A Deep Intro to Time Series Forecasting Prerequisite:

Rohit Umredkar · Follow

8 min read · Jan 7, 2024

Time Series Components

A time series can be decomposed into several components, and understanding these components is crucial for effective forecasting:

Level

The average value for a specific time period

Trend

The long-term movement or direction in the data.

It can be upward, downward, or remain relatively constant.

Trends can be linear or nonlinear.

Seasonality

Repeating patterns or cycles within a fixed time interval.

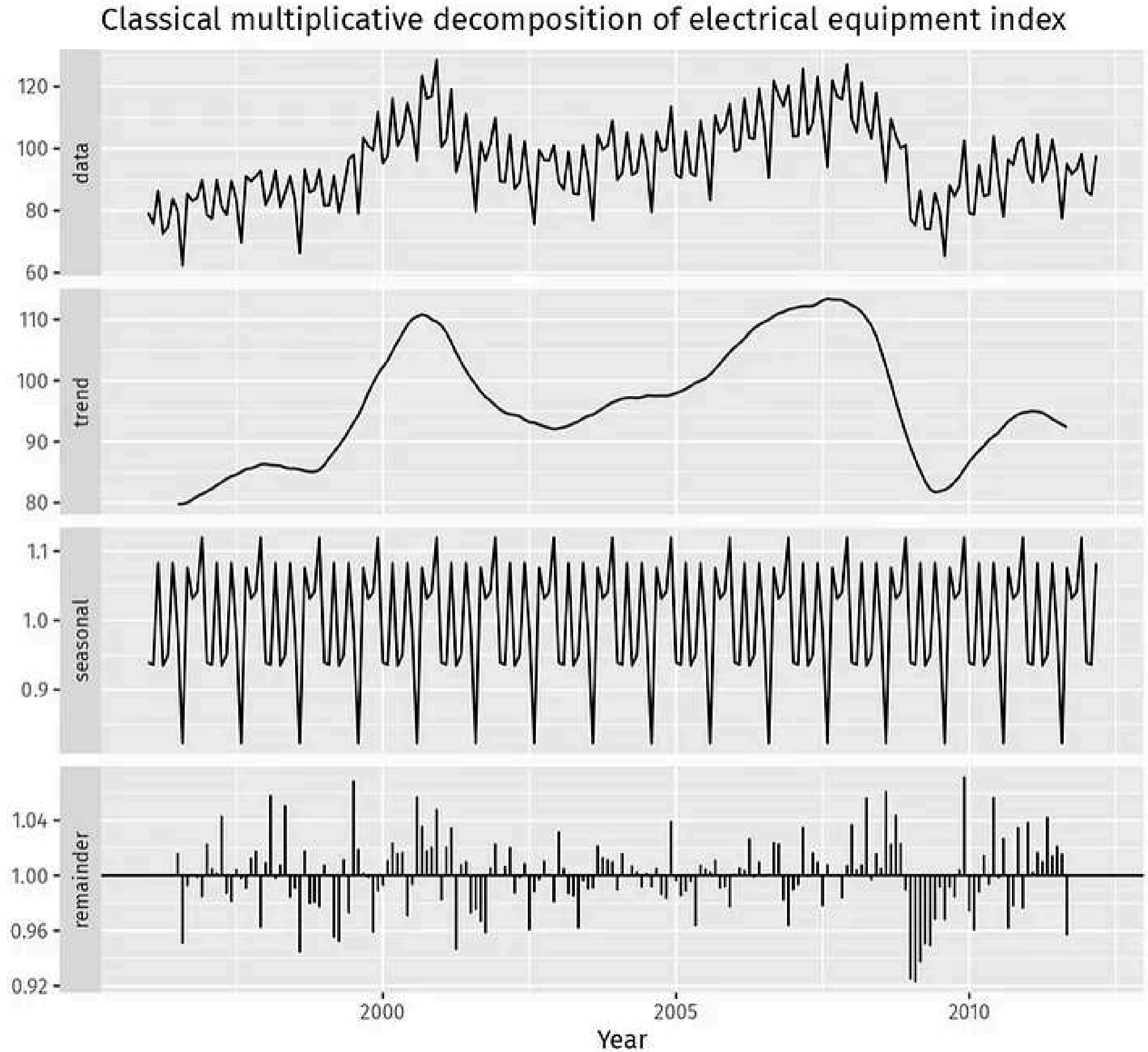
Seasonal effects are often driven by external factors like weather or holidays.

Seasonality can be daily, weekly, monthly, or yearly.

Noise (Residuals)

Random fluctuations or irregularities in the data that cannot be attributed to the trend or seasonality.

Represented by the difference between the observed and predicted values.



Stationarity

In time series analysis, stationarity refers to the statistical properties of a time series remaining constant over time. There are two main types of stationarity: additive stationarity and multiplicative stationarity. Additionally, we can consider the concept of seasonality and trend, which can contribute to non-stationarity.

Additive Stationarity: A time series is said to exhibit additive stationarity if the mean and variance remain constant over time.

