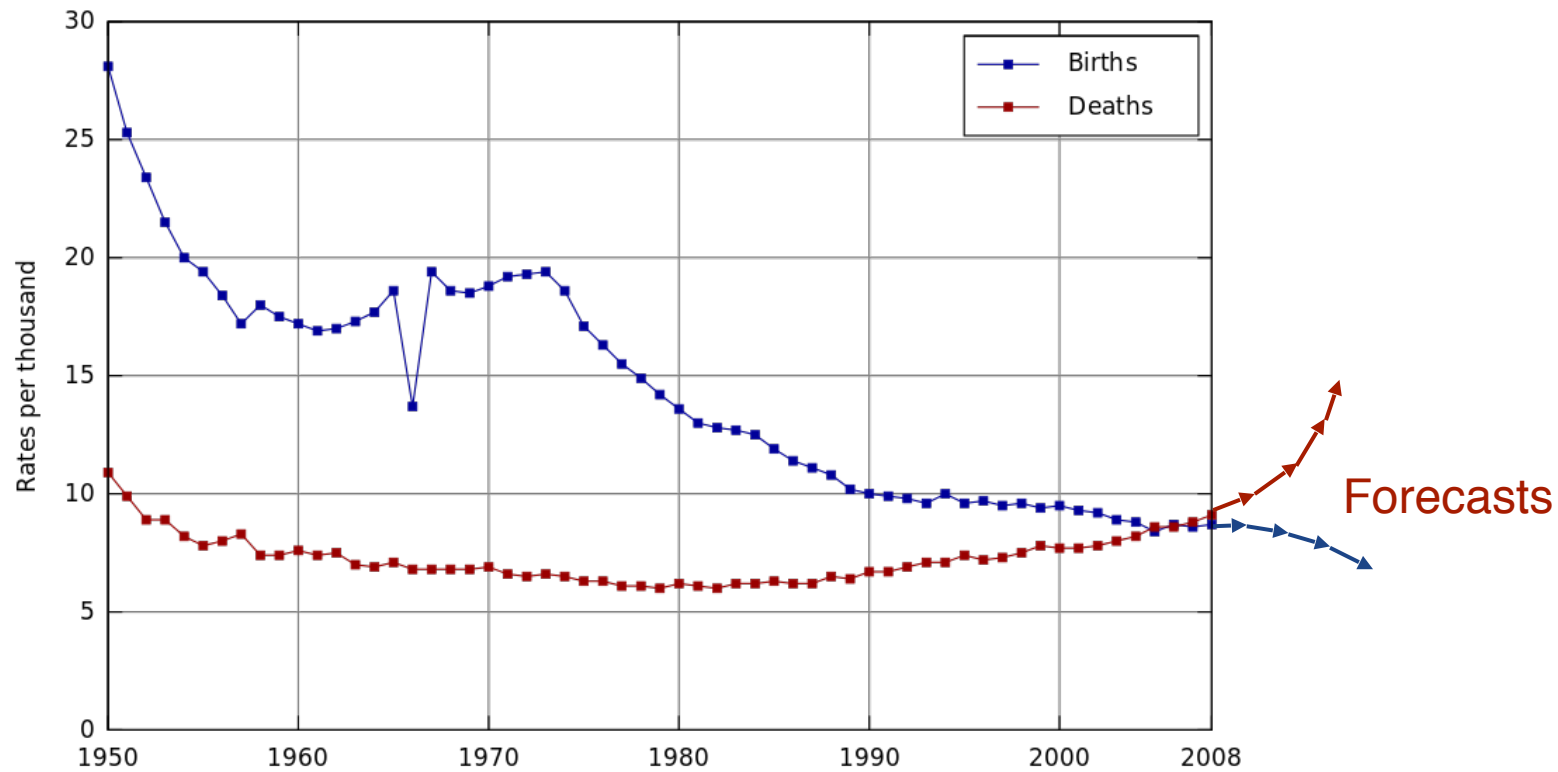


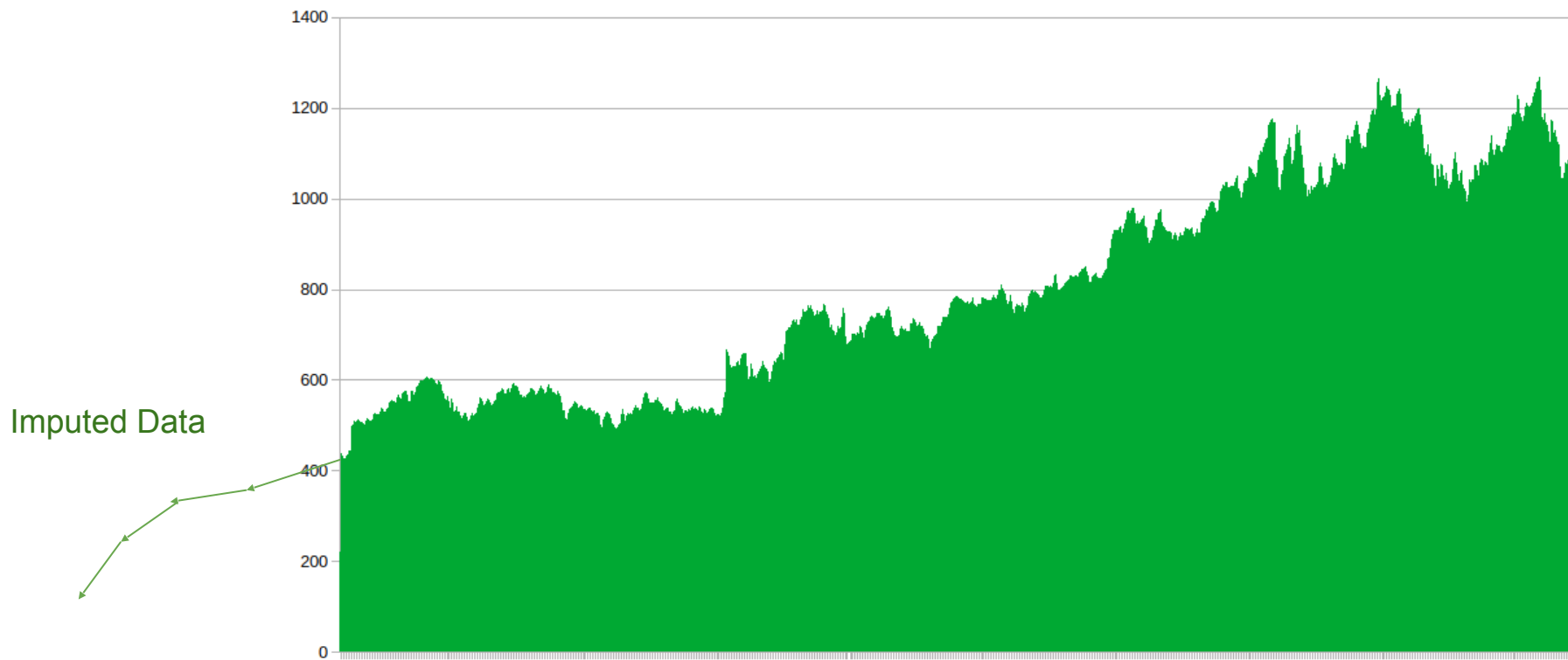
Now



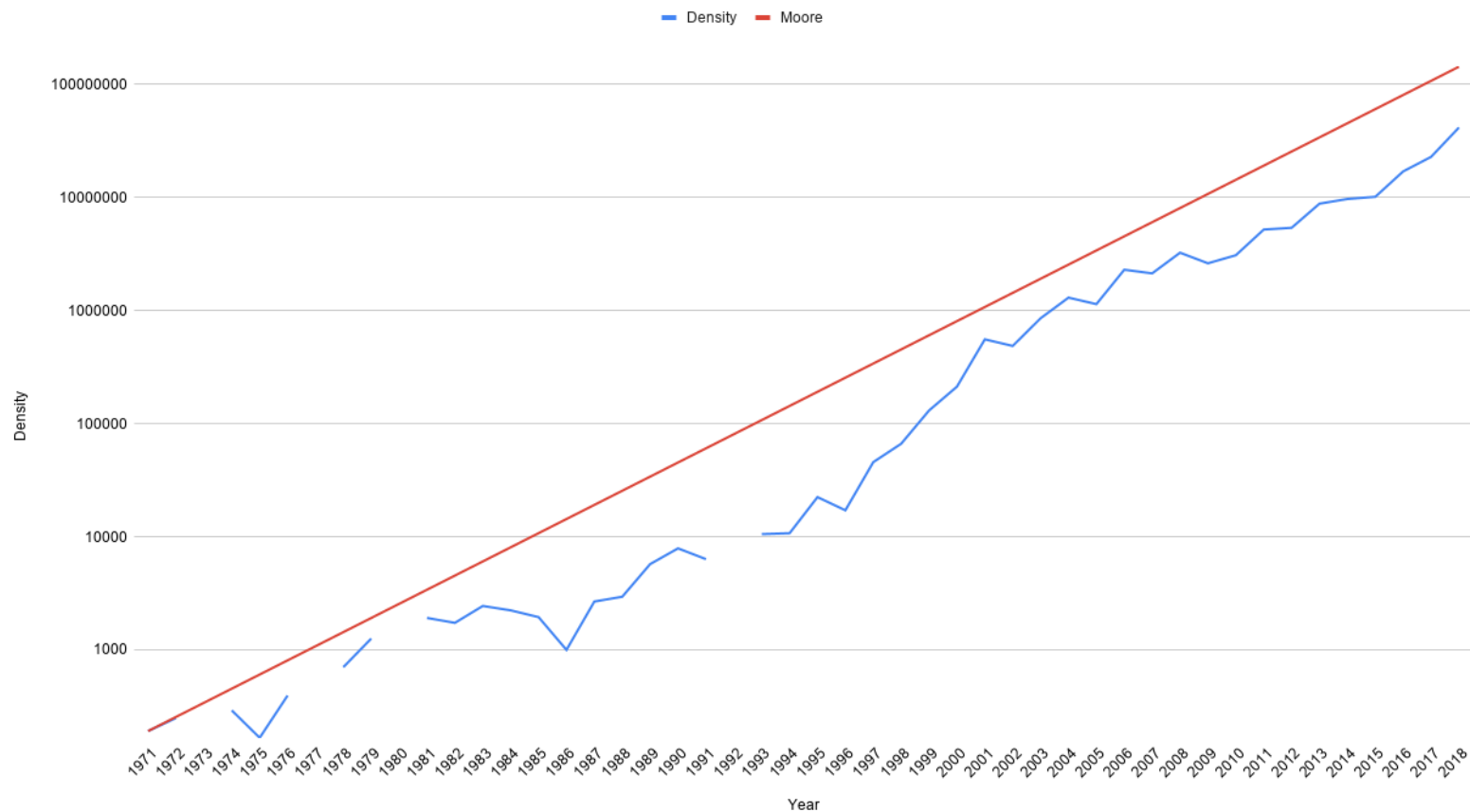
Time step 0

# Birth and Death Rate in Japan

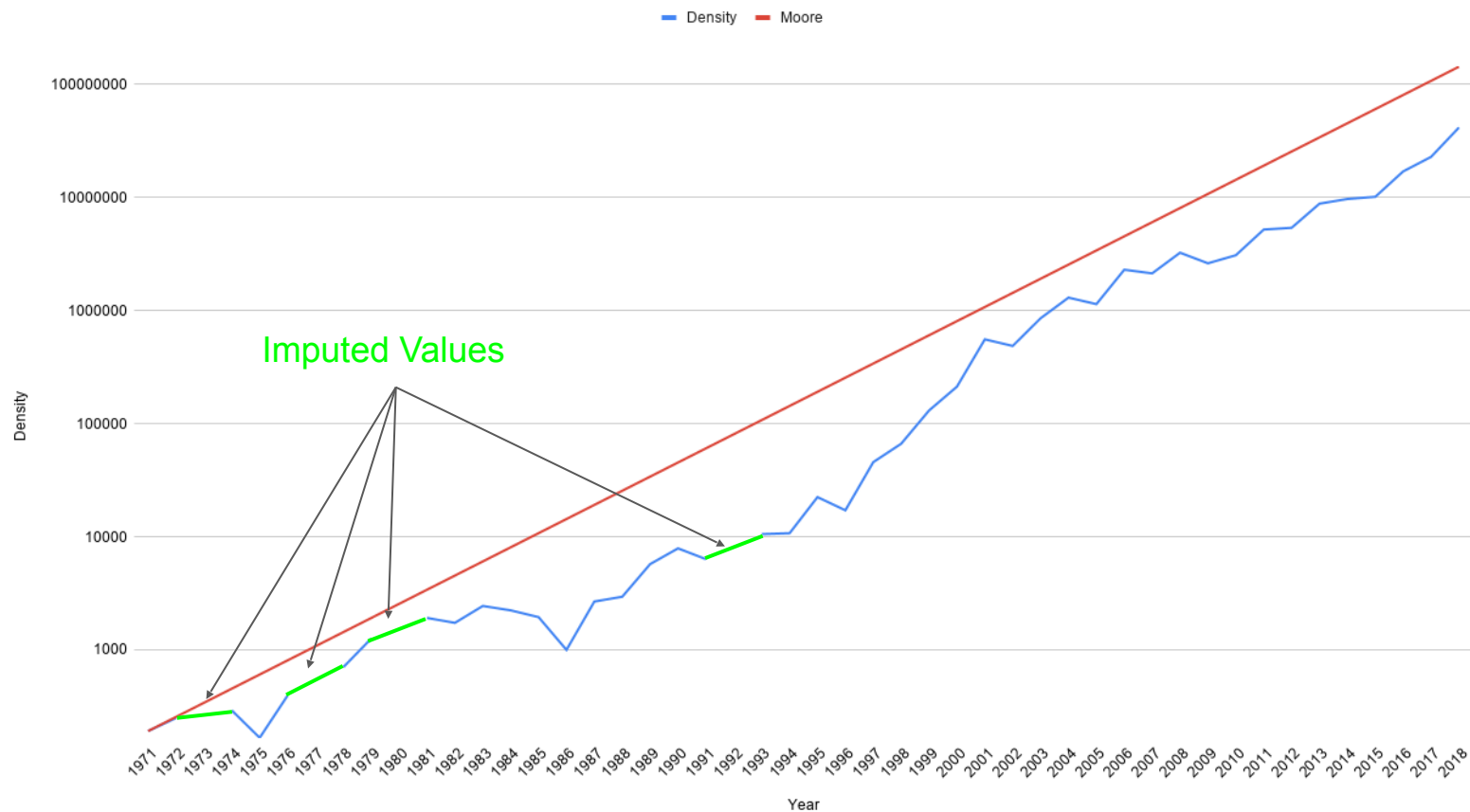




