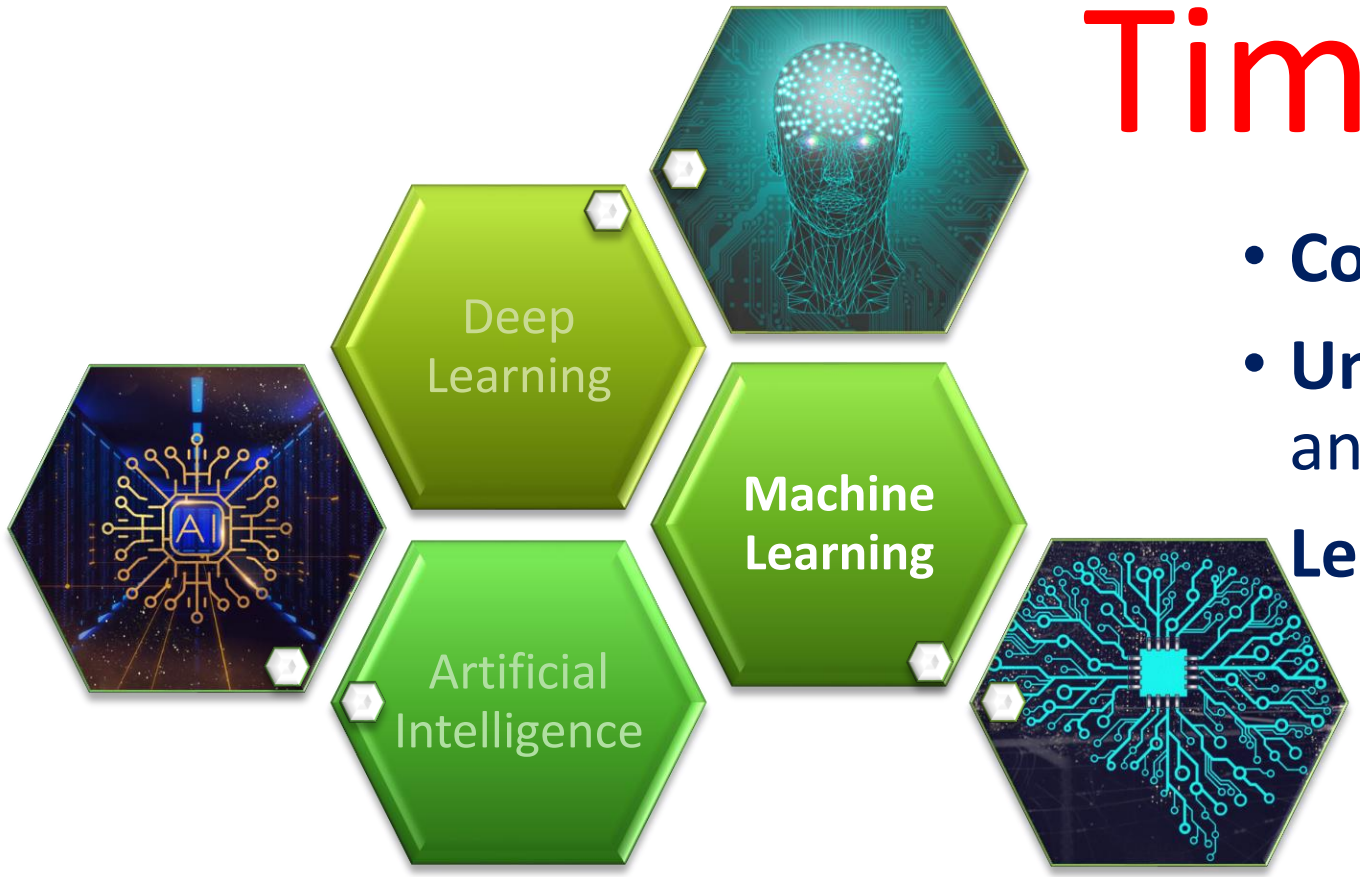


Time Series

- **Course Code: CSE471**
- **Unit 2:** Exploratory time Series analysis

Lecture1

Decomposition models

Time series data can include any of these components. However, not all of them include them in the same way. Let's find out the most common models that take into account these elements. For simplicity, we will ignore the cyclical element.

Additive Model

These models assume the observed time series is the sum of its elements:

$$y(t) = Trend + Seasonality + Residual$$

This model considers that the magnitudes of the seasonal and residual elements are independent of the trend.