Equation: $Yt = \mu + \epsilon t$
Here, Yt is the observed value at time t, μ is the constant mean, and ϵt is the white noise or random error term.

Multiplicative Stationarity: A time series is said to exhibit multiplicative stationarity if the mean and variance are proportional and remain constant over time.

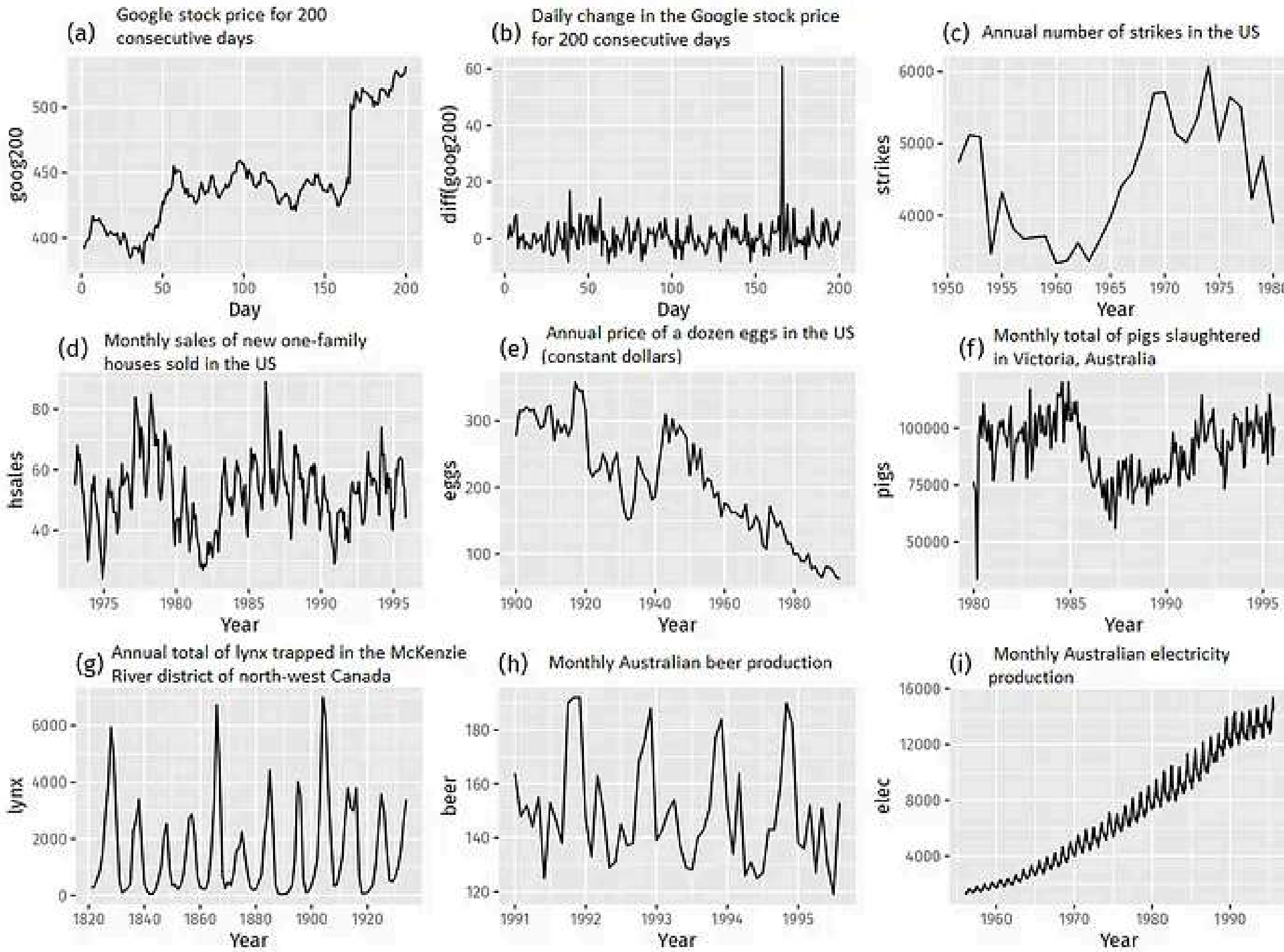
Equation: $Yt = \mu \times \epsilon t$
Similar to additive stationarity, but the observed value is the product of a constant mean μ and a random error term ϵt.

Seasonal Stationarity: Seasonal stationarity refers to a time series that exhibits a repeating pattern or seasonality over fixed time intervals.

Equation: $Yt = \mu + \gamma t + \epsilon t$
In addition to the mean μ and random error term ϵt, γt represents the seasonal component.

Trend Stationarity: Trend stationarity refers to a time series that maintains a consistent trend over time.

Equation: $Yt = \mu + \beta t + \epsilon t$
Here, in addition to the mean μ and random error term ϵt, βt represents the trend component.



Consider the nine series plotted in abovr Figure. Which of these do you think are stationary?

