# Finding Seasonal Trends in Time-Series Data with Python

A guide to understanding the different kinds of seasonality and how to decompose the time series into trends and seasons

Spencer Hayes 3 days ago · 5 min read

Photo by Craig Adderley from Pexels

## Why Explore Seasonality?

Seasonality in time-series data refers to a pattern that occurs at a regular interval. This is different from regular cyclic trends, such as the rise and fall of stock prices, that re-occur regularly but don't have a fixed period. There's a lot of insight to be gained from understanding seasonality patterns in your data and you can even use it as a baseline to compare your time-series machine learning models.

## Getting Started

*Quick note: For this article, I'll be using data published by the Quebec Professional Association of Real Estate Brokers. The association publishes monthly real estate stats. For convenience, I've put the monthly median condo prices for the Province of Quebec and the Montreal Metropolitan Area into a CSV file, available here: https://drive.google.com/file/d/1SMrkZPAa0aAl-ZhnHLLFbmdgYmtXgpAb/view?usp=sharing*

The quickest way to get an idea of whether or not your data has a seasonal trend is by plotting it. Let's see what we get when we plot the median house price in Montreal by

```
data_orig = pd.read_csv('quebec_real_estate.csv')

data_orig['Date'] = pd.to_datetime(data_orig['Date']) # convert date column to DateTime
ax = data_orig.plot(x='Date', y='Montreal_Median_Price', figsize=(12,6))
```

**read_and_plot.py** hosted with ❤ by **GitHub** **view raw**

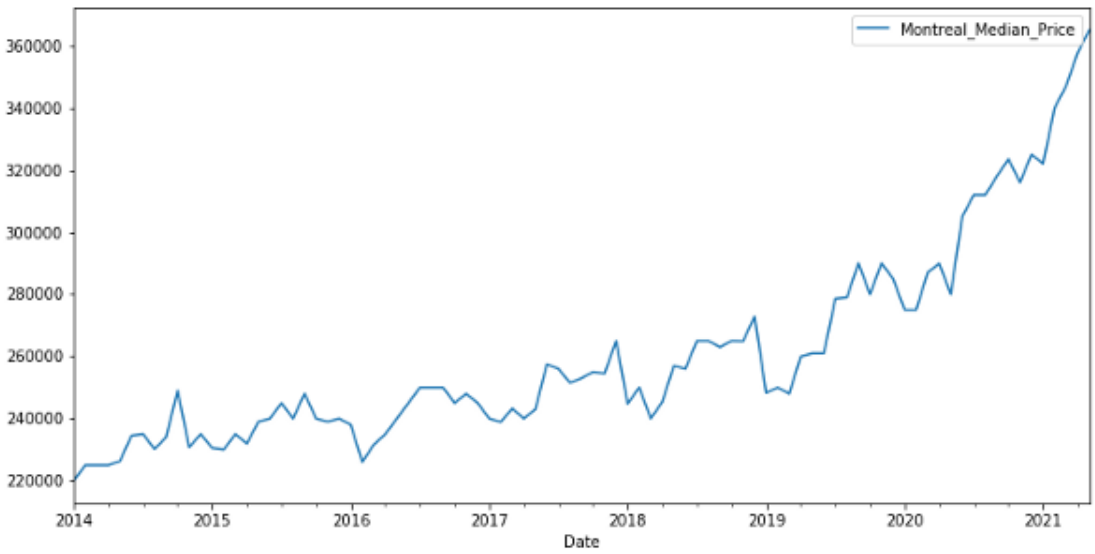



Image by author

A keen eye might already see from this plot that the prices seem to dip around the new year and peak a few months before, around late summer. Let's dive a little further into this by plotting a vertical line for January of every year.

```
ax = data_orig.plot(x='Date', y='Montreal_Median_Price', figsize=(12,6))
xcoords = ['2015-01-01', '2016-01-01','2017-01-01', '2018-01-01', '2019-01-01', '2020-01
           '2021-01-01']
for xc in xcoords:
    plt.axvline(x=xc, color='black', linestyle='--')
```