## Density vs. Year



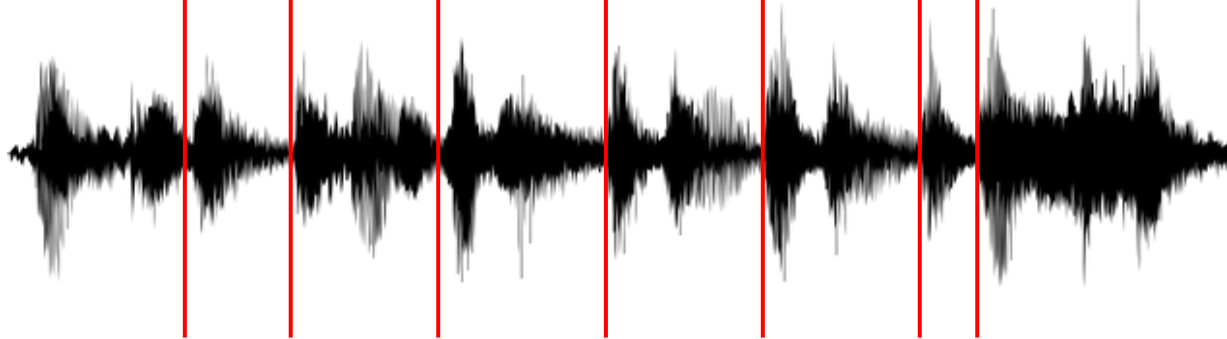
## Density vs. Year



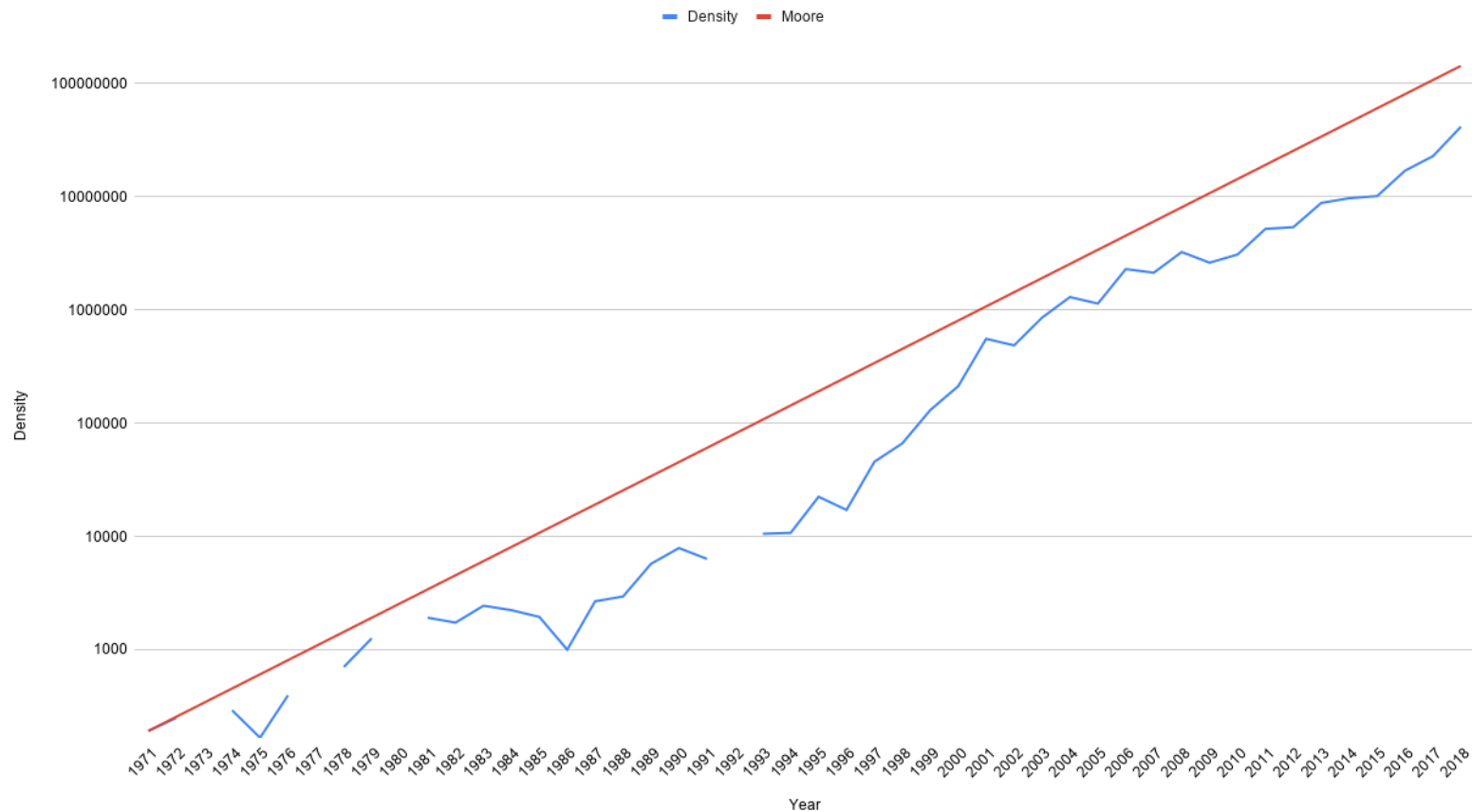


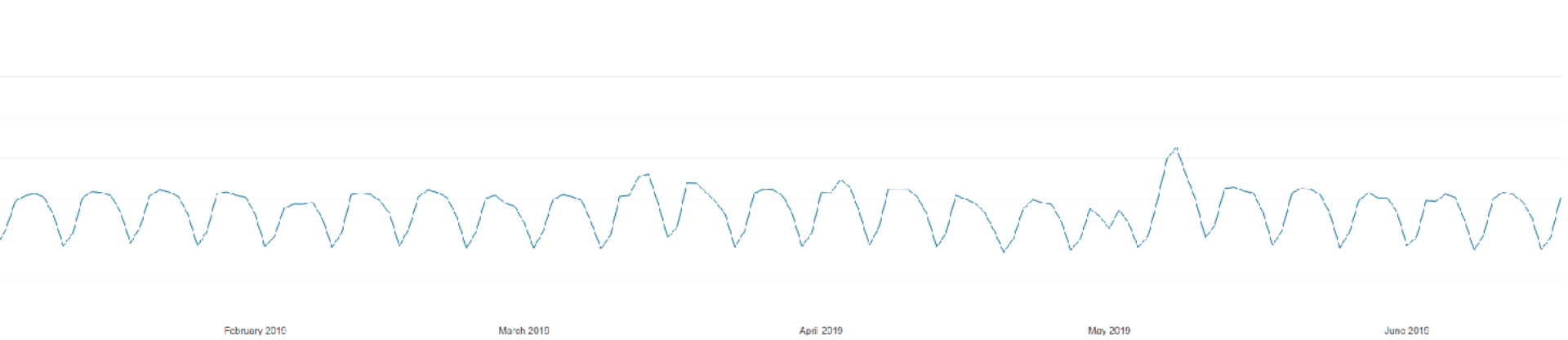


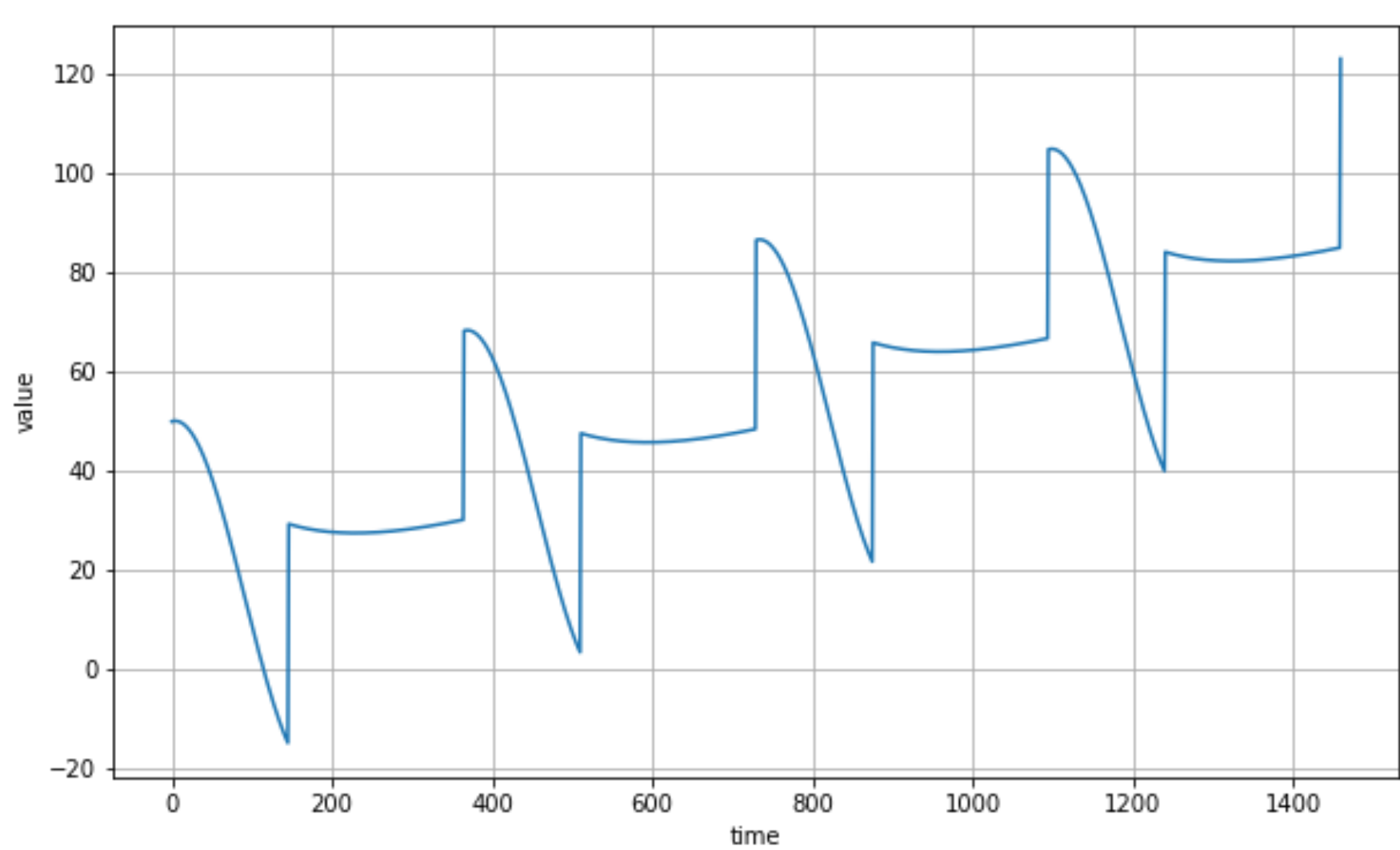
Today we're teaching coding neural layers in Tensorflow