Multiplicative Model

These models assume the observed time series is the product of its elements:

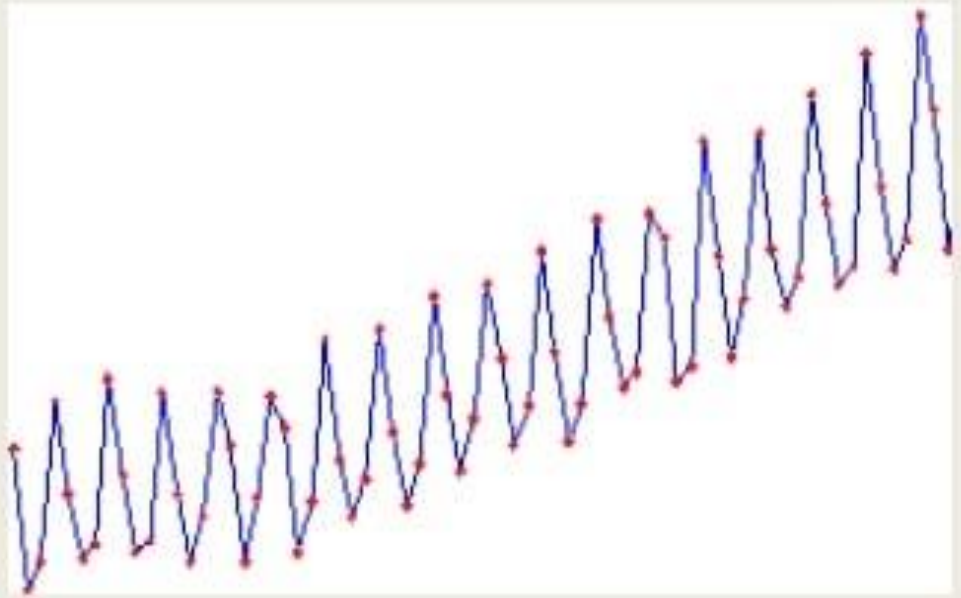
$$y(t) = Trend \times Seasonality \times Residual$$

This model implies that the seasonality and residuals are dependent on the trend.

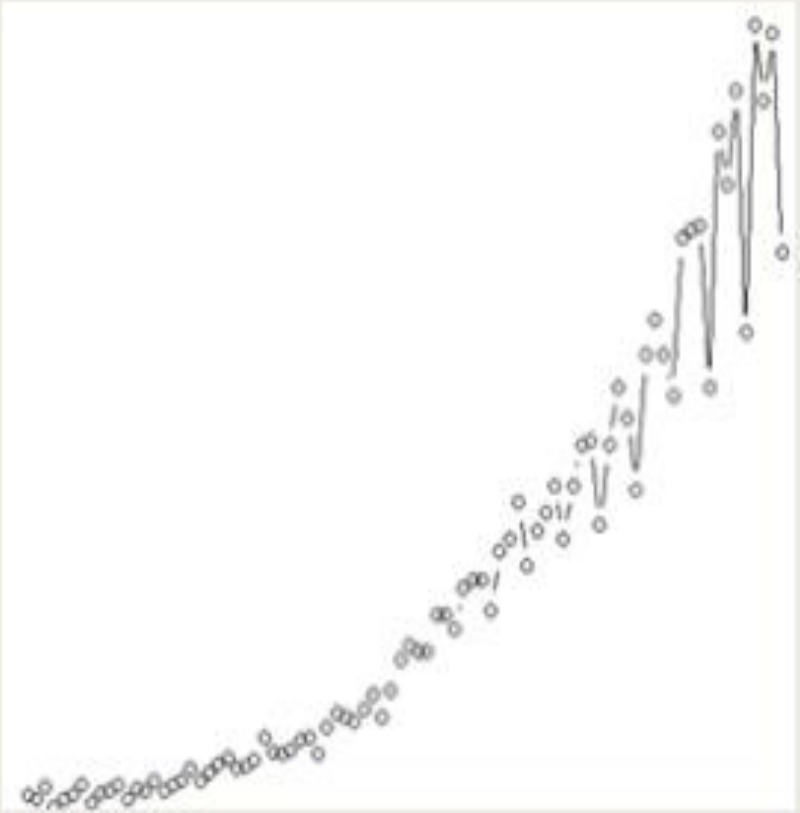
This model can be transformed into an additive model by applying logarithmic transformations:

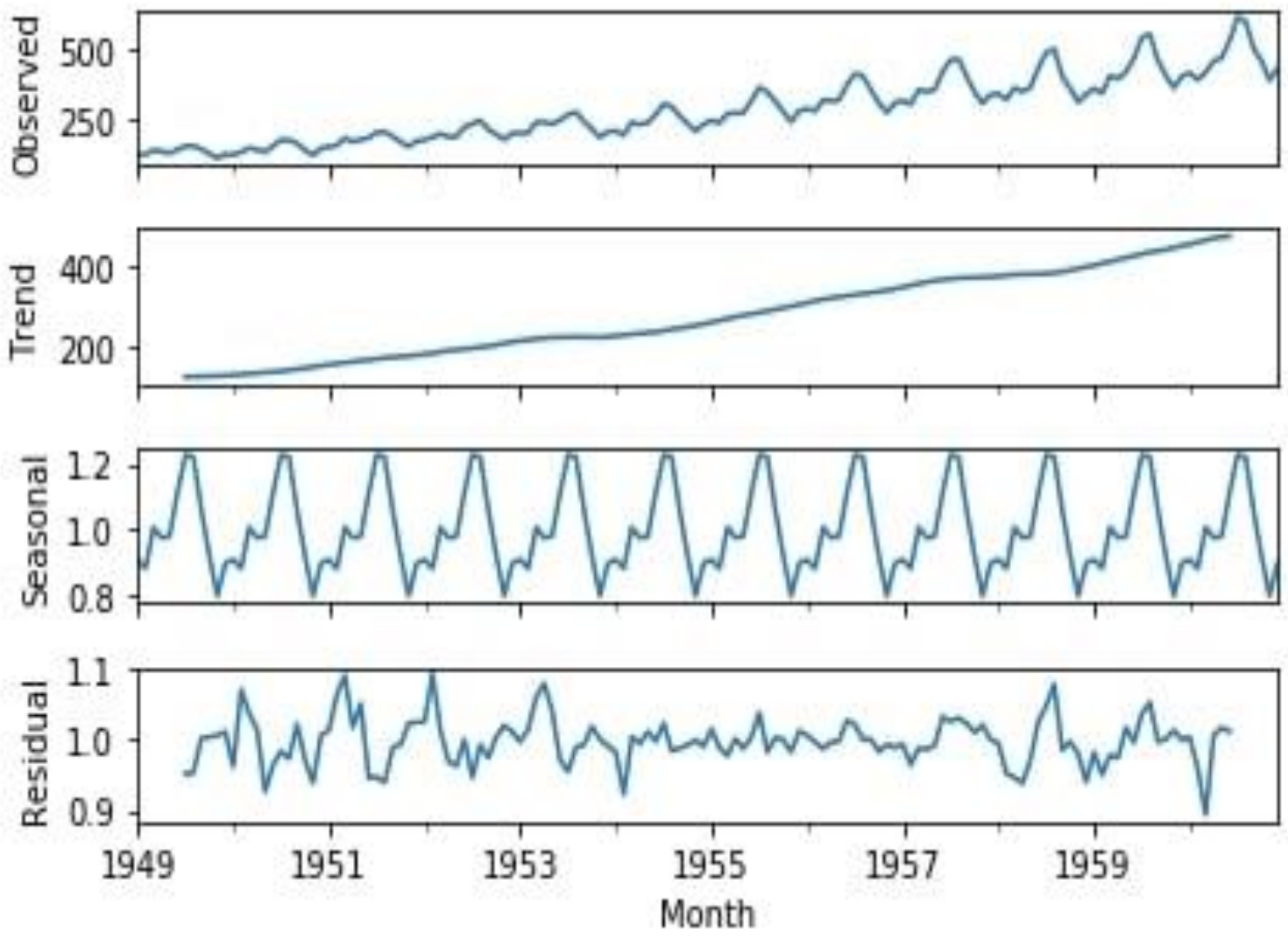
$$log(y(t)) = log(Trend) + log(Seasonality) + log(Residual)$$

Additive



Multiplicative





Python Code

```
from statsmodels.tsa.seasonal import seasonal_decompose
decompose_result = seasonal_decompose(air_passengers, model="multiplicative")

trend = decompose_result.trend
seasonal = decompose_result.seasonal
residual = decompose_result.resid

decompose_result.plot();
```

Remove Trend

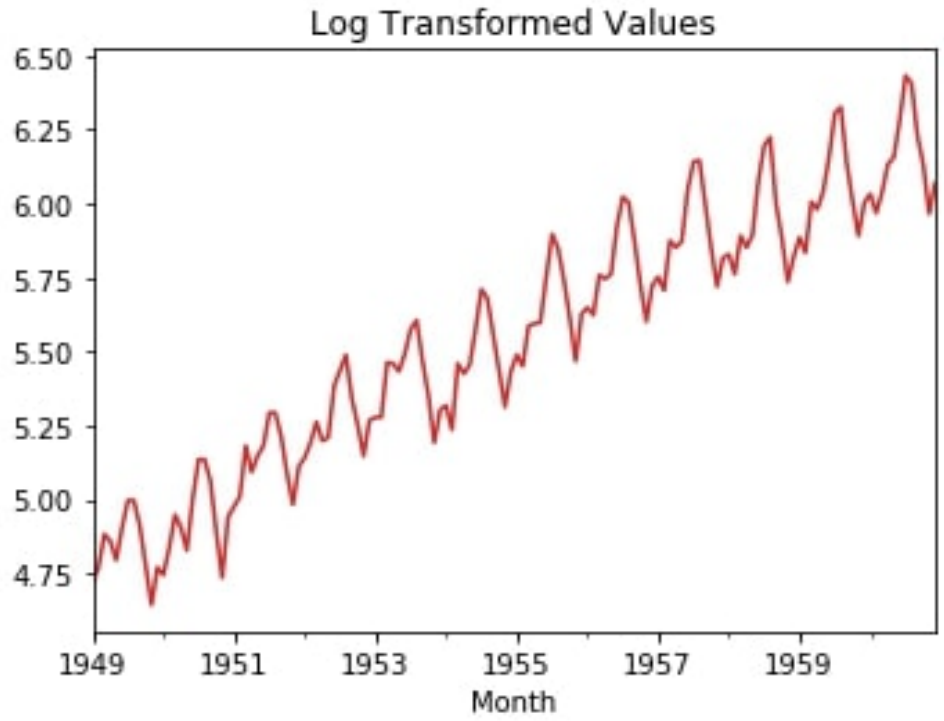
The trend represent an increase or decrease in time-series value over time. If we notice that the value of measurement over time is increasing or decreasing then we can say that it has an upward or downward trend.