Obvious seasonality rules out series (d), (h) and (i). Trends and changing levels rules out series (a), (c), (e), (f) and (i). Increasing variance also rules out (i). That leaves only (b) and (g) as stationary series.

At first glance, the strong cycles in series (g) might appear to make it nonstationary. But these cycles are aperiodic — they are caused when the lynx population becomes too large for the available feed, so that they stop breeding and the population falls to low numbers, then the regeneration of their food sources allows the population to grow again, and so on. In the long-term, the timing of these cycles is not predictable. Hence the series is stationary.

Achieving stationarity is often a goal in time series analysis because many forecasting methods assume a stationary time series. If a time series is not stationary, techniques such as differencing or transformations (e.g., logarithmic transformation) can be applied to stabilize the mean and variance.

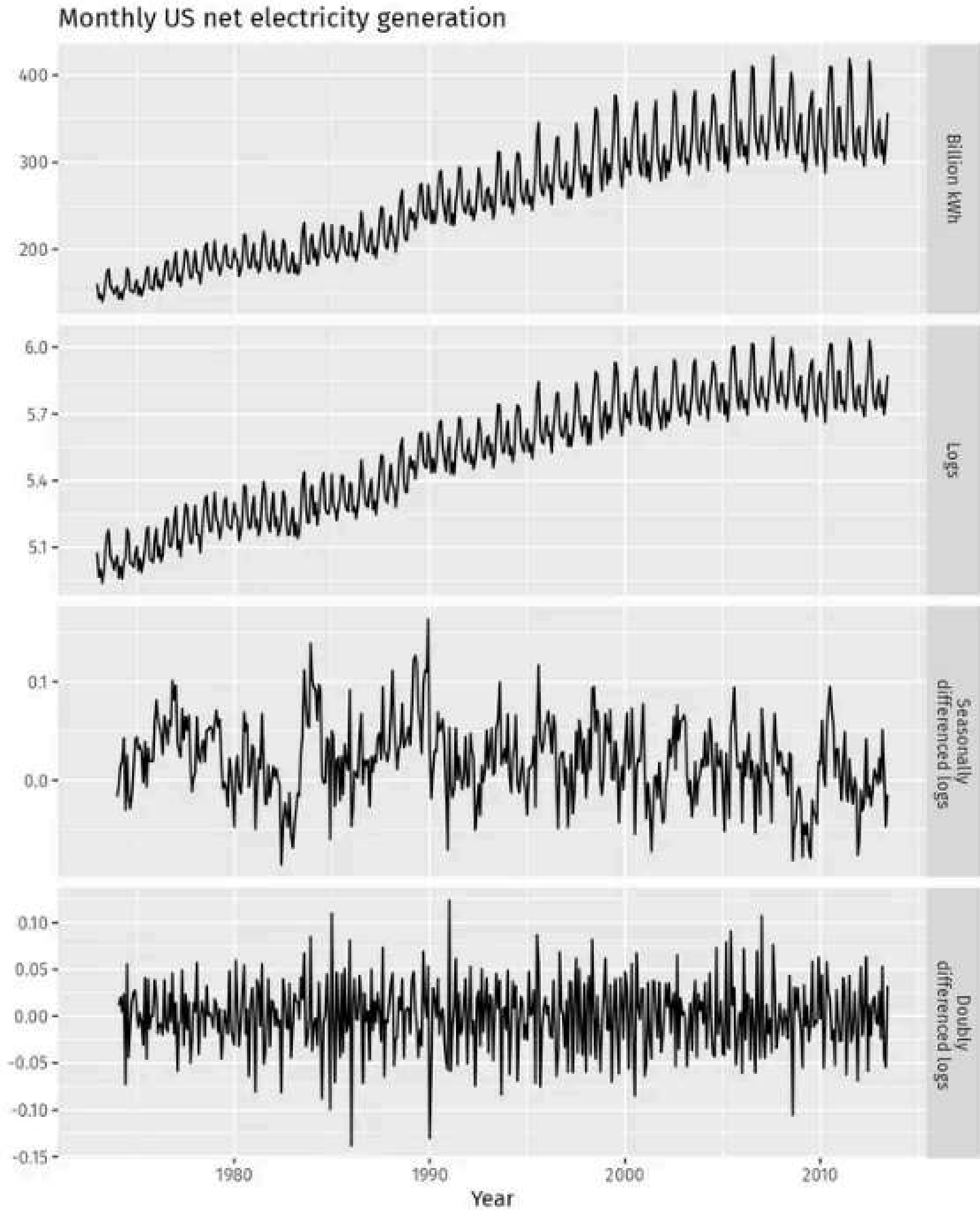
In practice, whether to treat a time series as additive or multiplicative, and the consideration of seasonality and trend, depends on the characteristics of the specific data and the goals of the analysis.

Below are methods with stationarity can be achieved :

Differencing:
Compute the difference between consecutive observations.
This helps stabilize the mean and reduce trend and seasonality.

Transformations:

Apply mathematical transformations (e.g., logarithm) to stabilize variance.