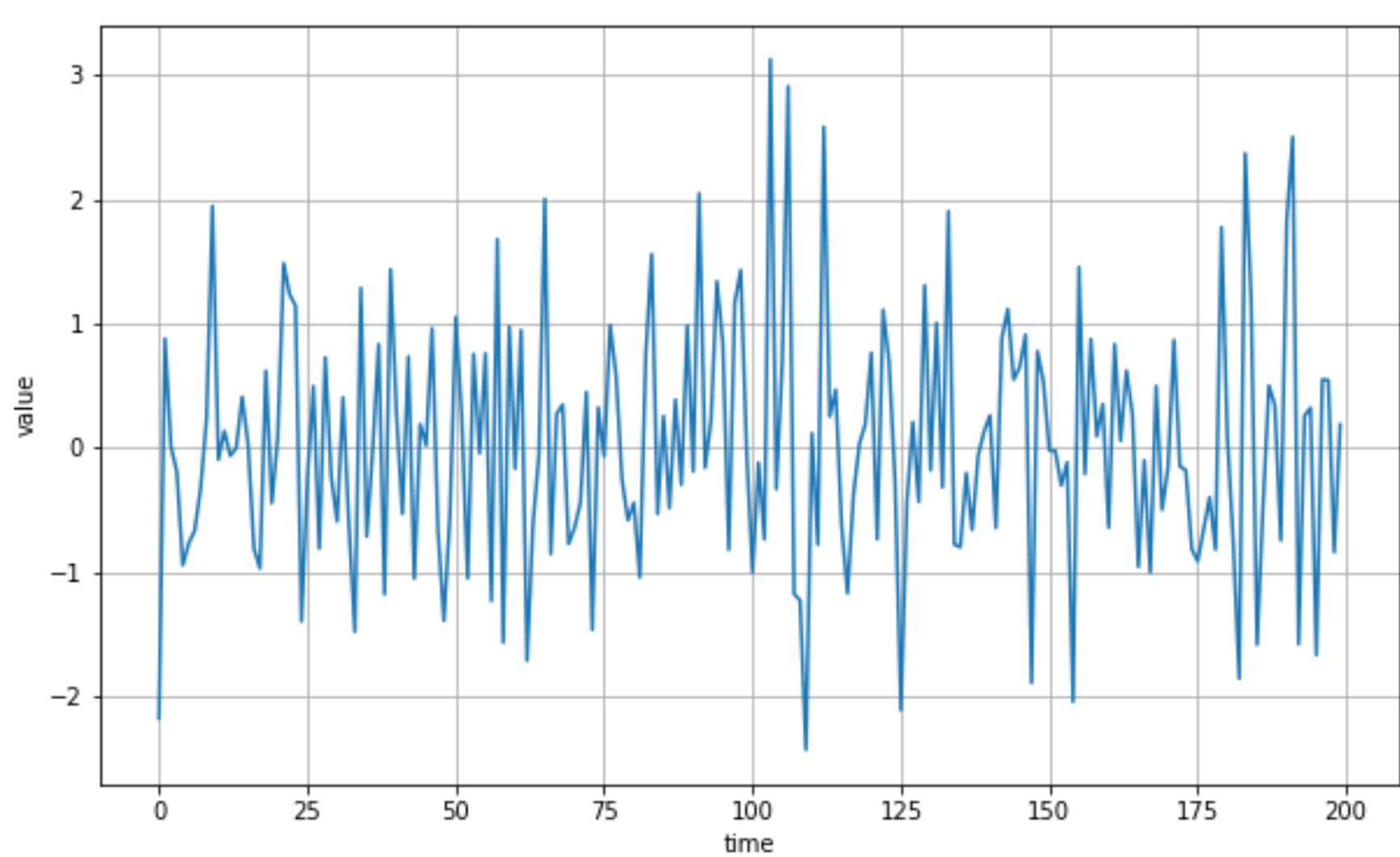


## Density vs. Year

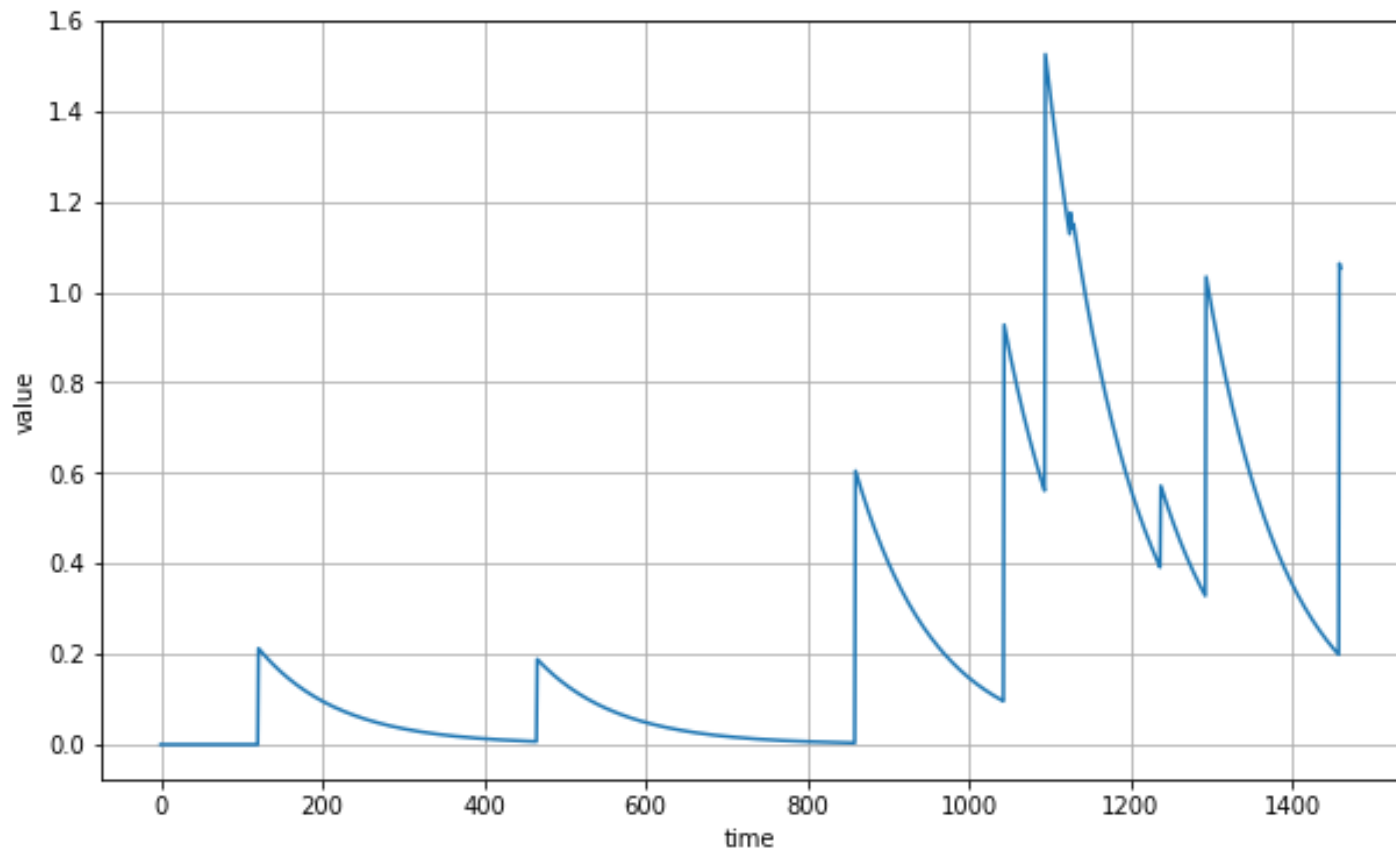




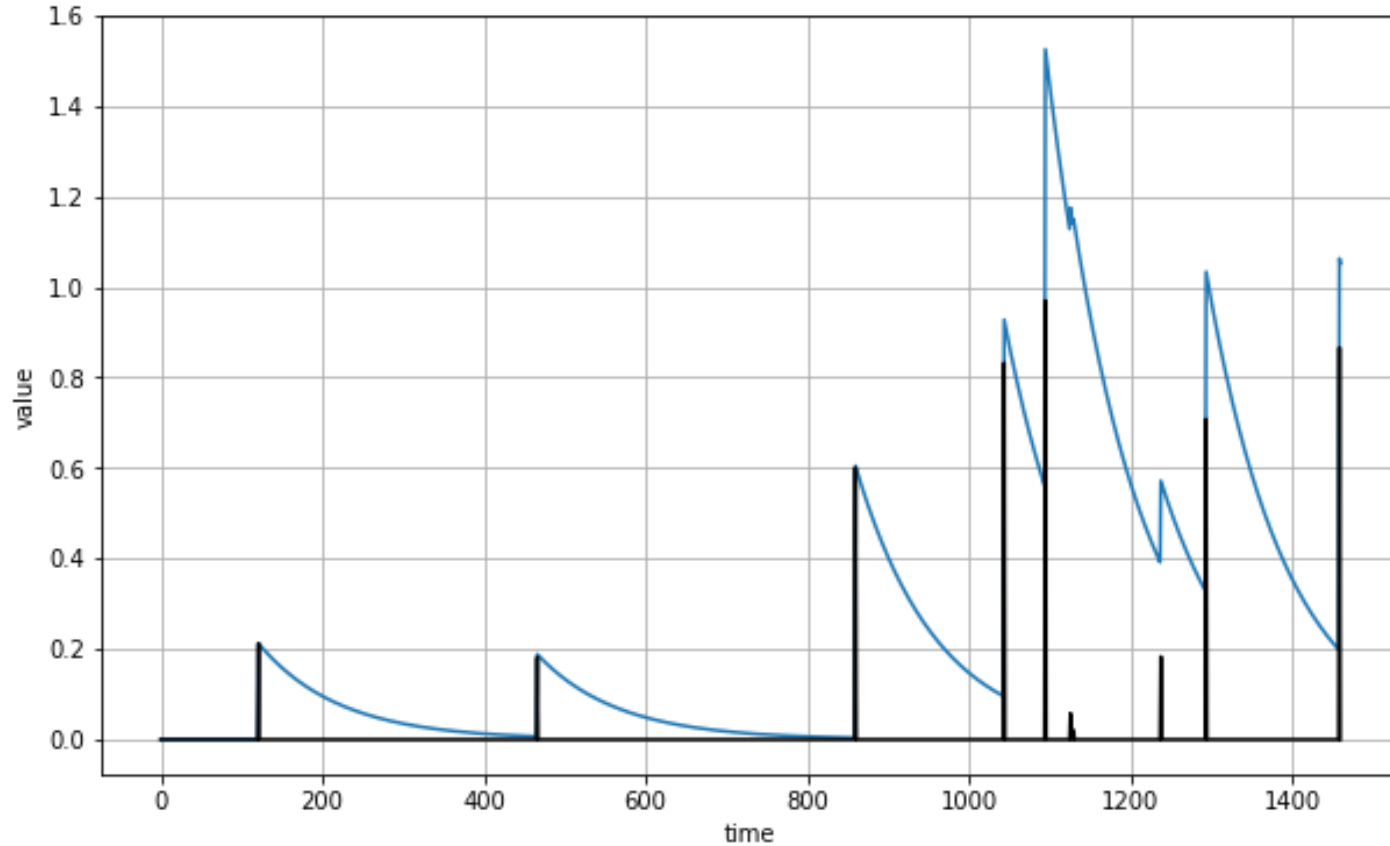




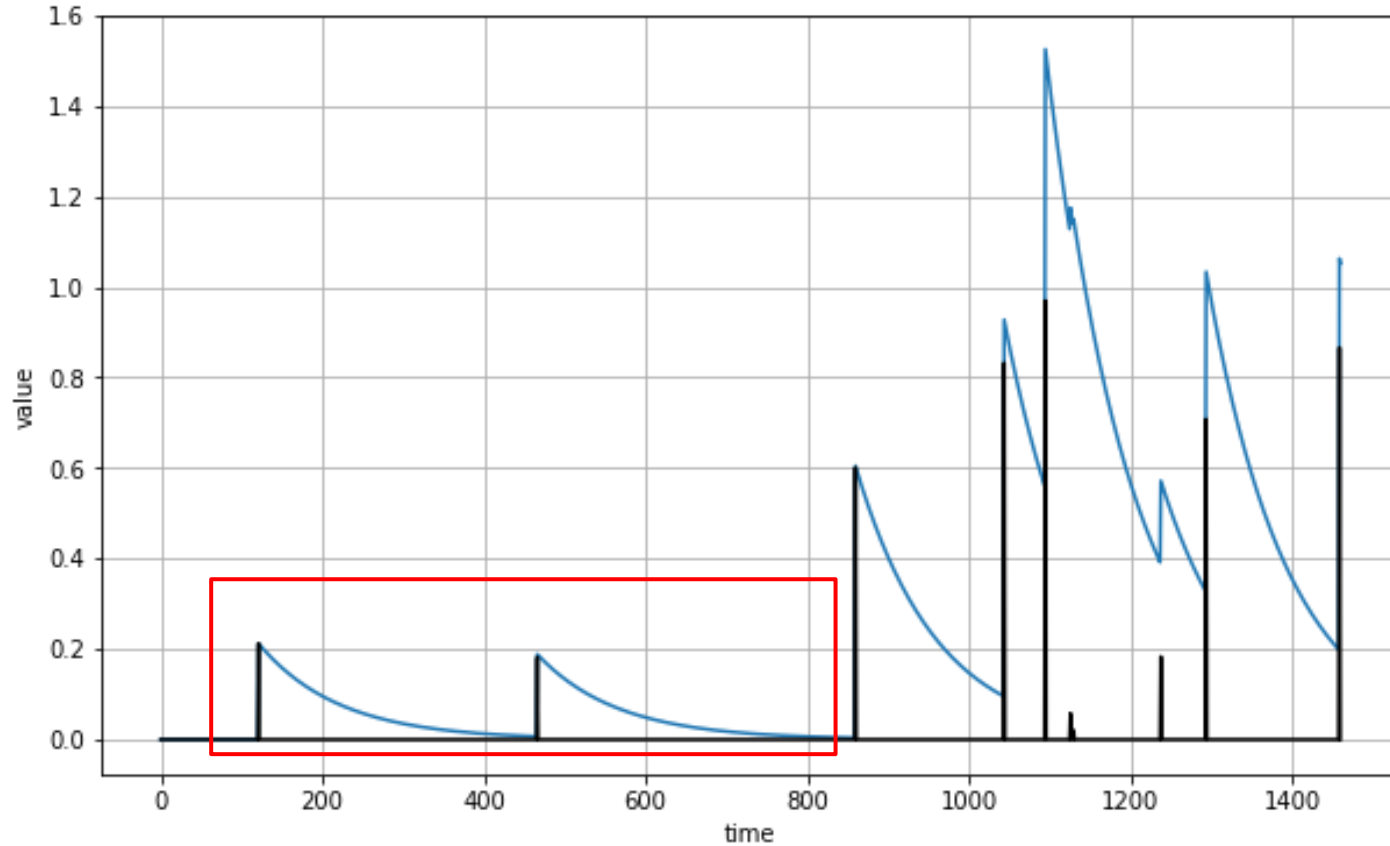
# Autocorrelation



$$v(t) = 0.99 \times v(t-1) + \text{occasional spike}$$

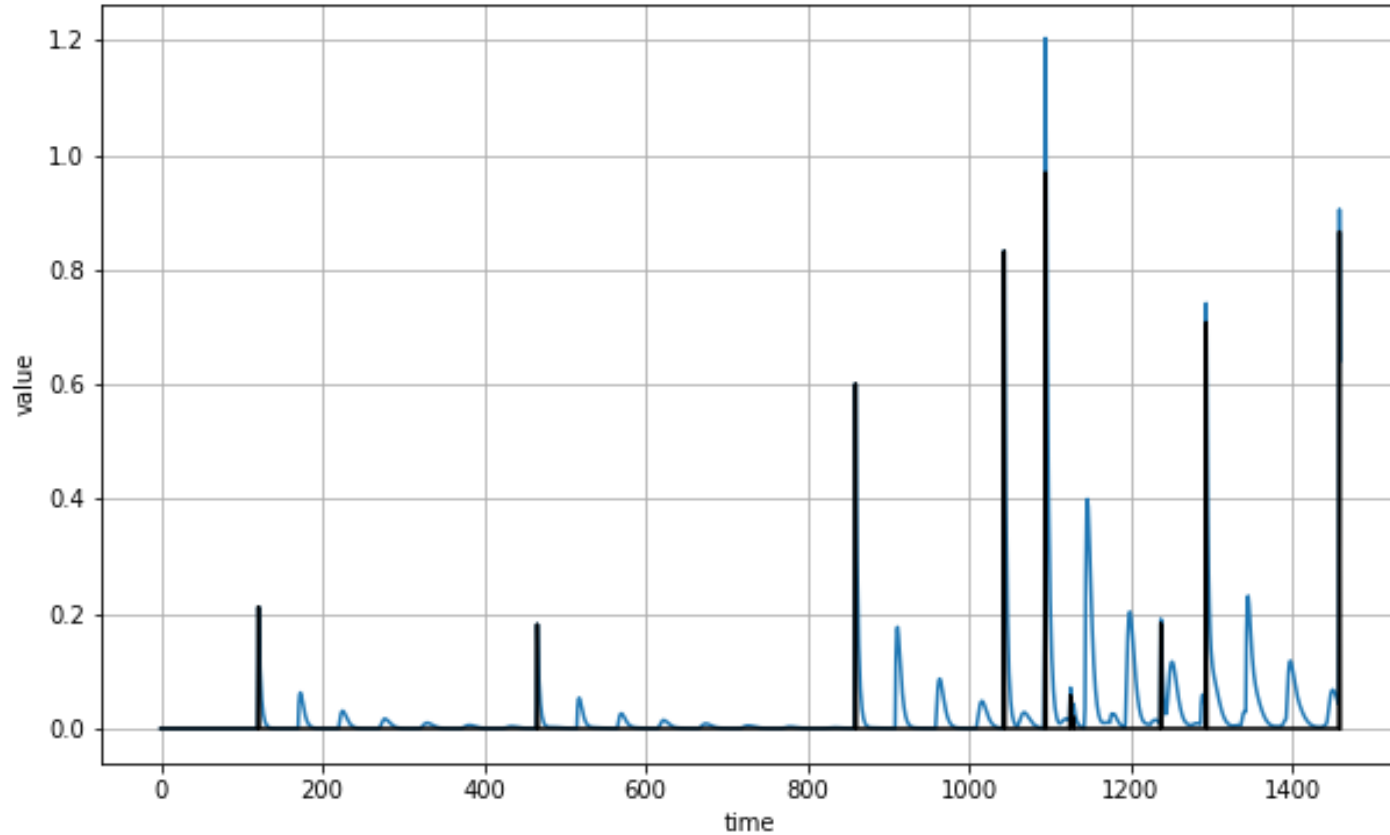