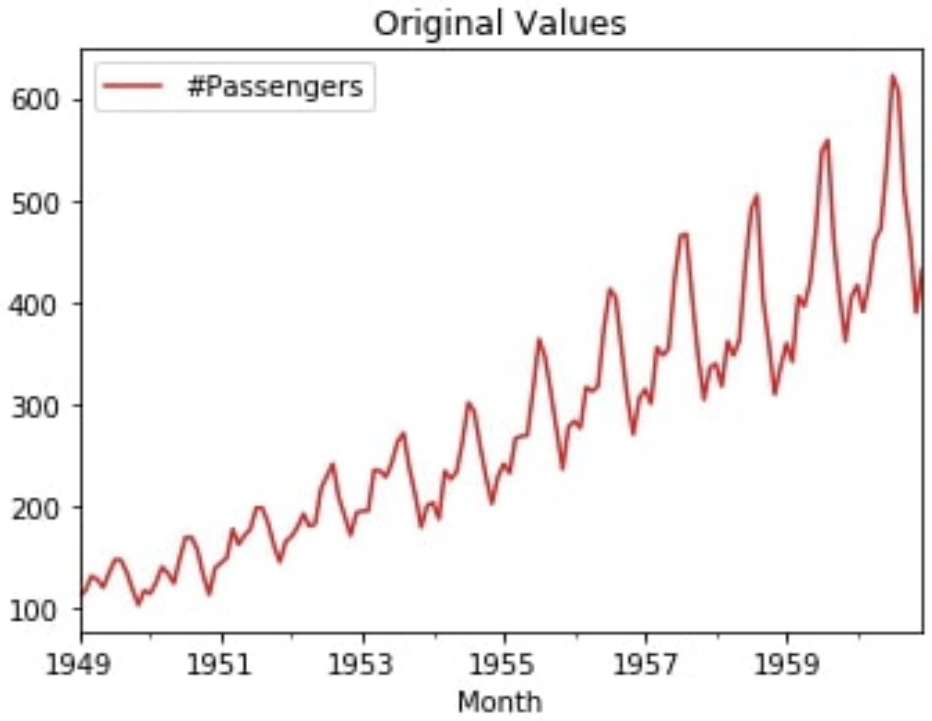
How to remove trend from time-series data?

- Log Transformation.
- Power Transformation.
- local smoothing - Applying moving window functions to time-series data.
- Differencing a time-series.
- Linear Regression.

Log Transformation.

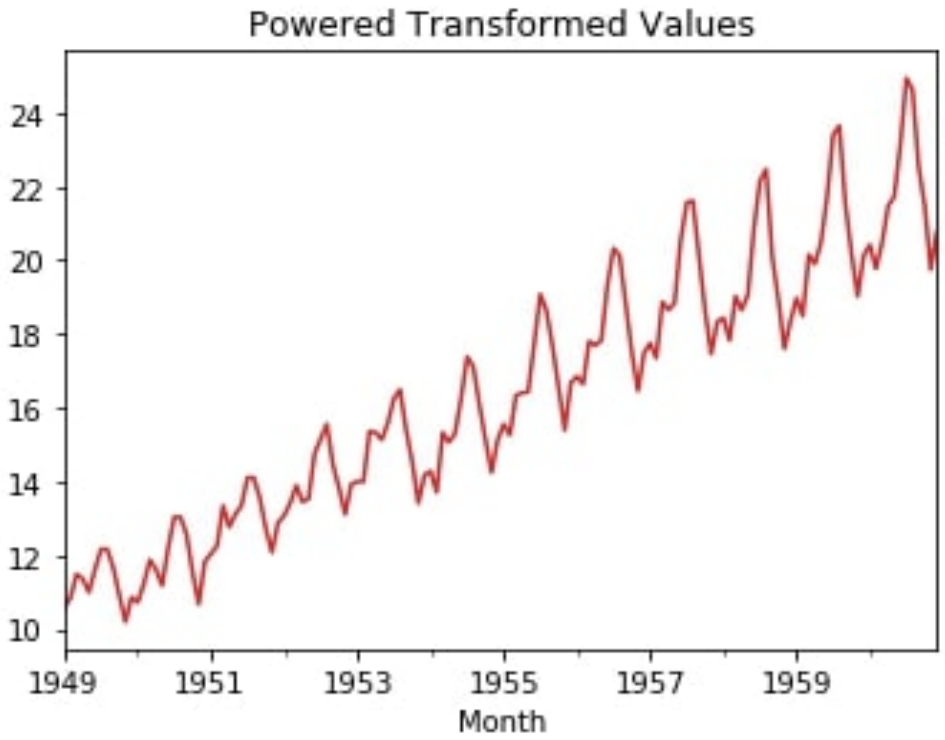
To apply log transformation, we need to take a log of each individual value of time-series data.