Monthly US net electricity generation
400
300
200
Billion kWh
6.0
5.7
5.4
5.1
Logs
0.1
0.0
Seasonally differenced logs
0.10
0.05
0.00
-0.05
-0.10
-0.15
Doubly differenced logs
1980
1990
2000
2010
Year

When both seasonal and first differences are applied, it makes no difference which is done first — the result will be the same. However, if the data have a strong seasonal pattern, we recommend that seasonal differencing be done first, because the resulting series will sometimes be stationary and there will be no need for a further first difference. If first differencing is done first, there will still be seasonality present.
It is important that if differencing is used, the differences are interpretable. First differences are the change between one observation and the next. Seasonal differences are the change between one year to the next. Other lags are unlikely to make much interpretable sense and should be avoided.

Some formal tests for differencing are discussed below:

Unit root tests:
One way to determine more objectively whether differencing is required is to use a unit root test. These are statistical hypothesis tests of stationarity that are designed for determining whether differencing is required.

In this test, the *null hypothesis is that the data are stationary,* and we look for *evidence that the null hypothesis is false*.

In our analysis, we use the *Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test* (Kwiatkowski, Phillips, Schmidt, & Shin, 1992)

```
goog %>% ur.kpss() %>% summary()
#>
#> #######################
#> # KPSS Unit Root Test #
#> #######################
#>
#> Test is of type: mu with 7 lags.
#>
#> Value of test-statistic is: 10.72
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected. That is, the data are not stationary. We can difference the data, and apply the test again.

```
goog %>% diff() %>% ur.kpss() %>% summary()
#>
#> #######################
#> # KPSS Unit Root Test #
#> #######################
#>
#> Test is of type: mu with 7 lags.
#>
#> Value of test-statistic is: 0.0324
#>
#> Critical value for a significance level of:
#>                 10pct  5pct 2.5pct  1pct
#> critical values 0.347 0.463  0.574 0.739
```

This time, the test statistic is tiny, and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

This process of using a sequence of KPSS tests to determine the appropriate number of first differences is carried out by the function `ndiffs()`