$$v(t) = 0.99 \times v(t-1) + \text{occasional spike}$$

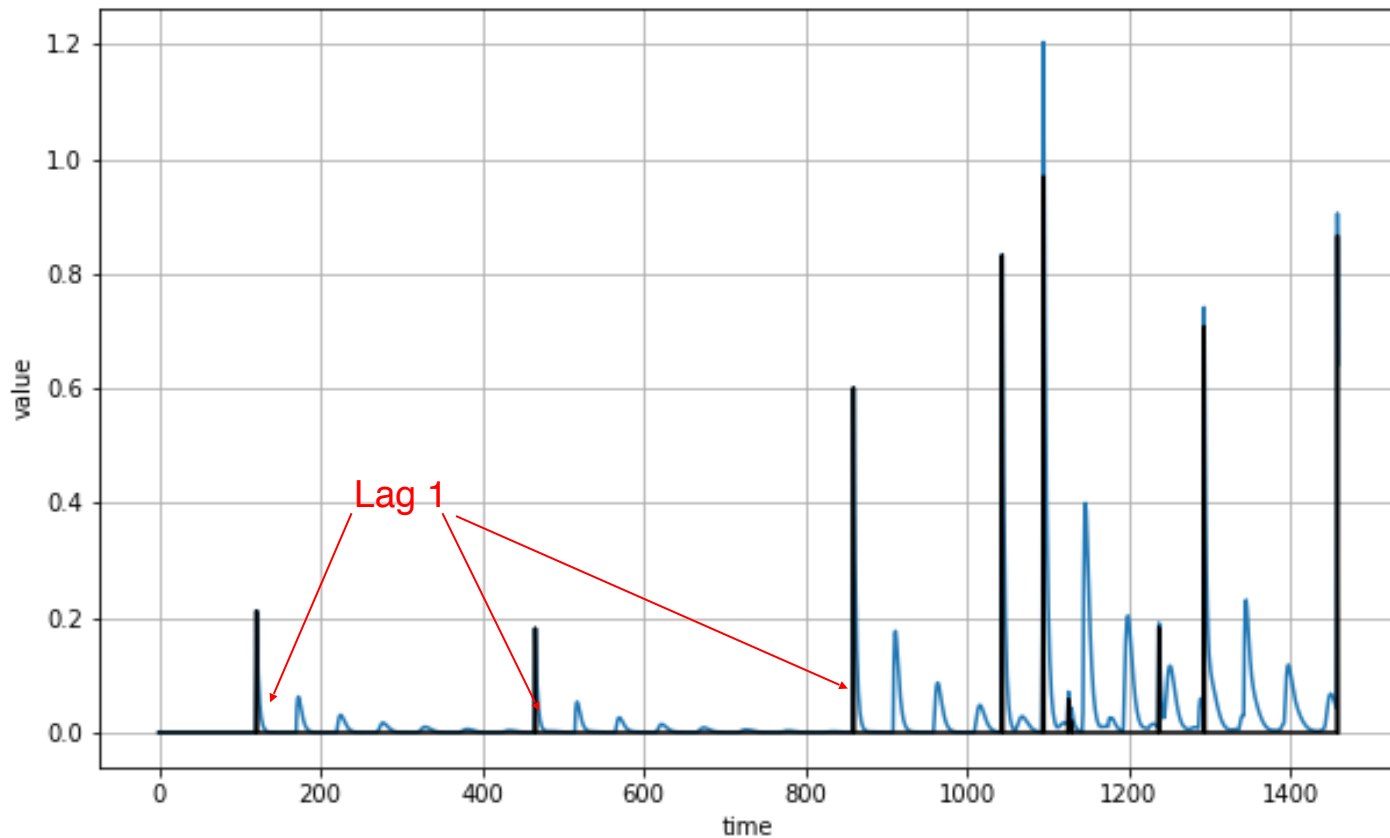




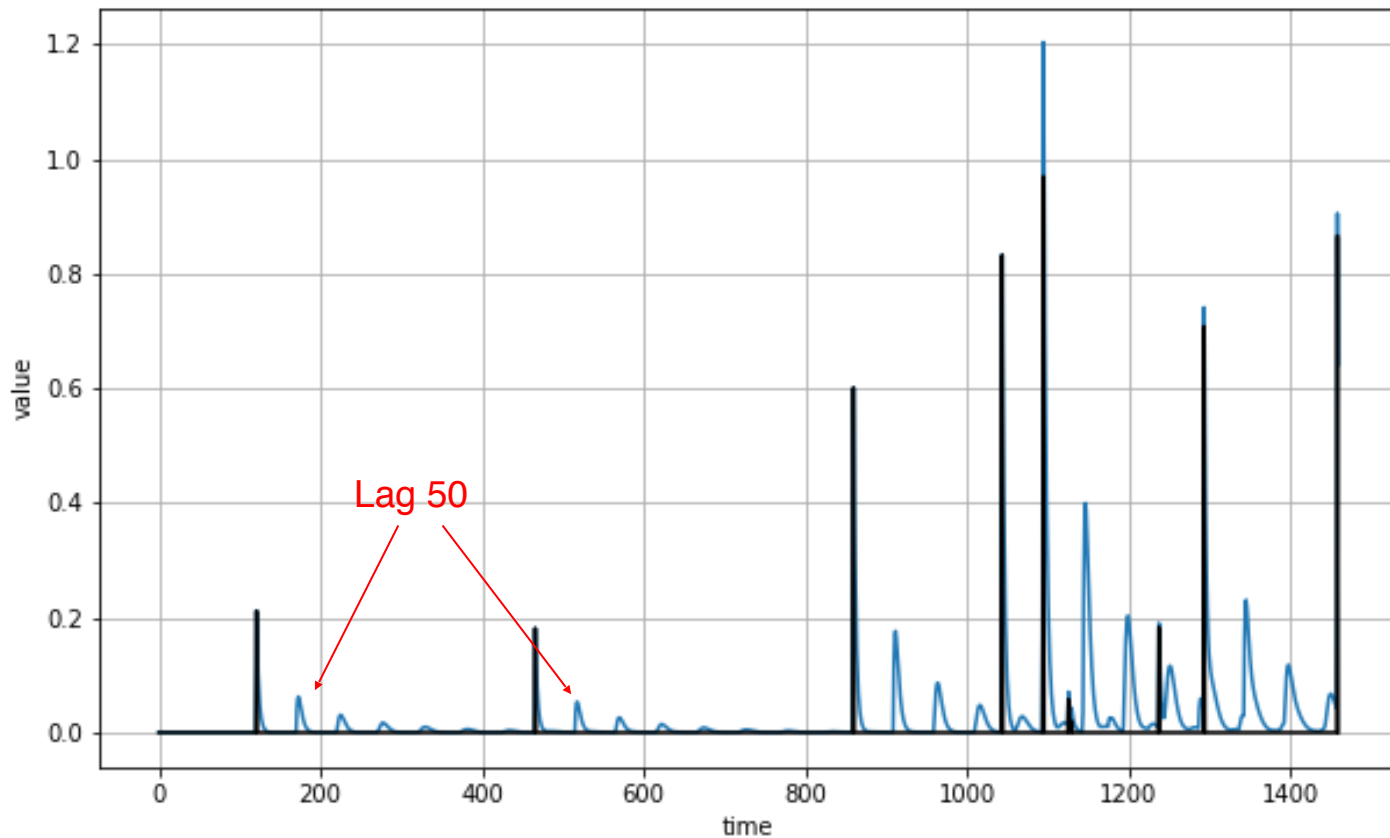
$$v(t) = 0.7 \times v(t-1) + 0.2 \times v(t-50) + \text{occasional spike}$$



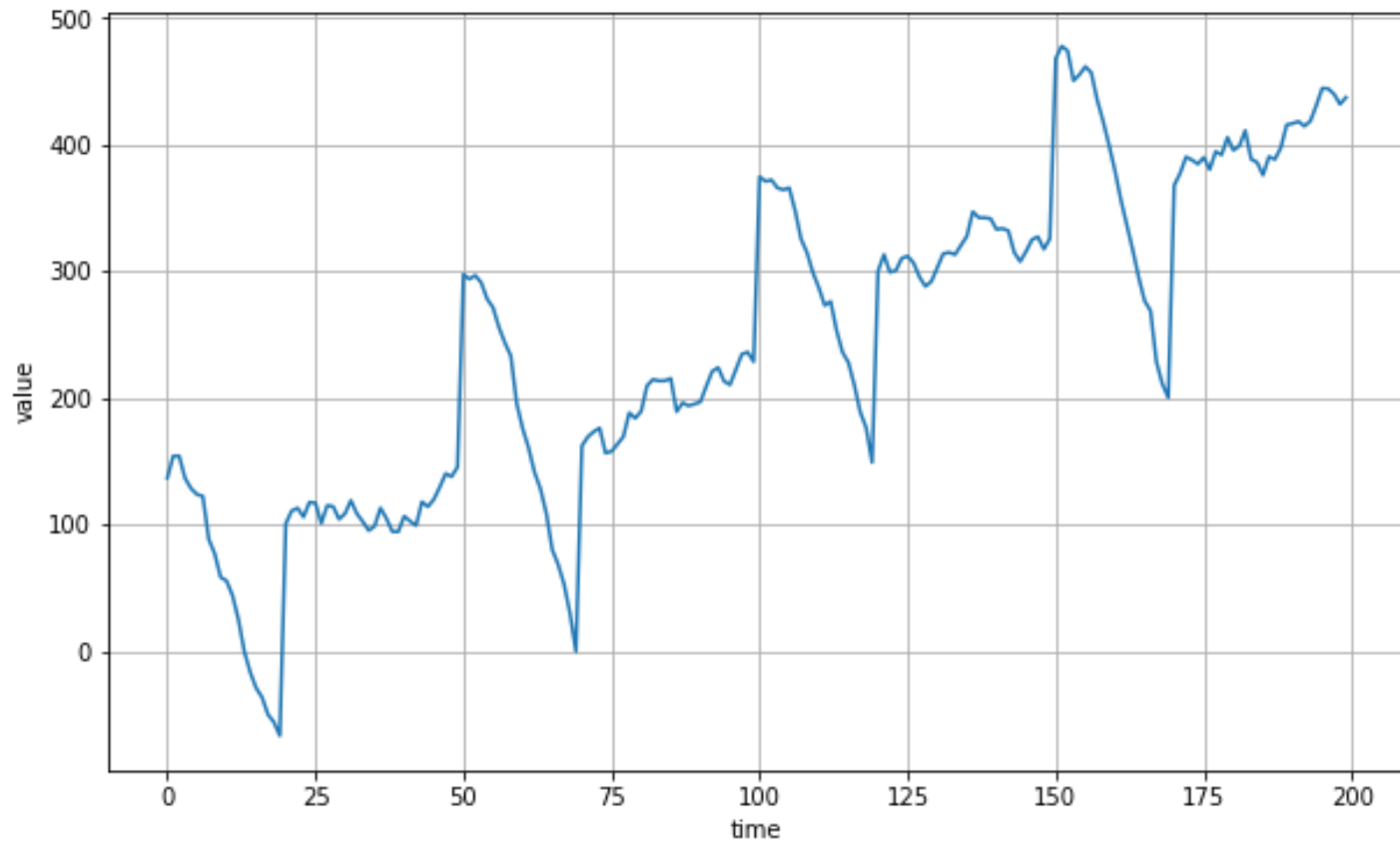
$$v(t) = 0.7 \times v(t-1) + 0.2 \times v(t-50) + \text{occasional spike}$$