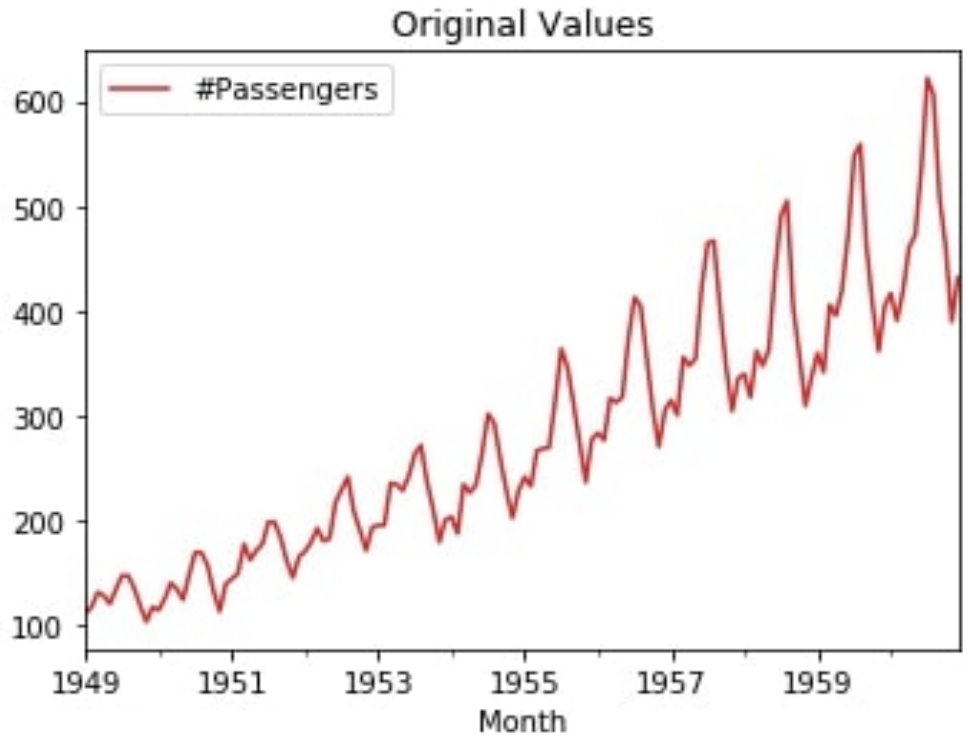




Power Transformation

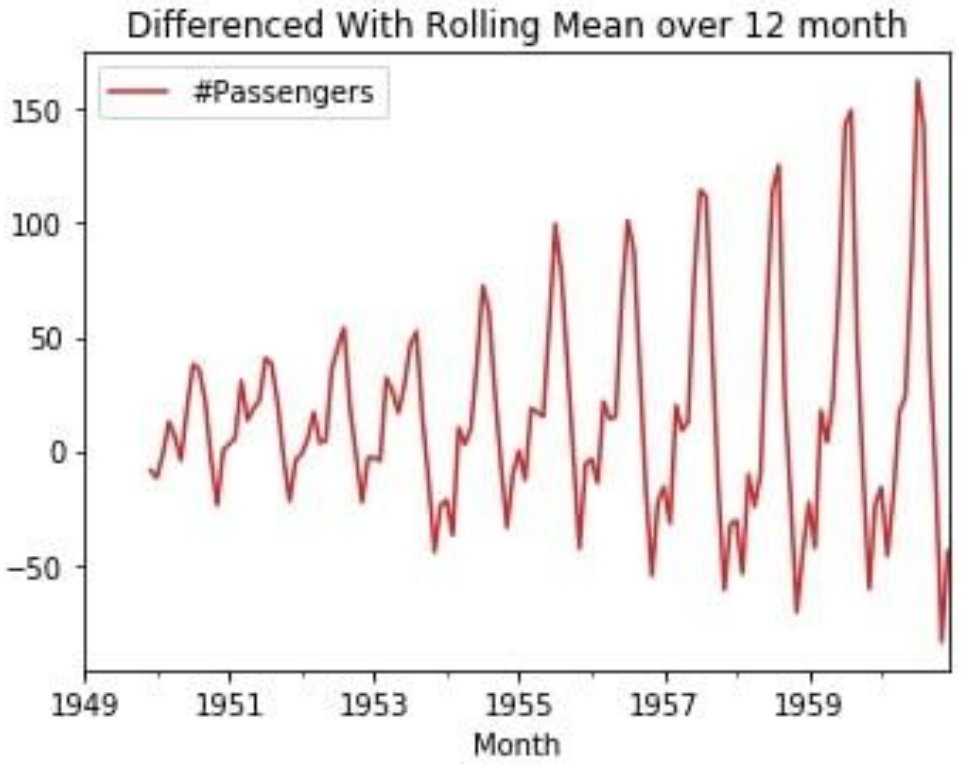
Apply power transformation in data same way as that of log transformation to remove trend