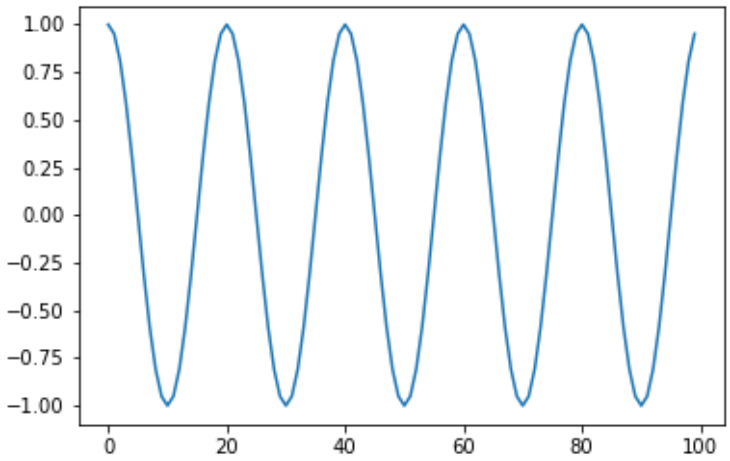
**plotting2.py** hosted with ❤ by **GitHub** **view raw**


Image by author

It seems like there's definitely a trend here. In this case, it appears the seasonality has a period of one year. Next, we'll look into a tool we can use to further examine the seasonality and break down our time series into its trend, seasonal, and residual components. Before we can do that though, you'll have to understand the difference between an additive and a multiplicative seasonality.

## Additive vs Multiplicative Seasonality

There are two types of seasonality that you may come across when analyzing time-series data. To understand the difference between them let's look at a standard time series with perfect seasonality, a cosine wave:



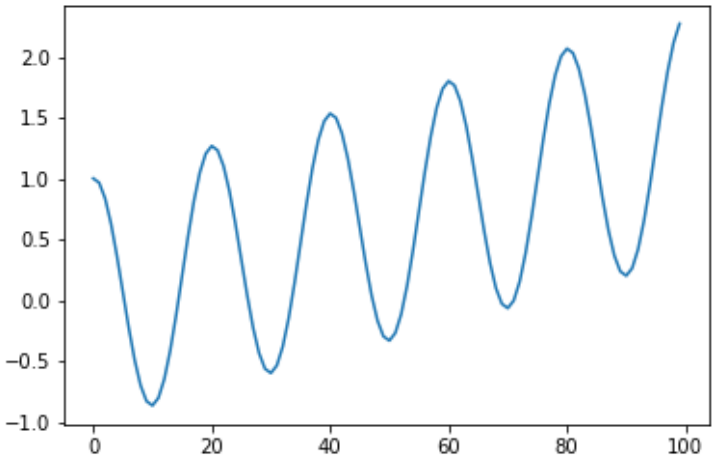

Sine Wave Plot — Image by author

We can clearly see that the period of the wave is 20 and the amplitude (distance from the centre line to the top of a crest or to the bottom of a trough) is 1 and remains constant.

## Additive Seasonality

It's pretty rare for actual time series to have constant crest and trough values and instead, we typically see some kind of general trend like an increase or a decrease over time. In our sales price plot, for example, the median price tends to go up over time.

If the amplitude of our seasonality tends to remain the same, then we have what's called an additive seasonality. Below is an example of an additive seasonality.



A great way to think about it is by imagining we took our standard cosine wave and simply added a trend to it:

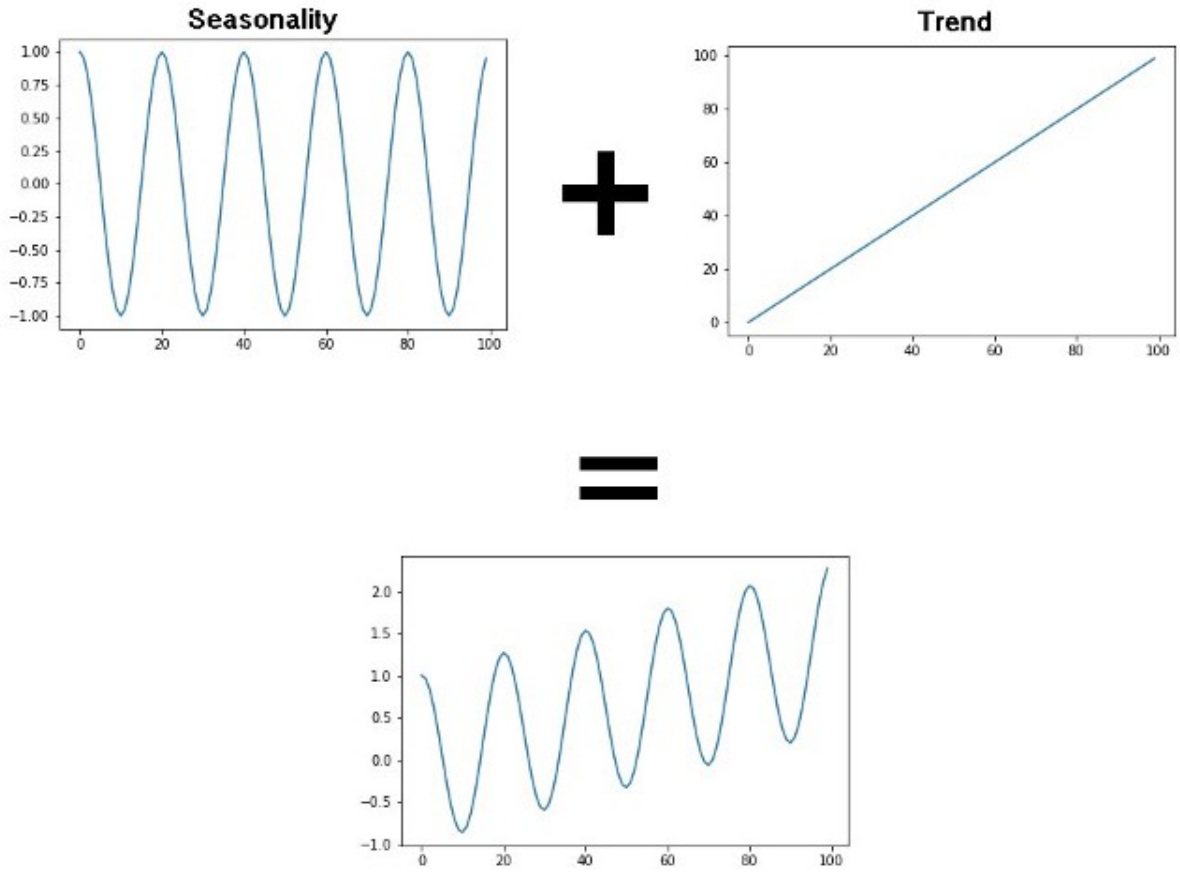


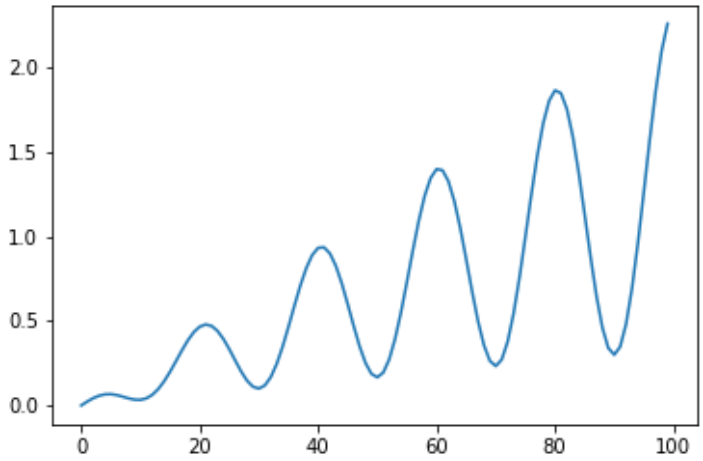

Image by author

We can even think of our basic cosine model from earlier as an additive model with a constant trend! We can model additive time series using the following simple equation:

$$Y[t] = T[t] + S[t] + e[t]$$

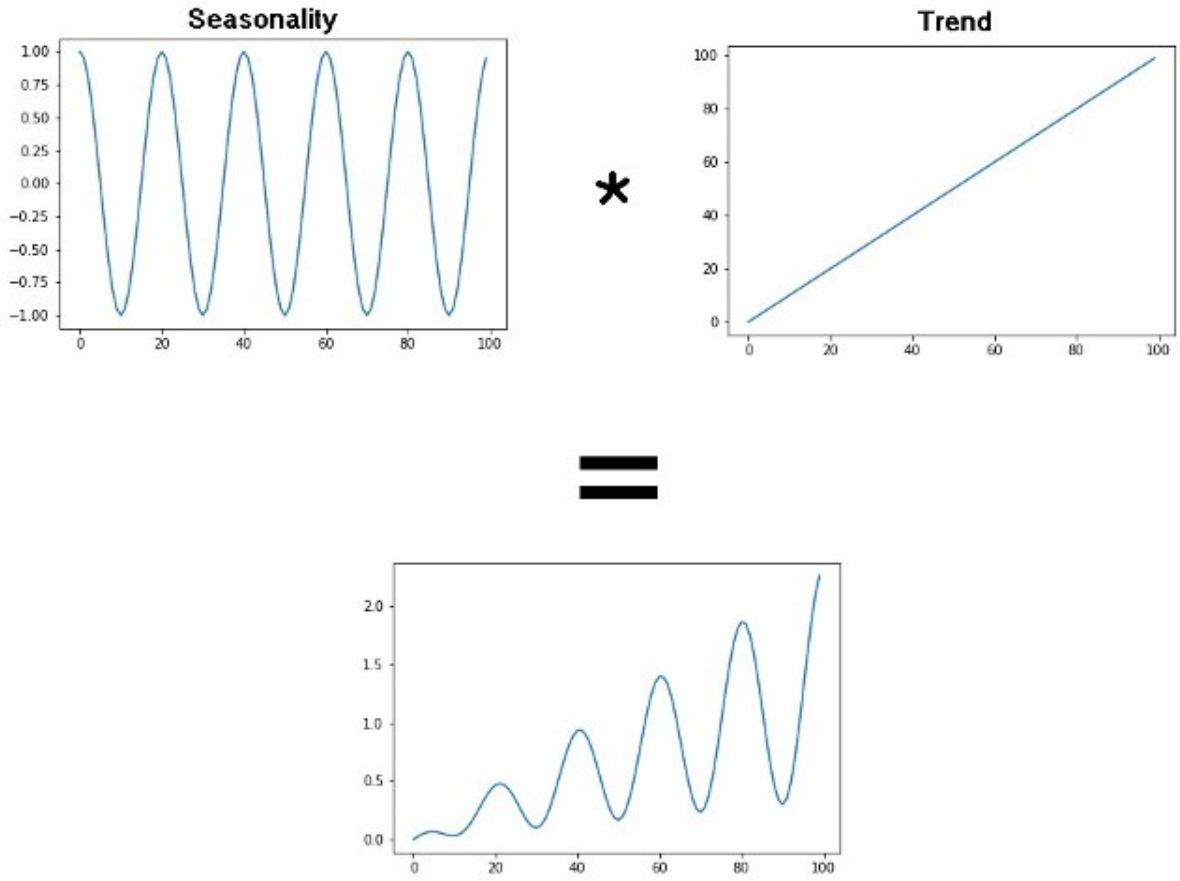
Y[t]: Our time-series function
T[t]: Trend (general tendency to move up or down)
S[t]: Seasonality (cyclic pattern occurring at regular intervals)
e[t]: Residual (random noise in the data that isn't accounted for in the trend or seasonality

The other type of seasonality that you may encounter in your time-series data is multiplicative. In this type, the amplitude of our seasonality becomes larger or smaller based on the trend. An example of multiplicative seasonality is given below.




Multiplicative seasonality — Image by author

We can apply a similar train of thought as we used with our additive model and imagine that we took our cosine wave but instead of adding the trend, we multiplied it (hence the name multiplicative seasonality):




Multiplicative seasonality — Image by author

We can model this with a similar equation as our additive model by just swapping the additions for multiplications.

$Y[t] = T[t] * S[t] * e[t]$

## Decomposing the dataset

Now that we have a clear picture of the different models, let's look at how we can break down our real estate time series into its trend, seasonality, and residual components. We'll be using the seasonal_decompose model from the statsmodels library.