```
ndiffs(goog)
#> [1] 1
```

As we saw from the KPSS tests above, one difference is required to make the `google` data stationary.

Autocorrelation and Partial Autocorrelation:

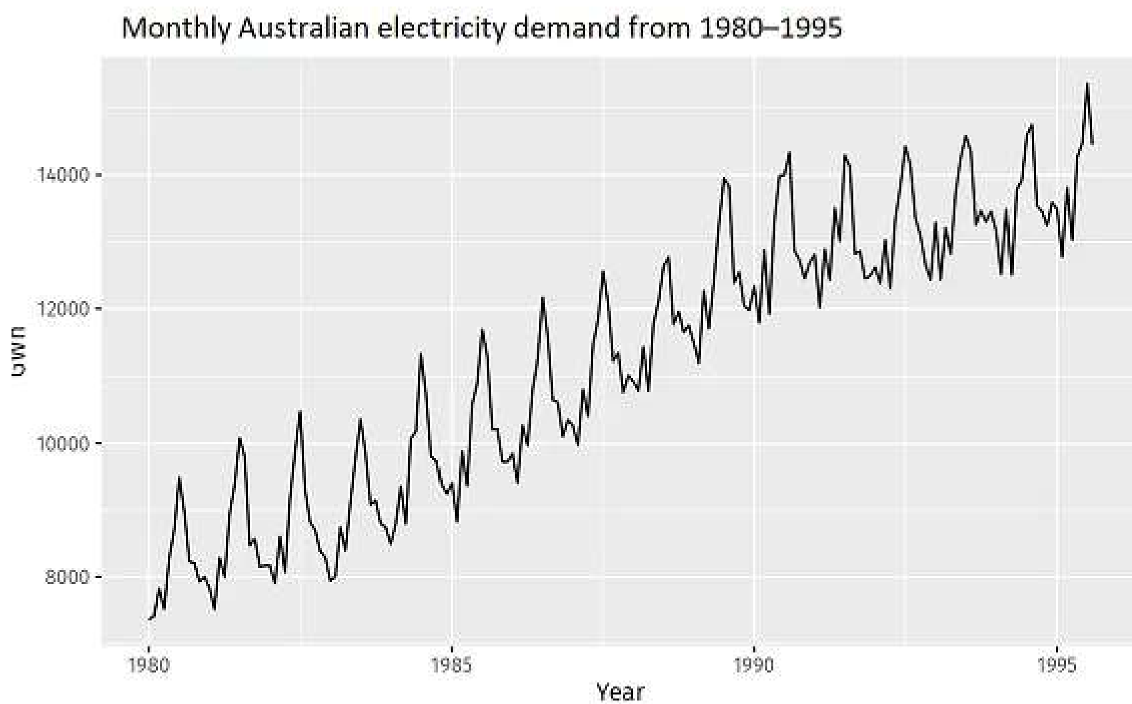
Understanding autocorrelation (ACF) and partial autocorrelation (PACF) is crucial for selecting appropriate models:

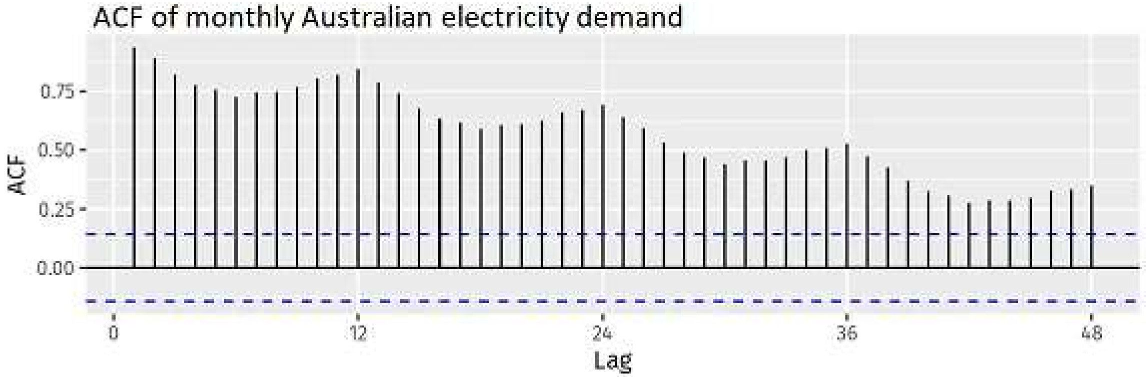
Autocorrelation Function (ACF):
Measures the correlation between a time series and its lagged values at different time intervals.
Helps identify the order of the Moving Average (MA) component in ARIMA models.

When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So the ACF of trended time series tend to have positive values that slowly decrease as the lags increase.

When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags.

When data are both trended and seasonal, you see a combination of these effects. The monthly Australian electricity demand series plotted in Figure below shows both trend and seasonality.





The slow decrease in the ACF as the lags increase is due to the trend, while the "scalloped" shape is due to the seasonality.

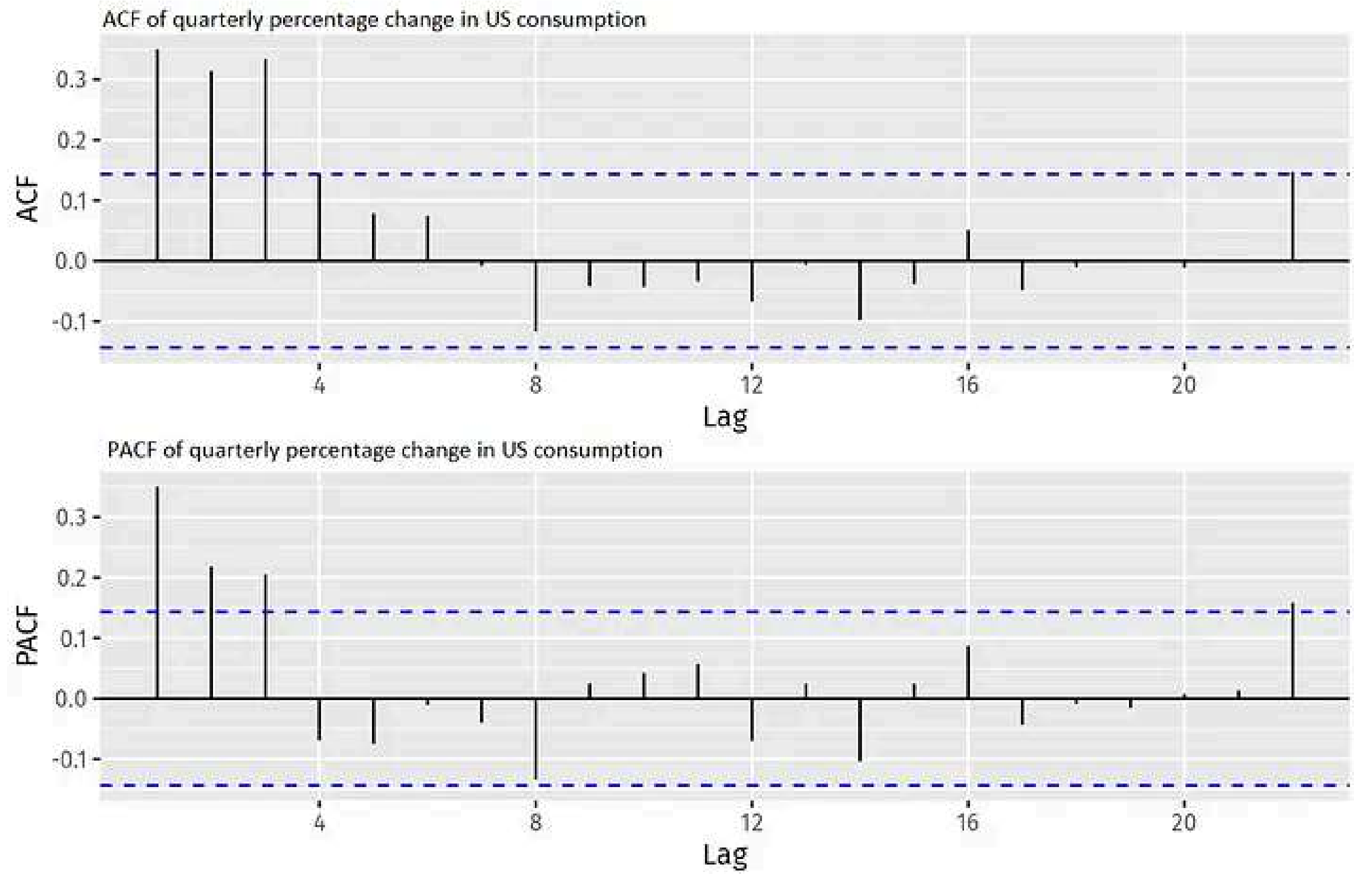
Partial Autocorrelation Function (PACF):

Measures the correlation between a time series and its lagged values, controlling for the effect of intervening lags.

Helps identify the order of the AutoRegressive (AR) component in ARIMA models.

In an ACF plot shows the autocorrelations which measure the relationship between Y(t) and Y(t−k) for different values of k i.e lag. Now if yt and yt−1 are correlated, then yt−1 and yt−2 must also be correlated. However, then yt and yt−2 might be correlated, simply because they are both connected to yt−1, rather than because of any new information contained in yt−2 that could be used in forecasting yt.