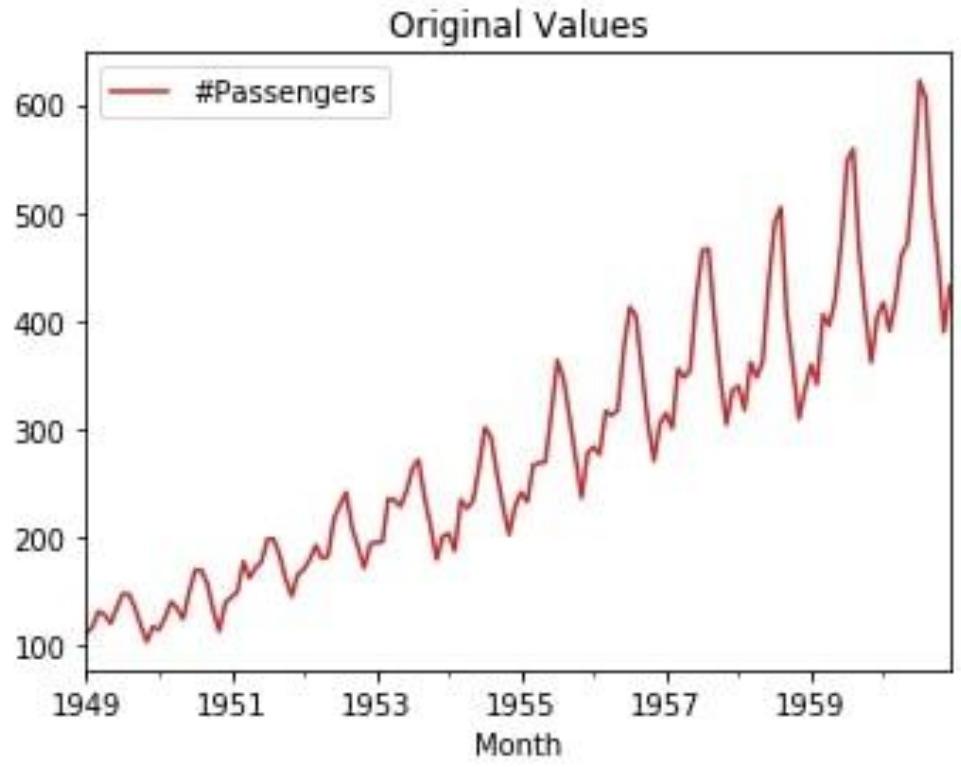




Applying moving window functions

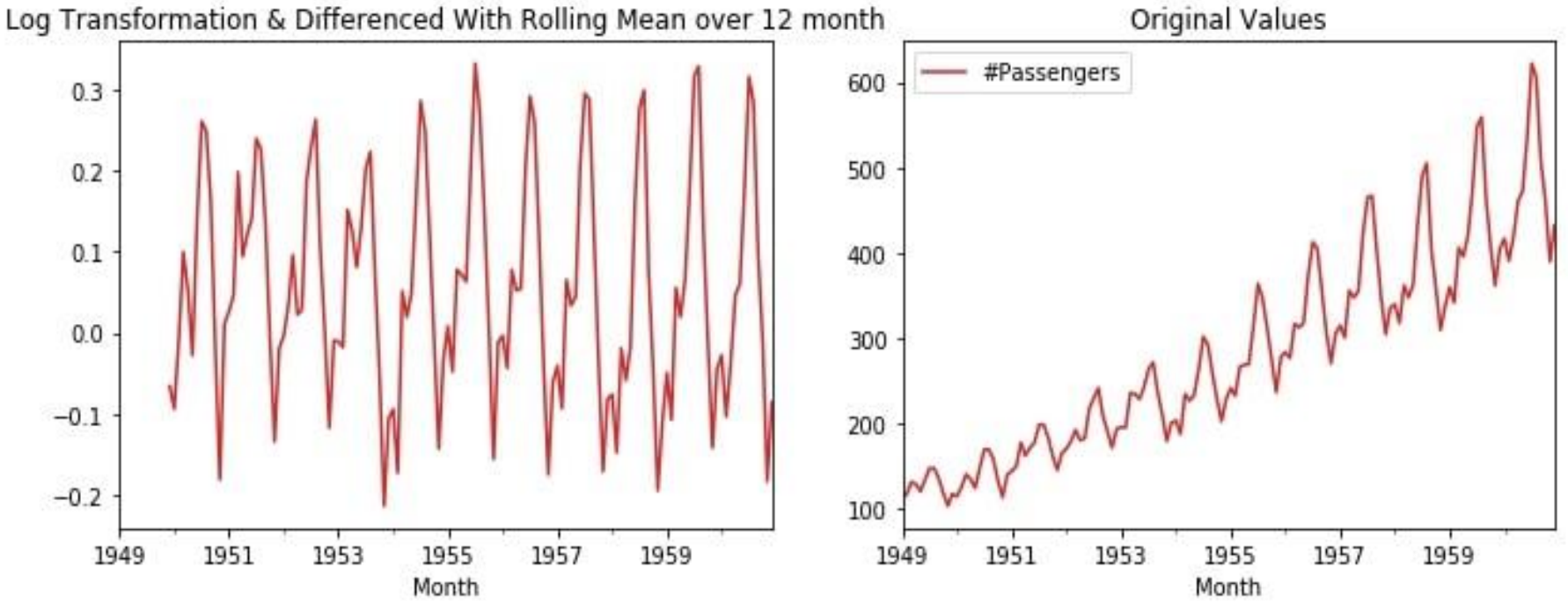
We can calculate rolling mean over a period of 12 months and subtract it from original time-series to get de-trended time-series.