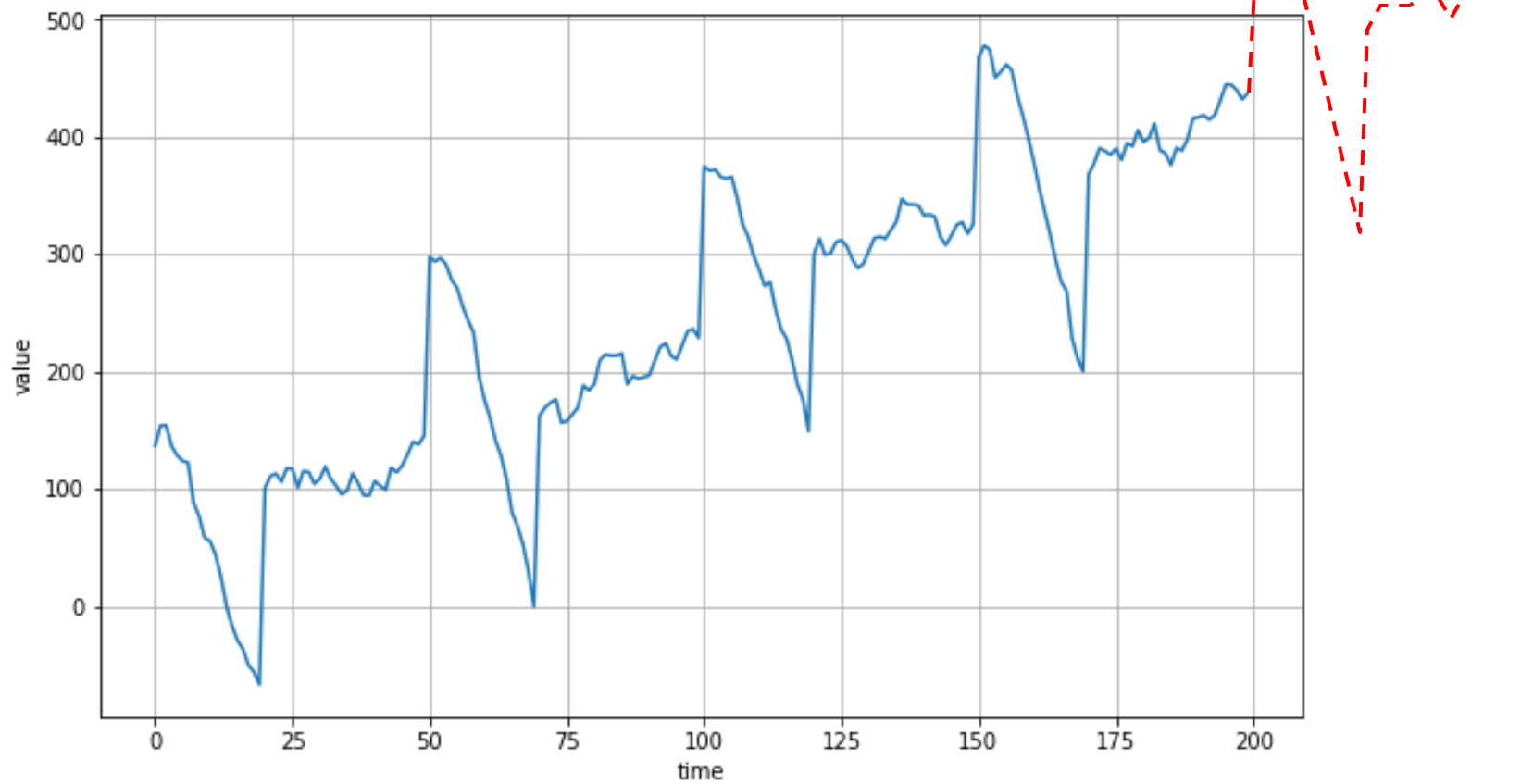
$$v(t) = 0.7 \times v(t-1) + 0.2 \times v(t-50) + \text{occasional spike}$$



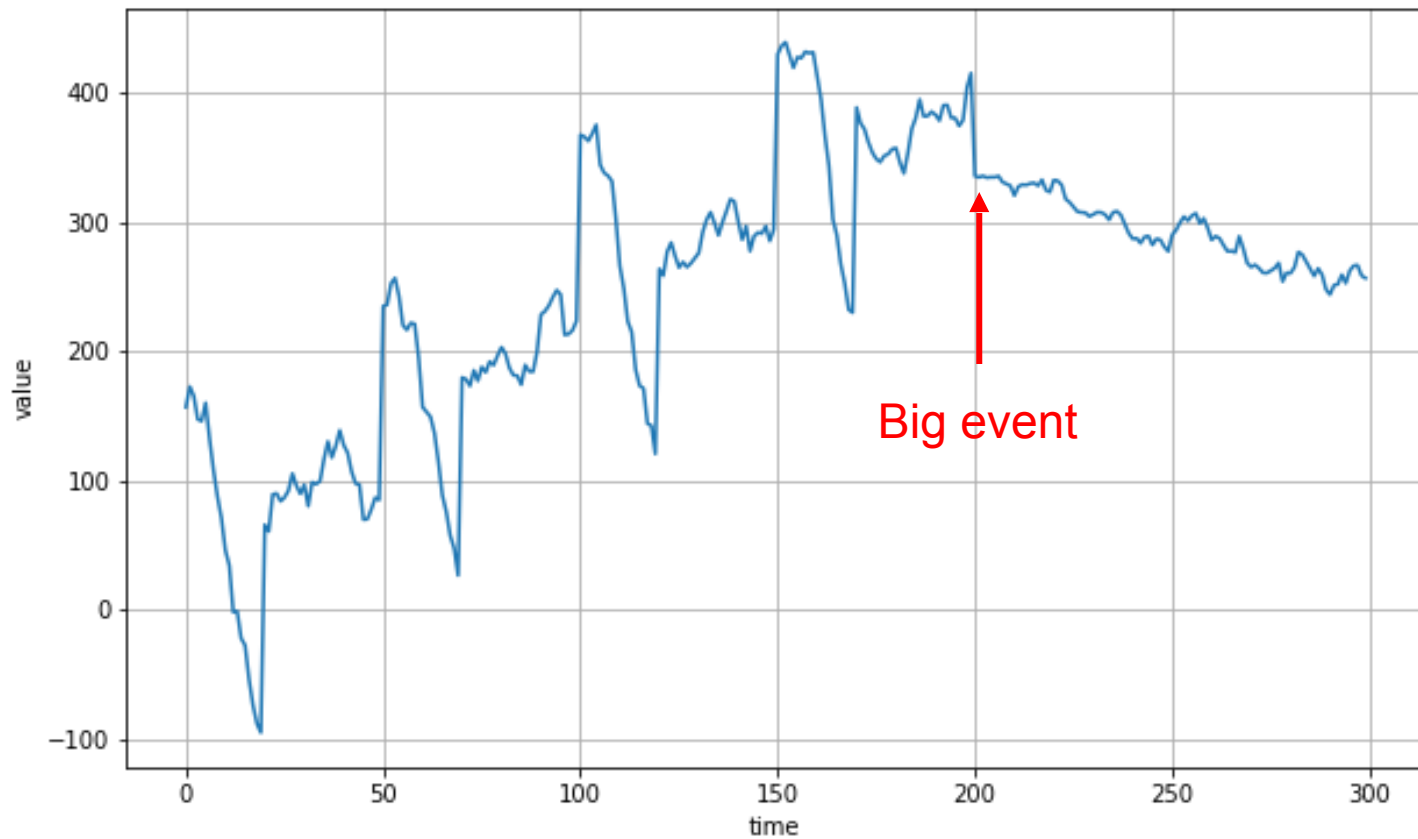
# Trend + Seasonality + Autocorrelation + Noise