To overcome this problem, we can use **partial autocorrelations.** These measure the relationship between yt and yt−k after removing the effects of lags 1,2,3,…,k−1.



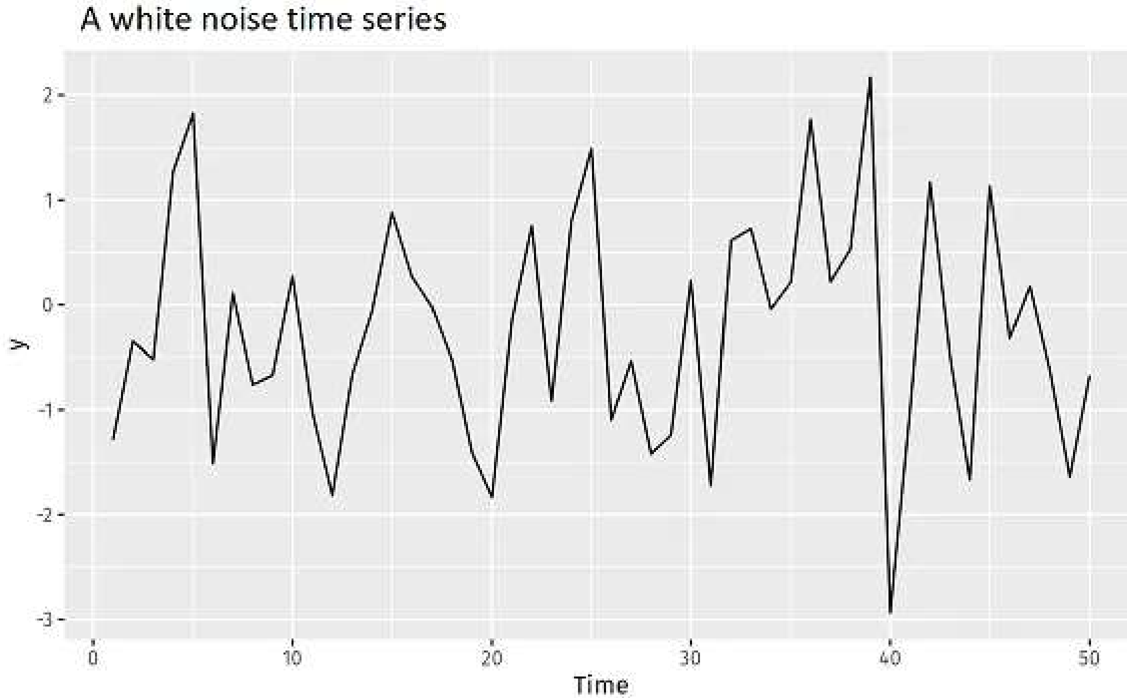
In above Figure, we see that there are three spikes in the ACF, followed by an almost significant spike at lag 4. In the PACF, there are three significant spikes, and then no significant spikes thereafter (apart from one just outside the bounds at lag 22). The pattern in the first three spikes is what we would expect from an ARIMA(3,0,0), as the PACF tends to decrease. So in this case, the ACF and PACF lead us to think an ARIMA(3,0,0) model might be appropriate.

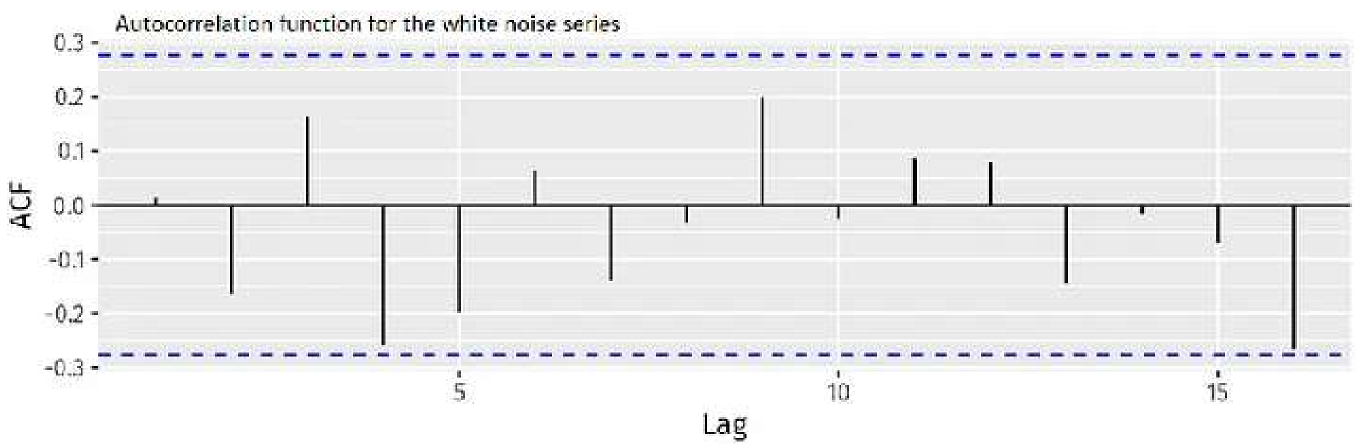
The data may follow an *ARIMA(p,d,0)* model, if the ACF and PACF plots of the differenced data show the following patterns:
1) The ACF is exponentially decaying or sinusoidal;
2) There is a significant spike at lag *p*, in the PACF, but none beyond lag *p*.

The data may follow an *ARIMA(0,d,q)* model if the ACF and PACF plots of the differenced data show the following patterns:
1) The PACF is exponentially decaying or sinusoidal;
2) There is a significant spike at lag *q* in the ACF, but none beyond lag *q*.

White noise

Time series that show no autocorrelation are called **white noise.** Figure below gives an example of a white noise series.





As we wrap up this deep introduction to time series forecasting, it's evident that the field demands a blend of statistical understanding, mathematical prowess, and practical implementation. The prerequisites discussed lay the groundwork for anyone looking to master the art of predicting future trends based on historical data.