The seasonal_decompose model requires you to select a model type for the seasonality (additive or multiplicative). We'll select a multiplicative model since it would appear the amplitude of the cycles is increasing with time. This would make sense since a large factor for housing prices is lending rates which are done as a percentage of the price.

```
2
3  data_orig.set_index('Date', inplace=True)
4
5  analysis = data_orig[['Montreal_Median_Price']].copy()
6
7
8  decompose_result_mult = seasonal_decompose(analysis, model="multiplicative")
9
10 trend = decompose_result_mult.trend
11 seasonal = decompose_result_mult.seasonal
12 residual = decompose_result_mult.resid
13
14 decompose_result_mult.plot();
```

**seasonal_decompose.py** hosted with ❤ by **GitHub** viev

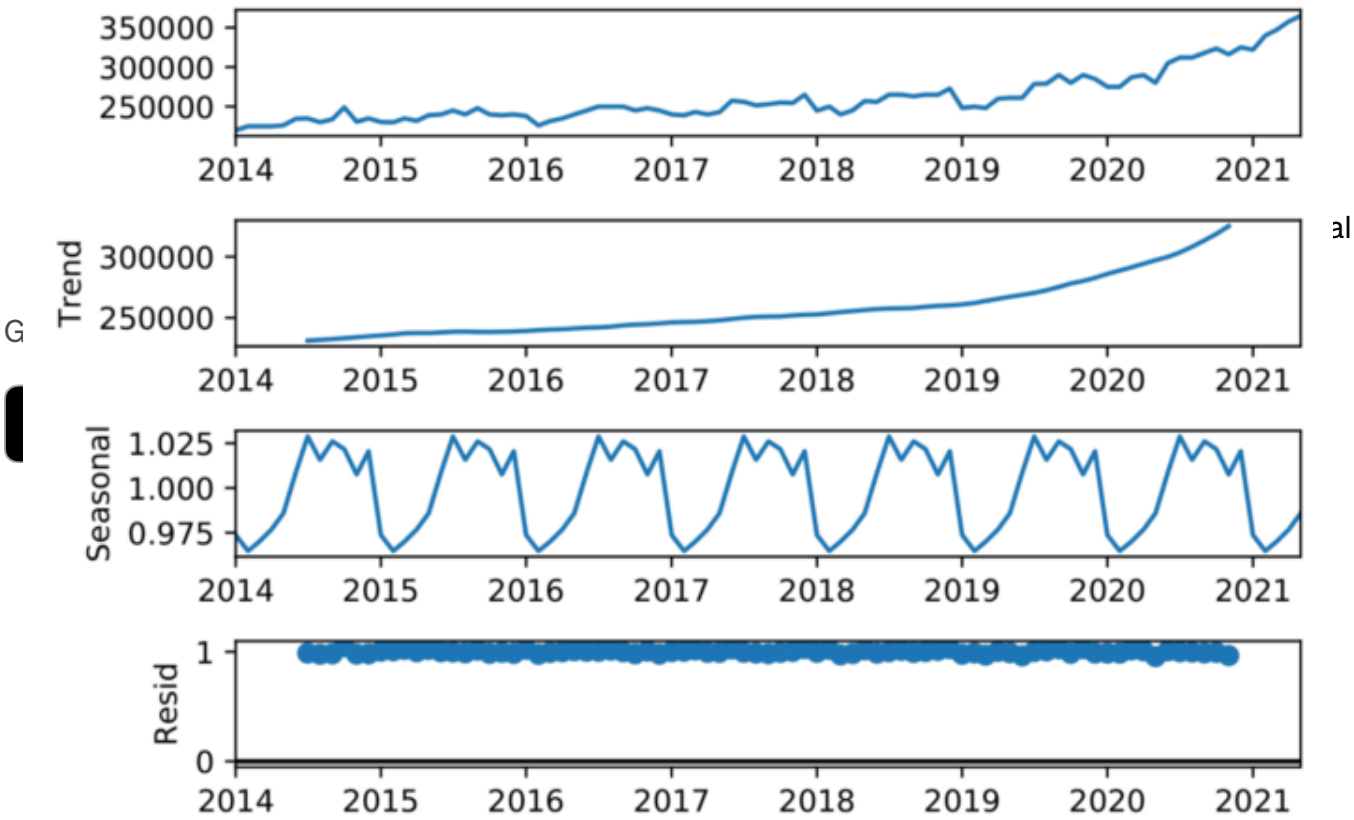



Image by author

Ta-da! The trend, seasonal, and residual components are returned as Pandas series so you can plot them by calling their plot() methods or perform further analysis on them.

strong correlation between the residual and the number of new babies born in the city.

## Conclusion

From our decomposition, we can see the model picked up on a 5% difference between the seasons. If you're looking to sell your house, you should probably list it in mid to late spring instead of late fall if you want to get top dollar!