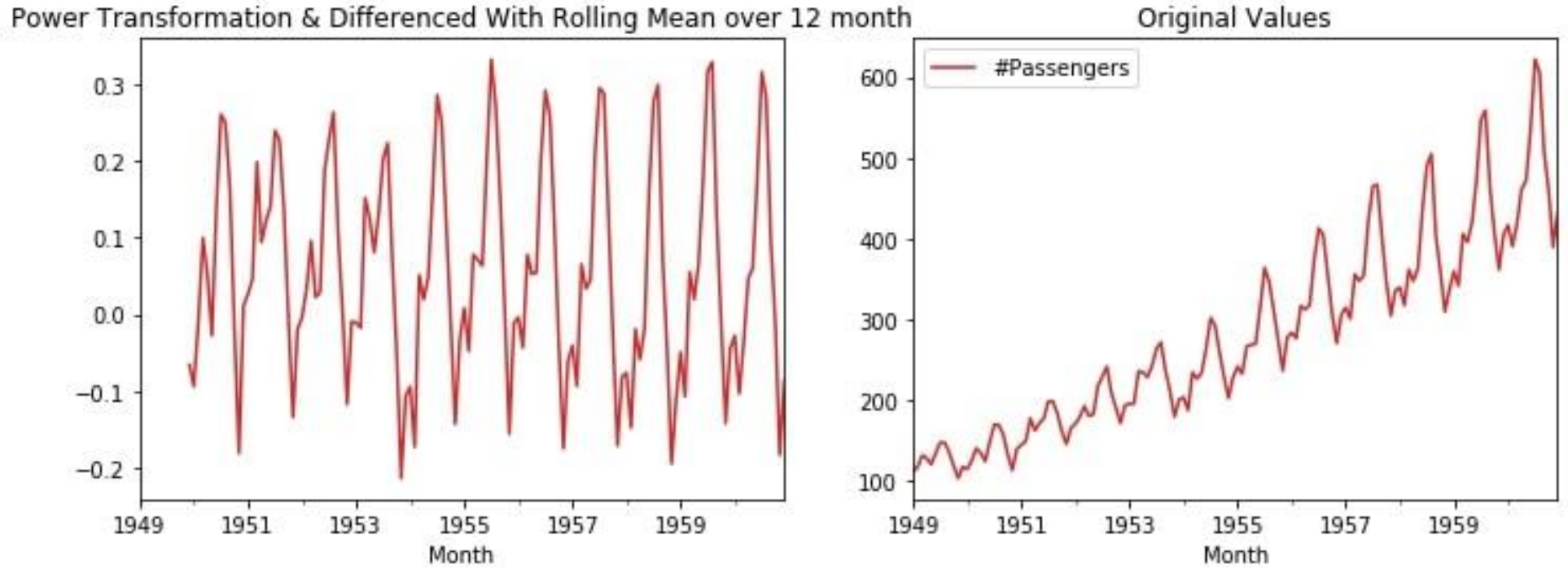
Applying Moving Window Function on Log Transformed Time-Series

We can apply more than one transformation as well. We'll first apply log transformation to time-series, then take a rolling mean over a period of 12 months and then subtract rolled time-series from log-transformed time-series to get final time-series.