In the vast landscape of time series forecasting, continuous learning, experimentation, and adaptation to diverse datasets are key. Armed with these prerequisites, you are now equipped to embark on your journey towards becoming a proficient time series forecaster.

Whether your goal is to make informed business decisions, predict stock prices, or optimize resource planning, the insights gained from this deep intro will serve as a compass, guiding you through the intricate terrain of time series forecasting.

Stay curious, keep experimenting, and forecast on! 📈 ✨

Forecasting | Timeseries Forecasting | Data Science | Data Anaytics | Time Series Analysis

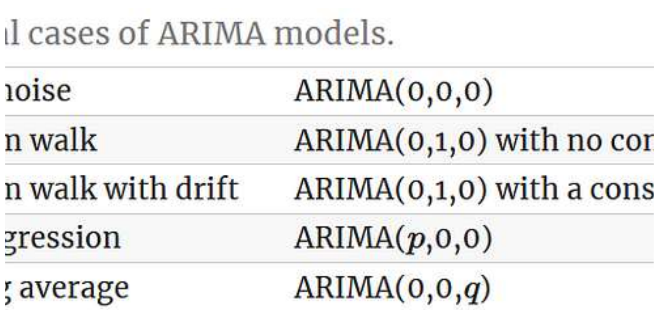
Written by Rohit Umredkar

Follow

2 Followers

Ads Analytics - Airtel Digital

More from Rohit Umredkar

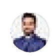

ıl cases of ARIMA models.

ıoise	ARIMA(0,0,0)
n walk	ARIMA(0,1,0) with no cor
n walk with drift	ARIMA(0,1,0) with a cons
;ression	ARIMA(p,0,0)
; average	ARIMA(0,0,q)

 Rohit Umredkar

The Guide to AR, MA, ARIMA, SARIMA Forecasting Method

ARIMA models provide another approach to time series forecasting. Exponential...

 Rohit Umredkar

Unlocking Efficiency with Postman Flow: To Test & Automate API

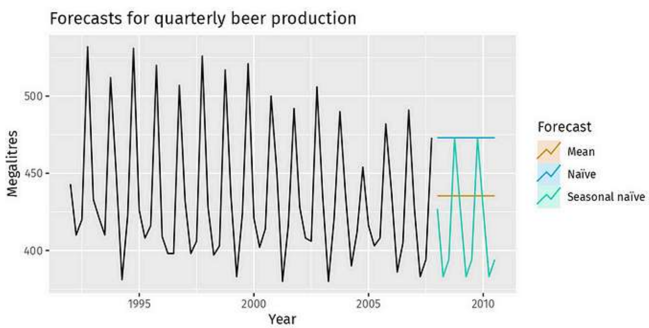
Excited to share my recent experiences with Postman-Flow used in one of the use-case a...

7 min read · Jan 10, 2024

4

1 min read · Jan 1, 2024

1



Rohit Umredkar

"A Guide to Simple Univariate Time Series Forecasting Techniques"

Some forecasting methods are extremely simple and surprisingly effective. We will us...

6 min read · Jan 7, 2024

Rohit Umredkar

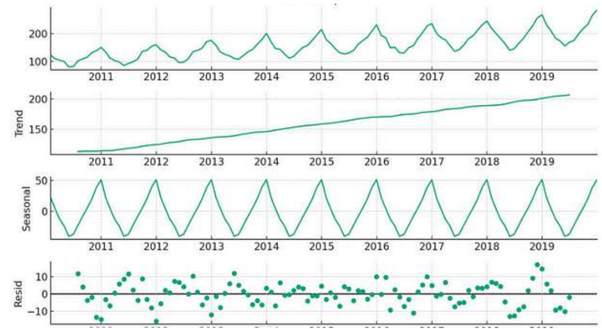
SQL Data Types: A Quick Guide to Database Fields 🗄️ 🔍