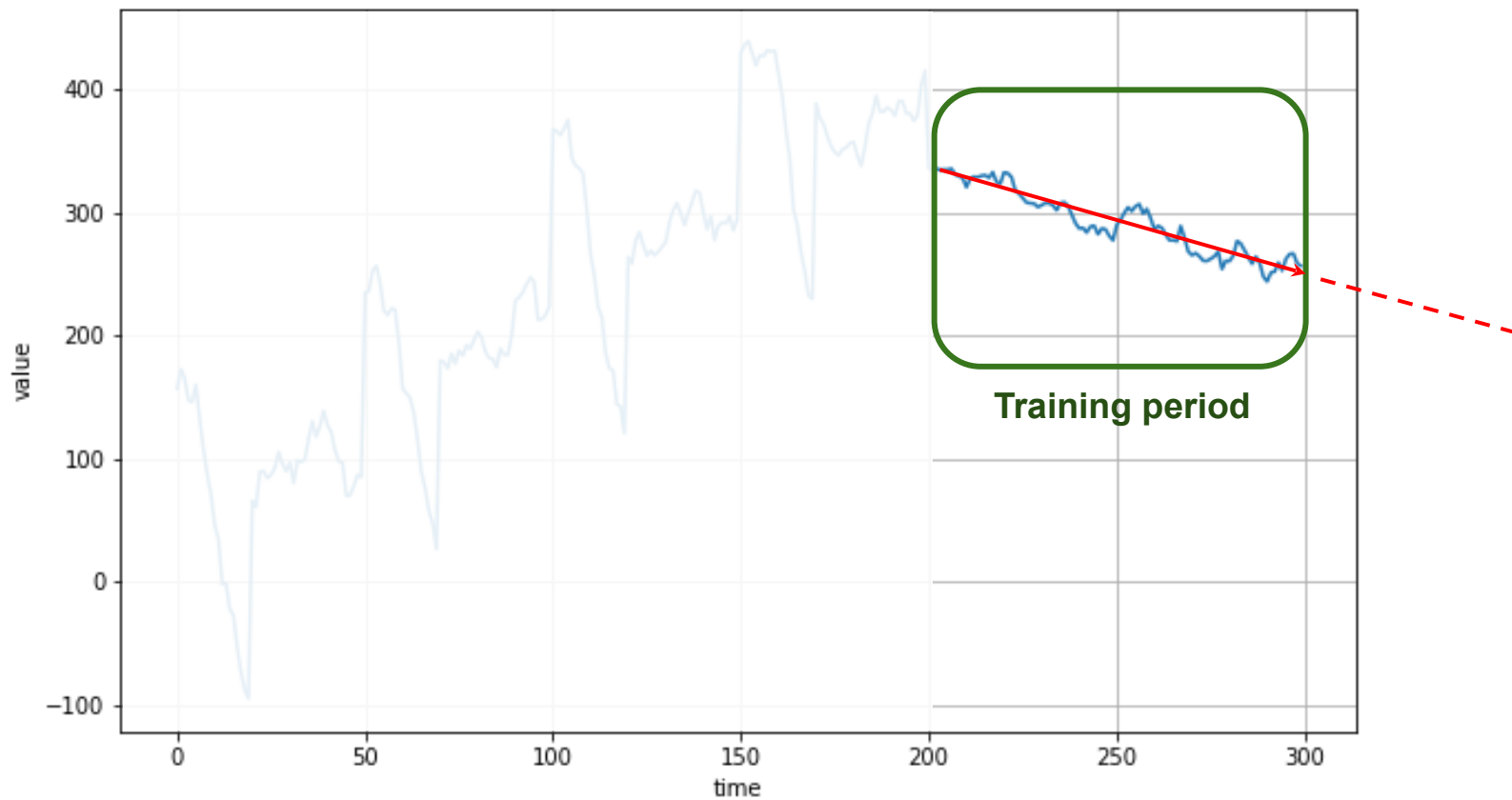
# Forecast Learned Patterns



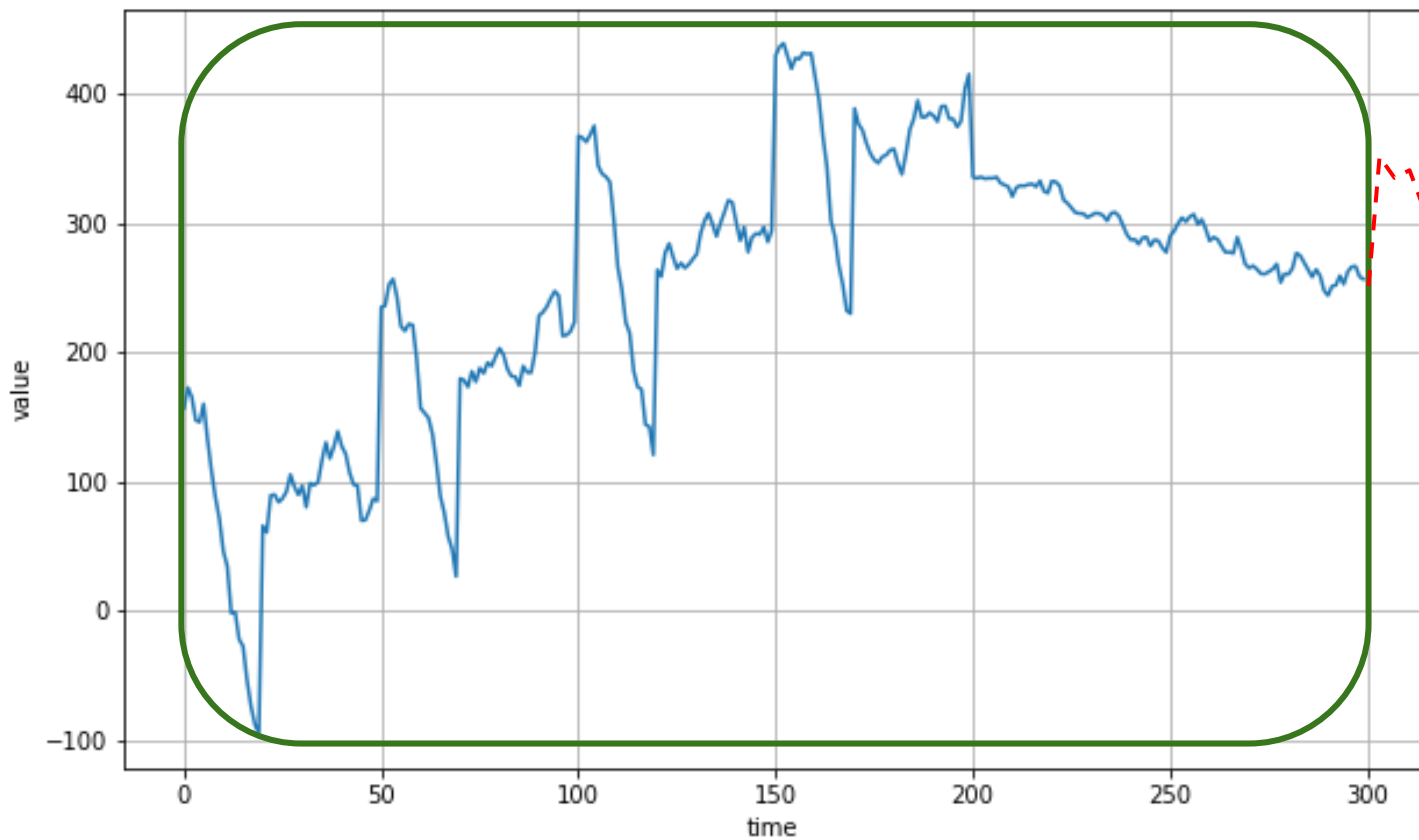
# Non-Stationary Time Series



# Non-Stationary Time Series

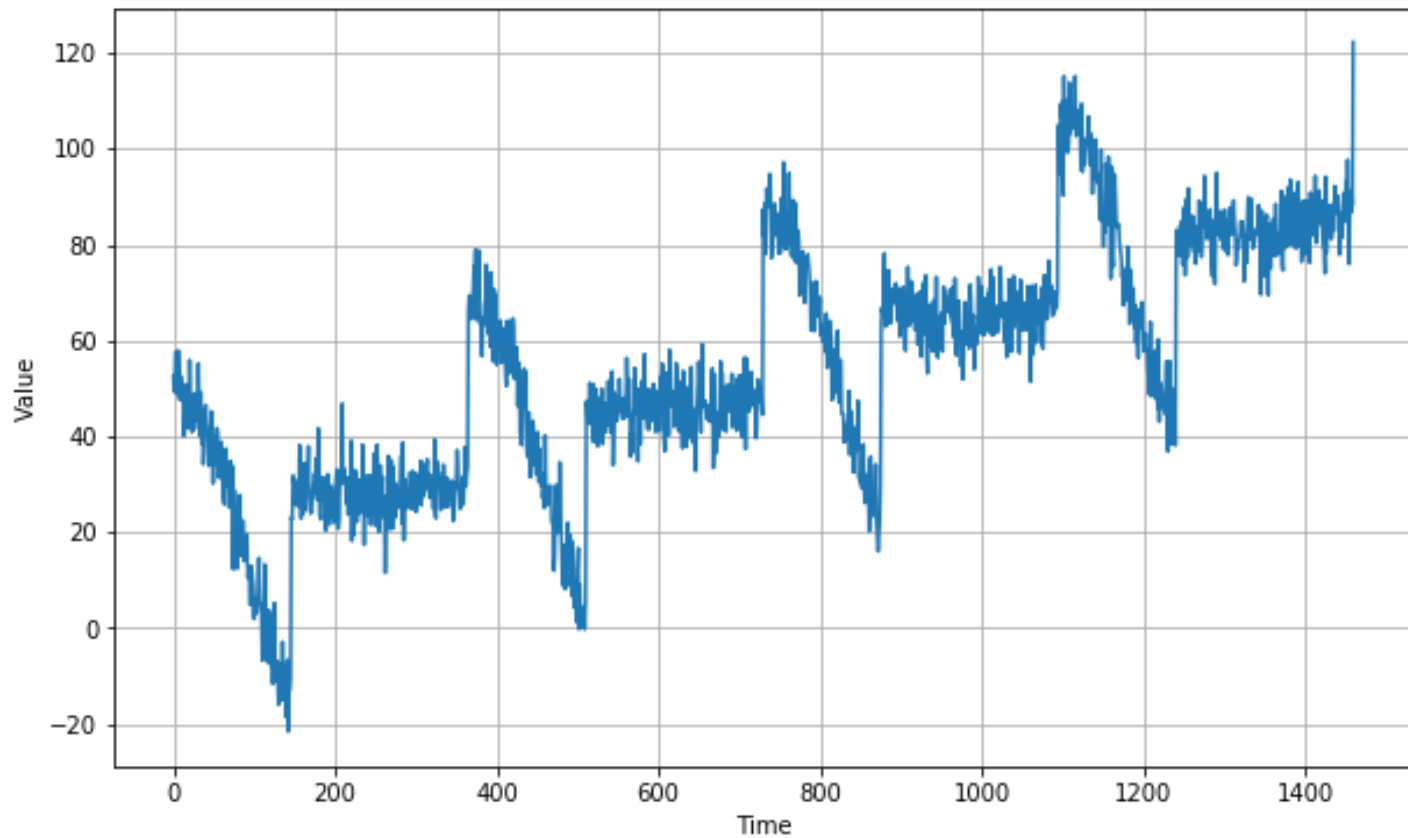


# Non-Stationary Time Series

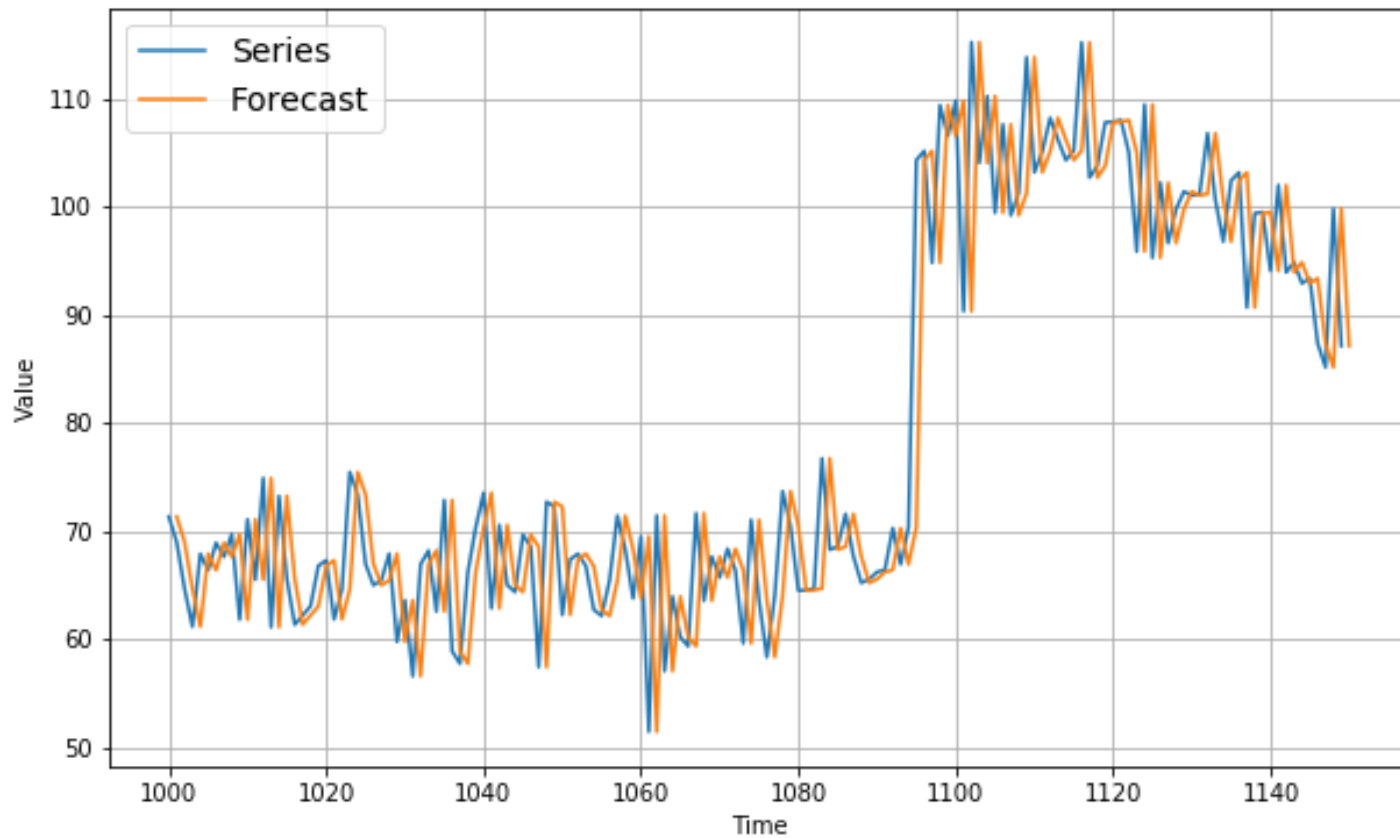




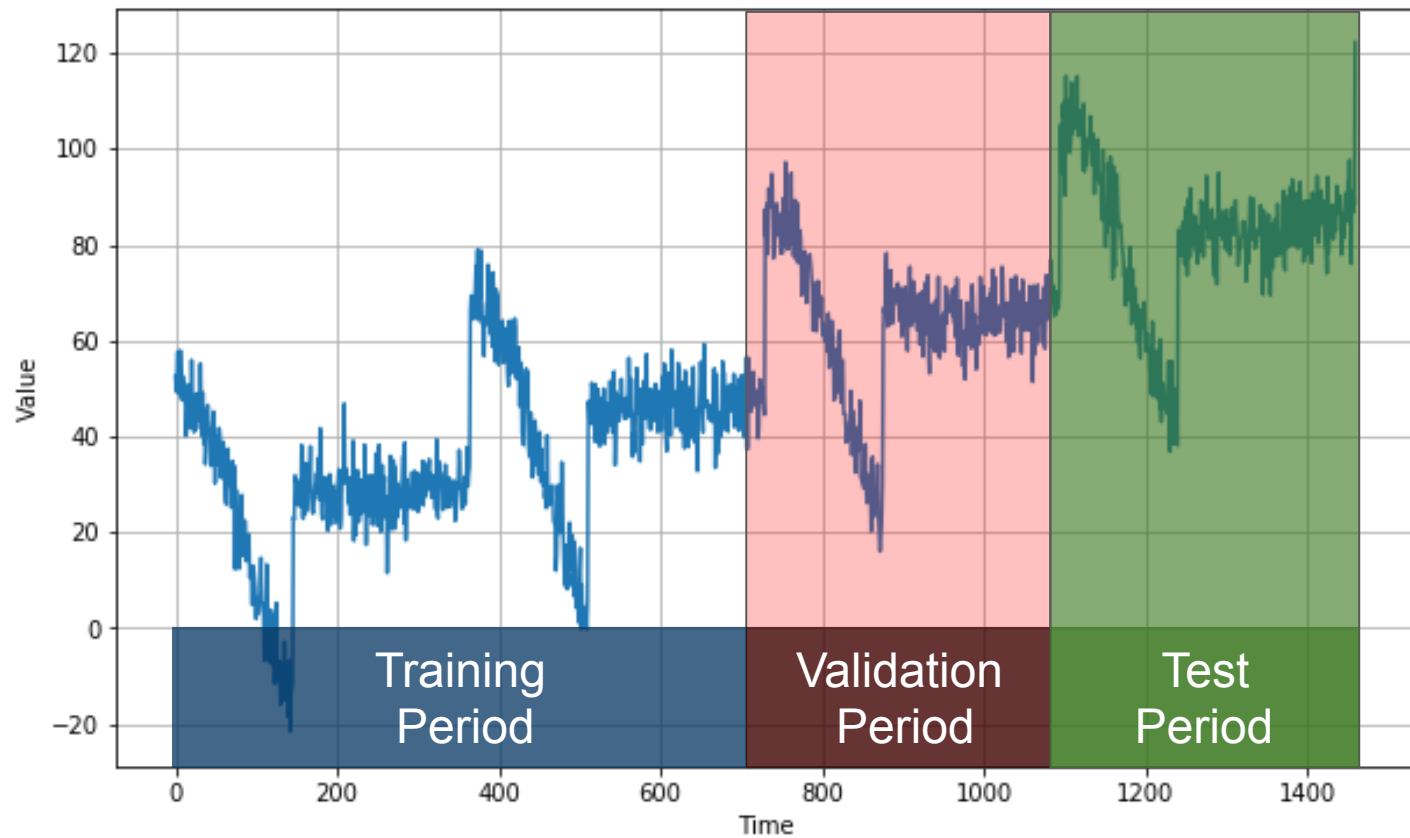
# Trend + Seasonality + Noise



# Naive Forecasting



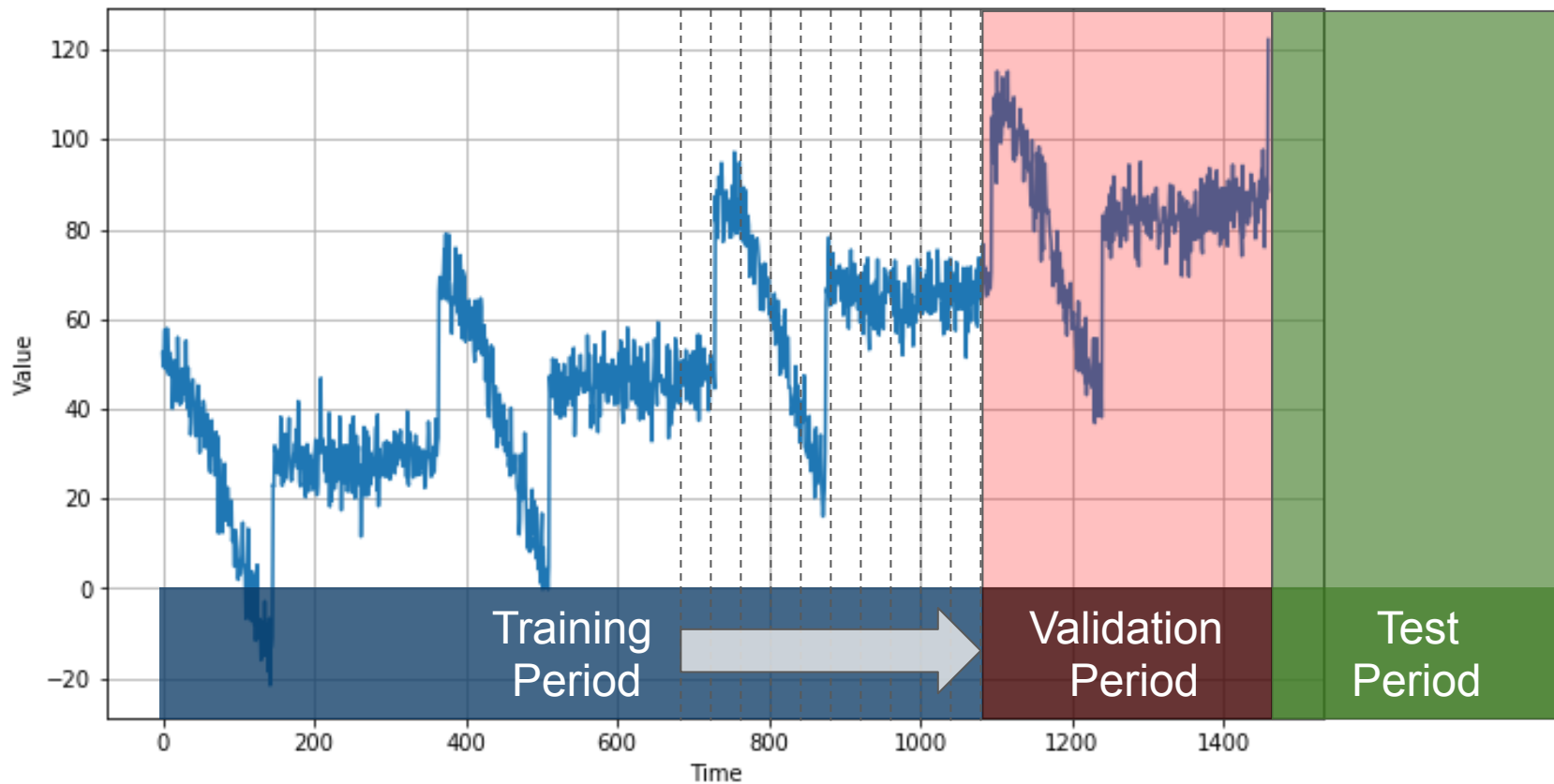
# Fixed Partitioning



# Fixed Partitioning



# Roll-Forward Partitioning



## Metrics

```
errors = forecasts - actual
```

```
mse = np.square(errors).mean()
```

```
rmse = np.sqrt(mse)
```

```
mae = np.abs(errors).mean()
```

```
mape = np.abs(errors / x_valid).mean()
```

## Metrics

```
errors = forecasts - actual
```

```
mse = np.square(errors).mean()
```

```
rmse = np.sqrt(mse)
```

```
mae = np.abs(errors).mean()
```

```
mape = np.abs(errors / x_valid).mean()
```

## Metrics

```
errors = forecasts - actual
```

```
mse = np.square(errors).mean()
```

```
rmse = np.sqrt(mse)
```

```
mae = np.abs(errors).mean()
```

```
mape = np.abs(errors / x_valid).mean()
```



## Metrics

```
errors = forecasts - actual
```

```
mse = np.square(errors).mean()
```

```
rmse = np.sqrt(mse)
```

```
mae = np.abs(errors).mean()
```

```
mape = np.abs(errors / x_valid).mean()
```

## Metrics

```
errors = forecasts - actual
```

```
mse = np.square(errors).mean()
```

```
rmse = np.sqrt(mse)
```

```
mae = np.abs(errors).mean()
```

```
mape = np.abs(errors / x_valid).mean()
```

## Metrics

```
errors = forecasts - actual
```

```
mse = np.square(errors).mean()
```

```
rmse = np.sqrt(mse)
```

```
mae = np.abs(errors).mean()
```

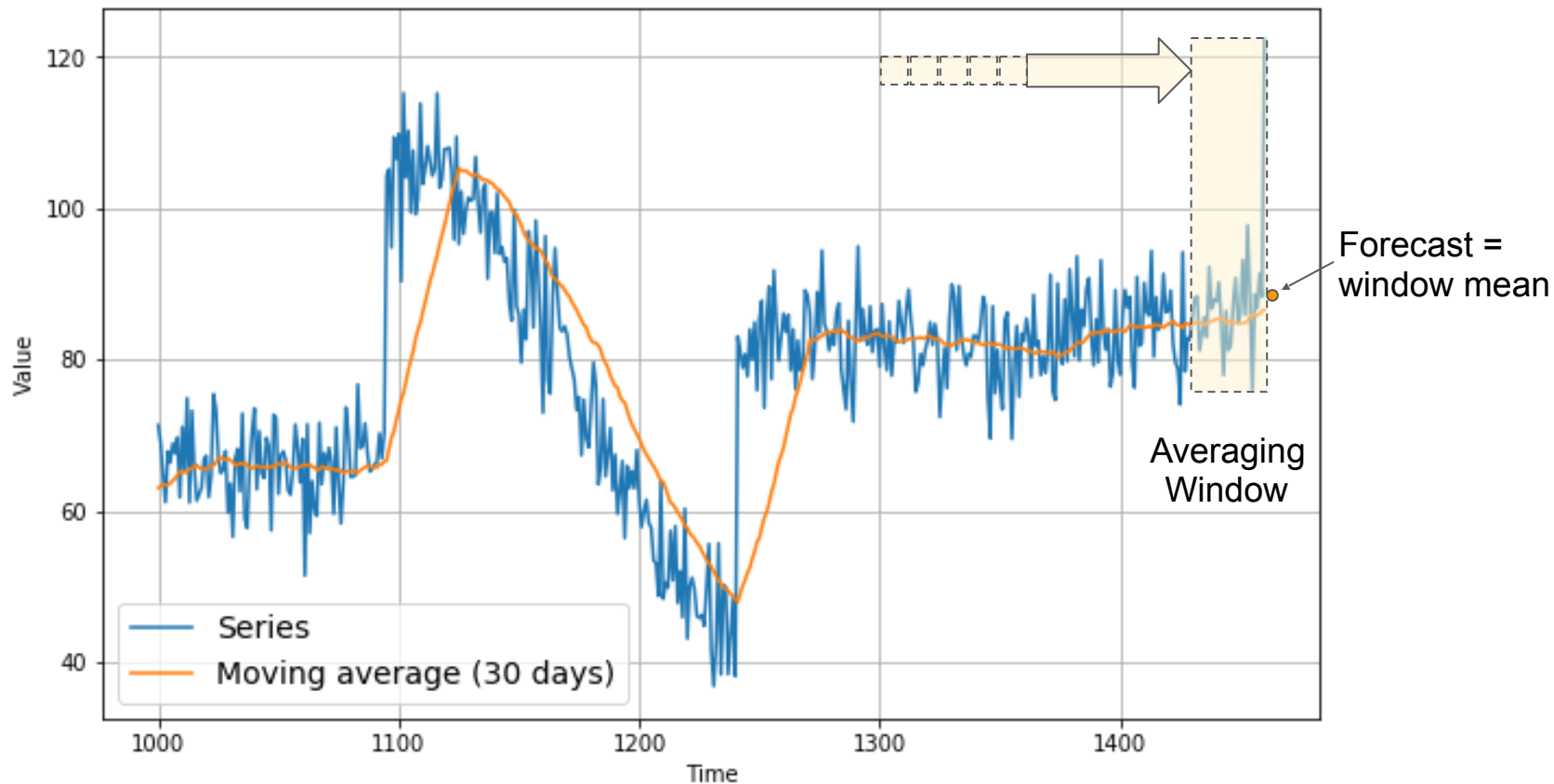
```
mape = np.abs(errors / x_valid).mean()
```

# Naive Forecast MAE

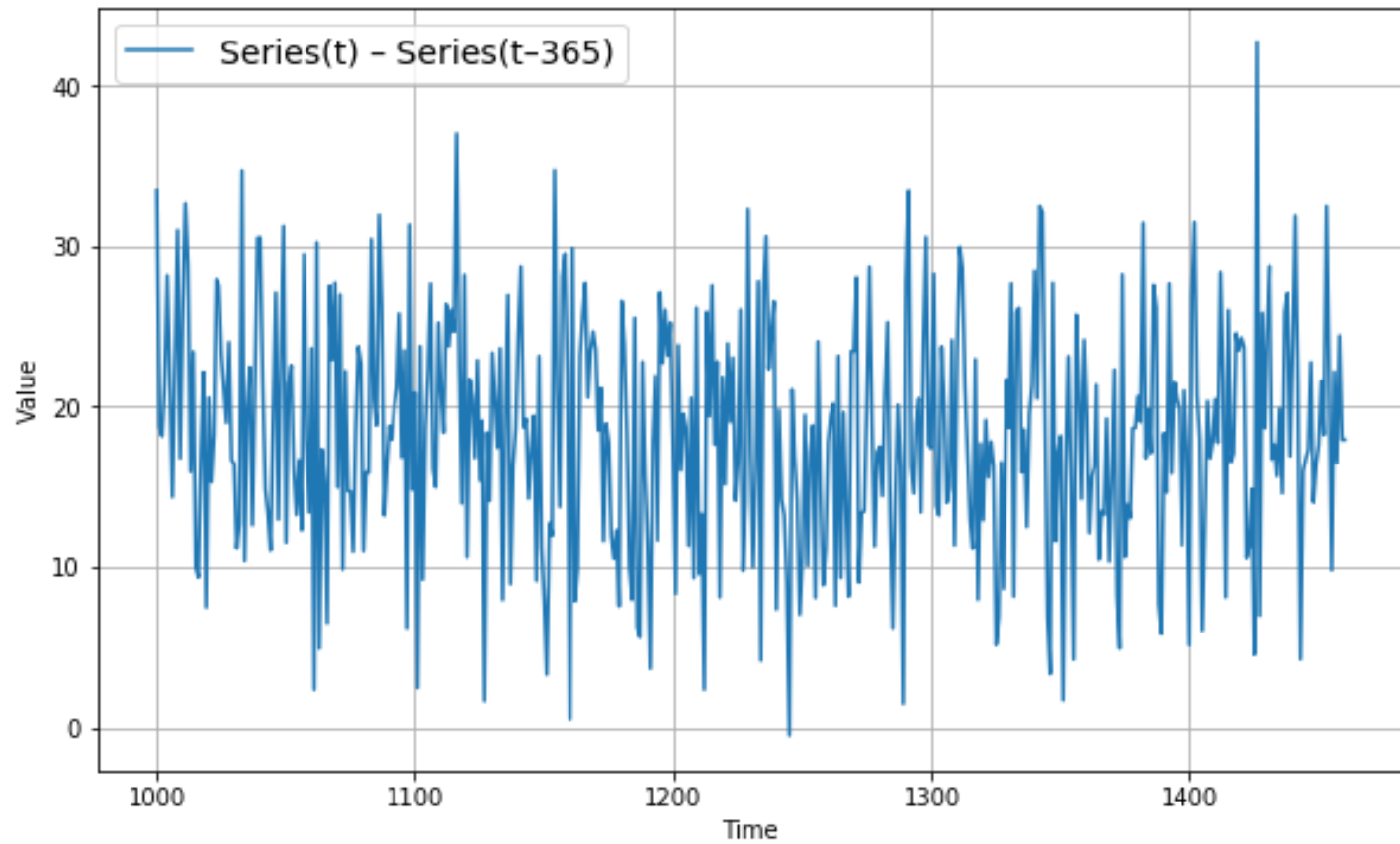
```
keras.metrics.mean_absolute_error(x_valid, naive_forecast).numpy()
```

```
5.937908515321673
```