In the realm of databases, understanding SQL data types is akin to wielding a master key....

3 min read · Jan 1, 2024

See all from Rohit Umredkar

Recommended from Medium



Vansh Jatana in The Deep Hub

Mastering Time Series Forecasting

Time series forecasting is an important aspect of Data science and is used in differe...

8 min read · Feb 18, 2024

87 2

Tapas

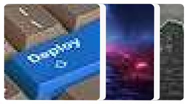
Time Series Accuracy Test: sMAPE vs MAPE

In predictive model performance analysis, several metrics are used to assess the...

5 min read · May 28, 2024

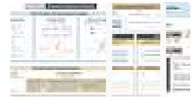
5

Lists



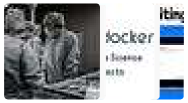
Predictive Modeling w/ Python

20 stories · 1271 saves



Practical Guides to Machine Learning

10 stories · 1528 saves

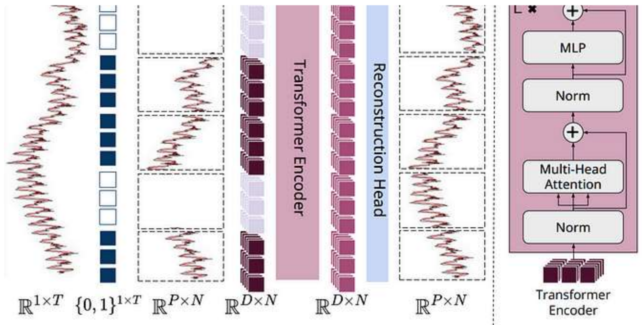


Coding & Development

11 stories · 647 saves

ChatGPT prompts

48 stories · 1655 saves



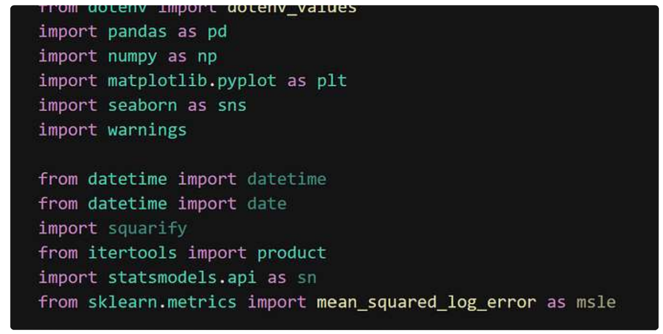
Hadar Sharvit

MOMENT: a familiy of open time-series foudation models

Moment is yet another attempt in foundational time series models. The...

4 min read · May 7, 2024

12



Esawokev

Time-Series Regression Sales Prediction

STORE SALES FORECASTING/PREDICTION-TIME SERIES REGRESSION MODEL

7 min read · May 19, 2024

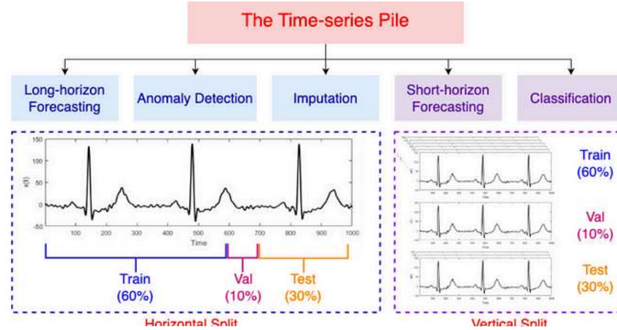
Vitor Cerqueira in Towards Data Science

3 Types of Seasonality and How to Detect Them

Understanding time series seasonality

· 6 min read · Jun 30, 2023

490 4



samuel cha... in Artificial Intelligence in Plain Engl...

Moment: A Family of Open Time-Series Foundation Models

Pre-training large models on time-series data is challenging because of the absence of a...

4 min read · Mar 4, 2024

69 1

See more recommendations

Help Status About Careers Press Blog Privacy Terms Text to speech Teams