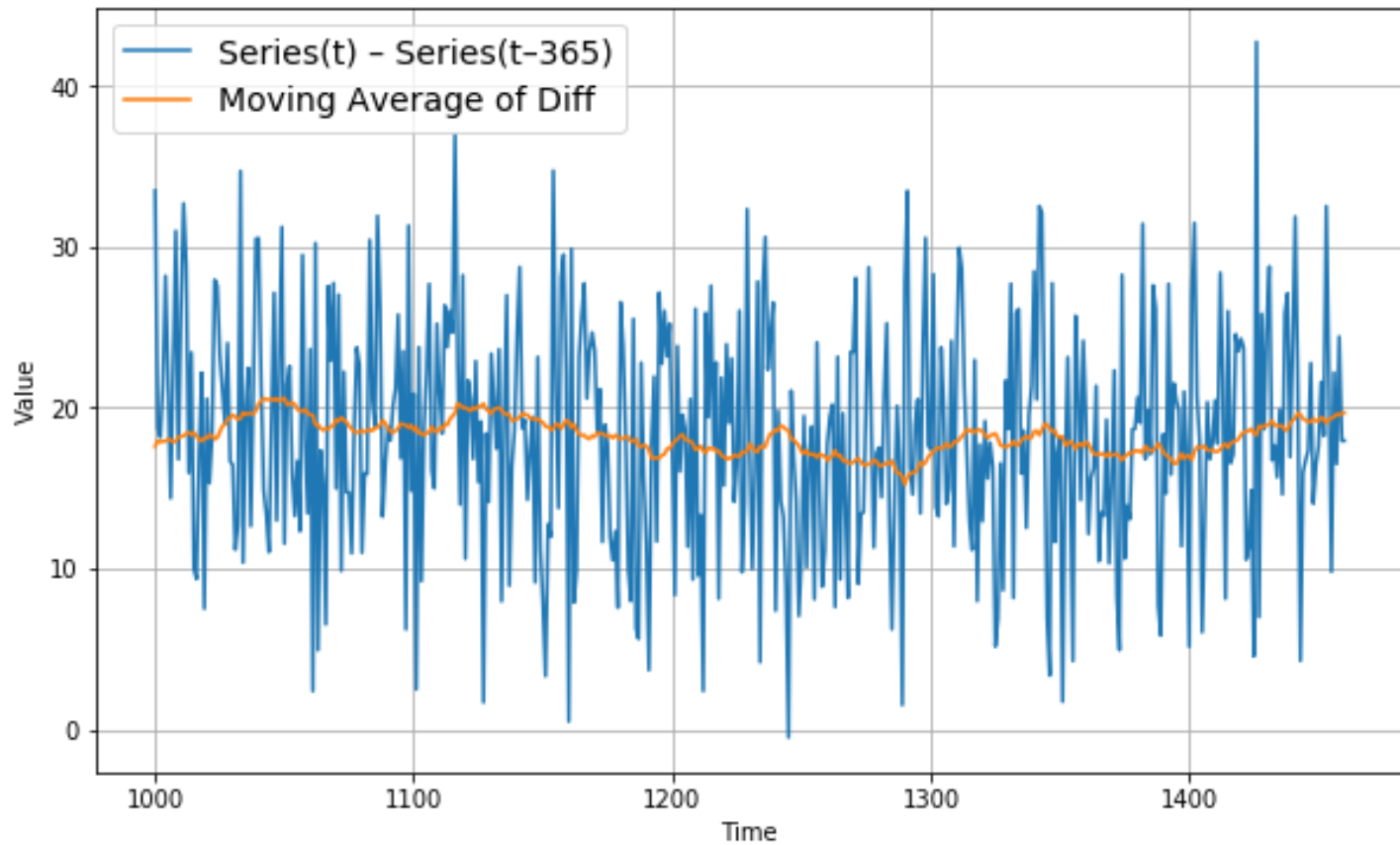
# Moving Average



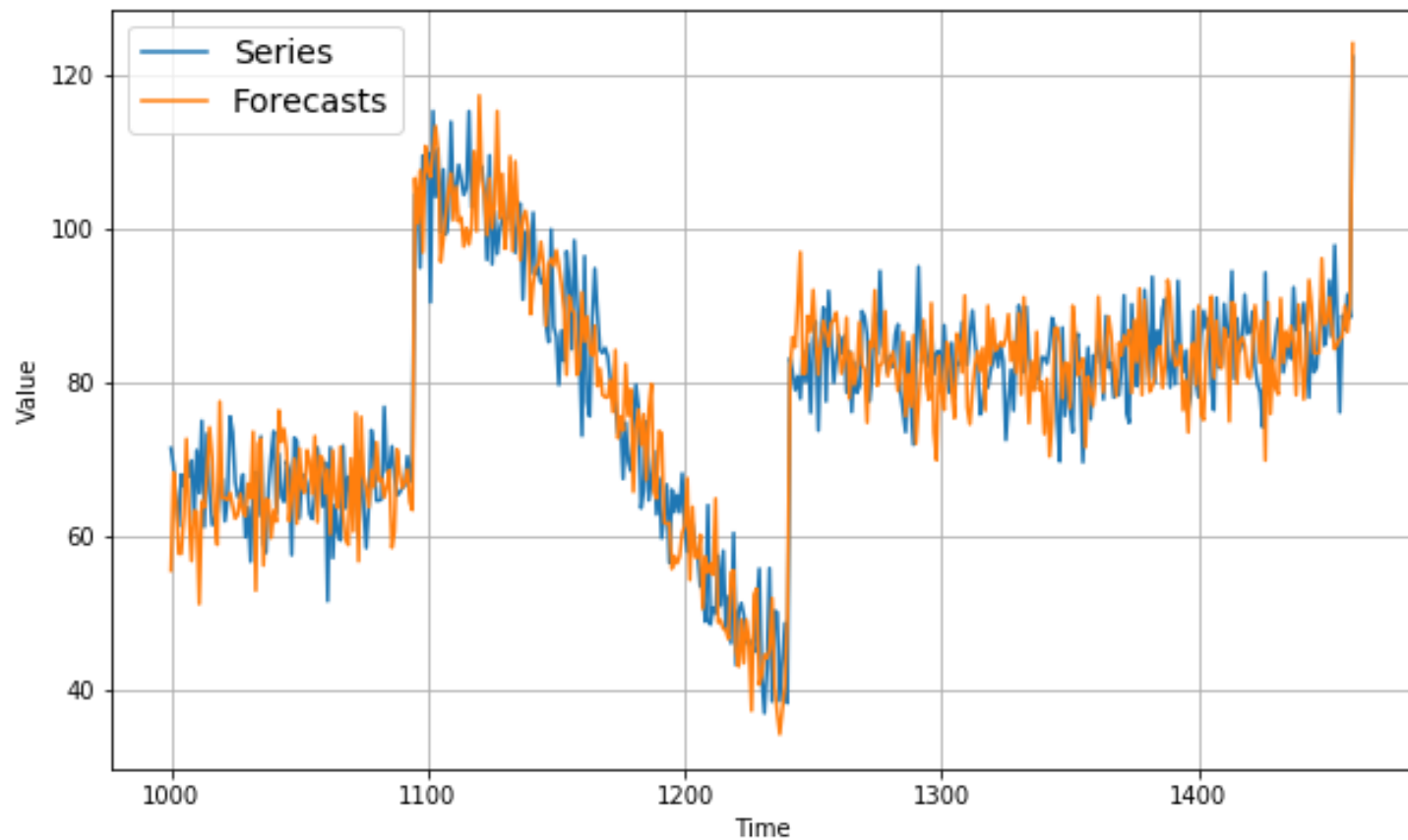
# Differencing



# Moving Average on Differenced Time Series



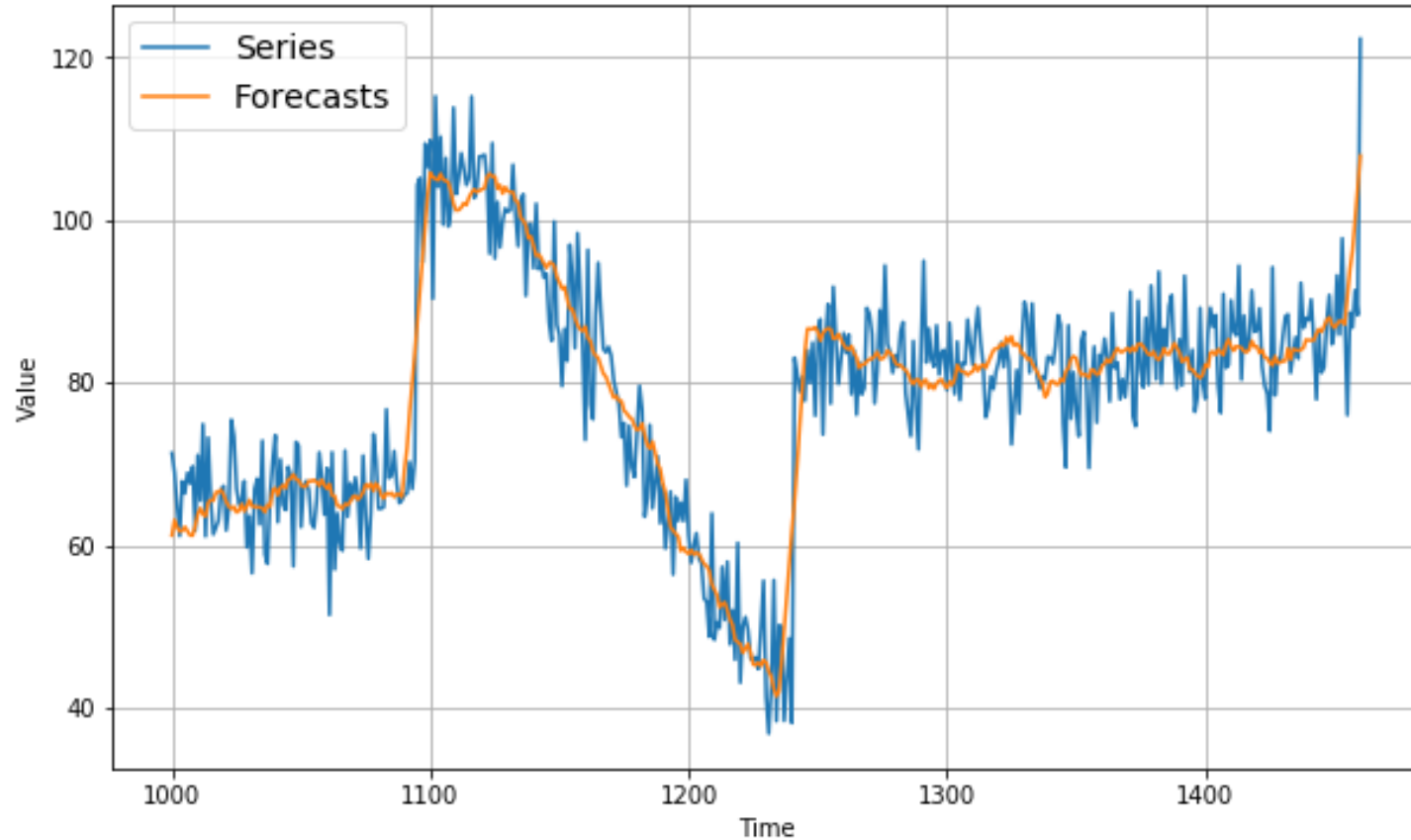
# Restoring the Trend and Seasonality



Forecasts = moving average of differenced series + series( $t - 365$ )

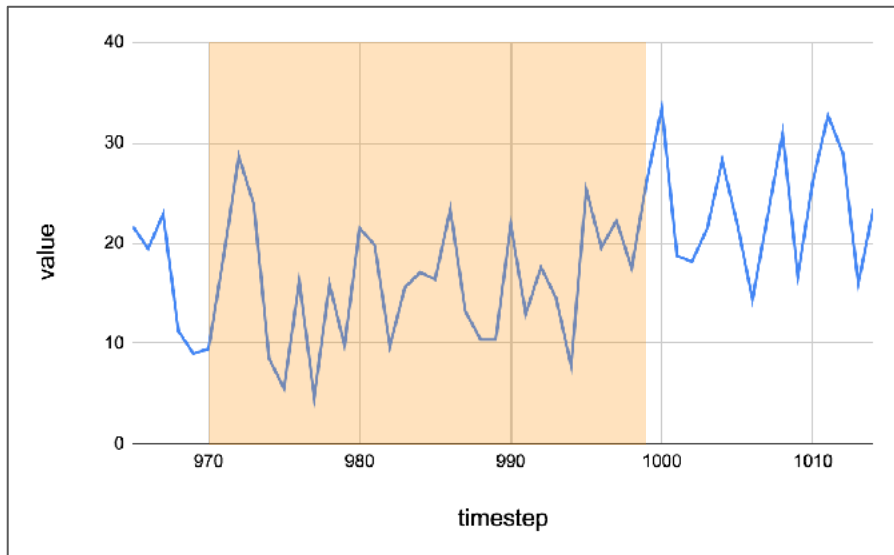


# Smoothing Both Past and Present Values

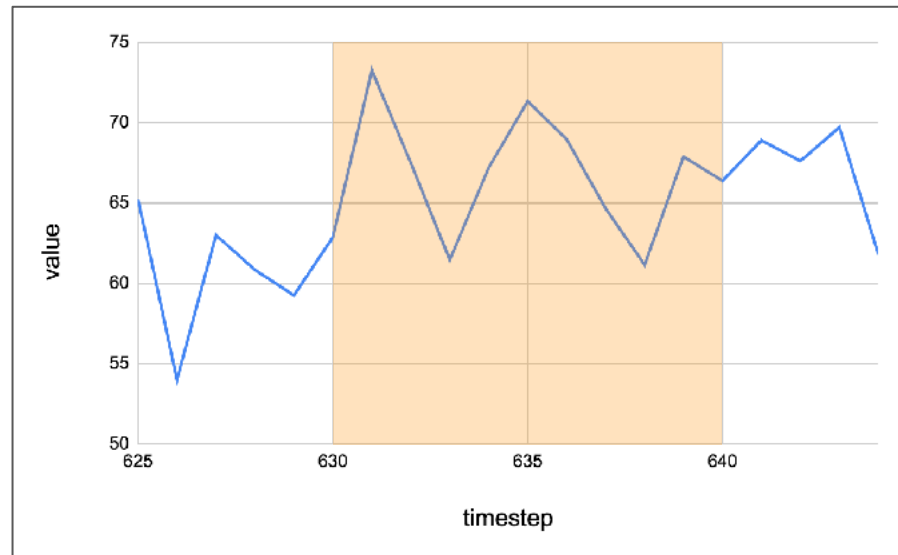


Forecasts = trailing moving average of differenced series + centered moving average of past series ( $t - 365$ )

Trailing Moving Average of Differenced Series  
(zoomed at  $t_{1000}$ , window size = 30)



Centered Moving Average of Past Series ( $t - 365$ )  
(zoomed at  $t_{635}$ , window size = 11)



$$TMA_{t_{1000}} = (v_{t_{970}} + v_{t_{971}} + v_{t_{972}} + \dots + v_{t_{999}}) /$$

$$CMA_{t_{635}} = (v_{t_{630}} + v_{t_{631}} + v_{t_{632}} + \dots + v_{t_{640}}) / 11$$

30

forecast at  $t_{1000} = TMA_{t_{1000}} + CMA_{t_{635}}$