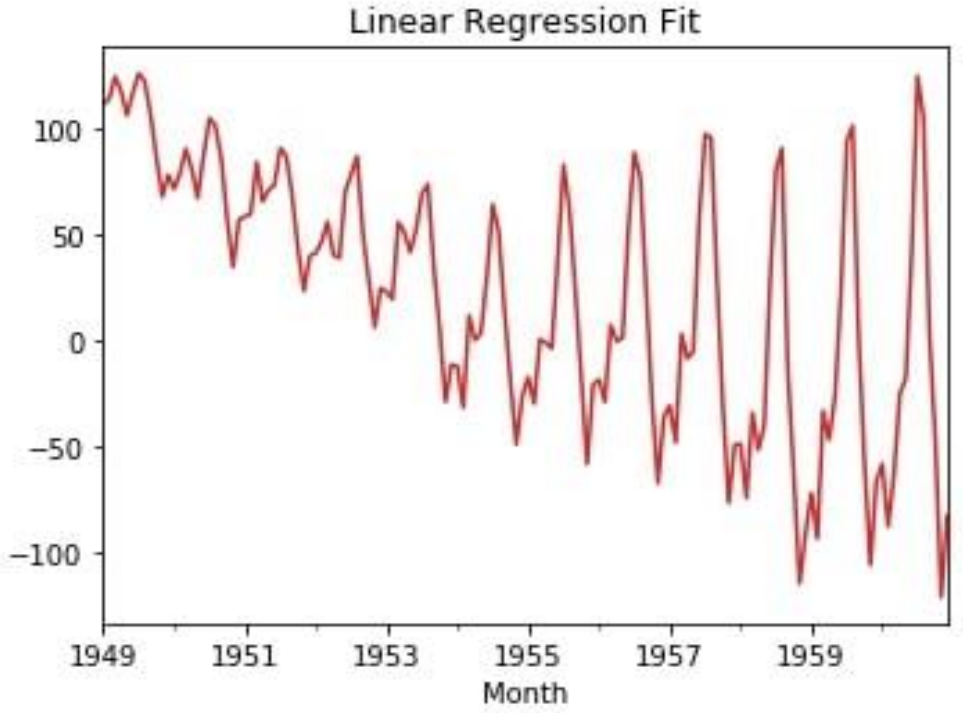
Applying Moving Window Function on Power Transformed Time-Series

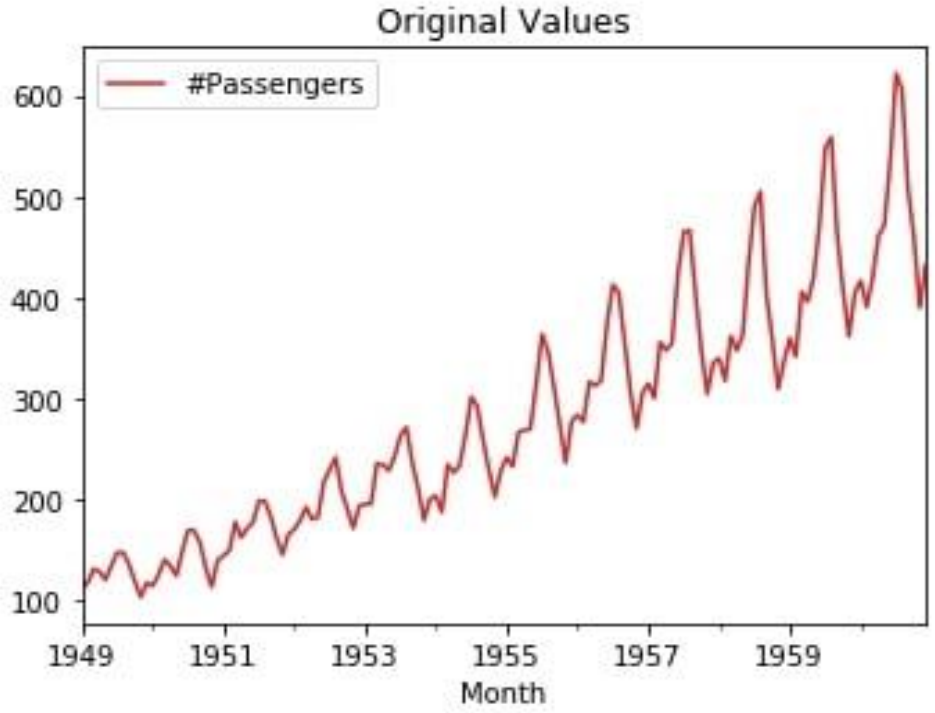
We can apply more than one transformation as well. We'll first apply power transformation to time-series, then take a rolling mean over a period of 12 months and then subtract rolled time-series from power-transformed time-series to get final time-series.



Applying Linear Regression to Remove Trend

We can also apply a linear regression model to remove the trend. Below we are fitting a linear regression model to our time-series data. We are then using a fit model to predict time-series values from beginning to end. We are then subtracting predicted values from original time-series to remove the trend.





Remove Seasonality

The seasonality represents variations in measured value which repeats over the same time interval regularly. If we notice that particular variations in value are happening every week, month, quarter or half-yearly then we can say that time series has some kind of seasonality.

How to remove seasonality from time-series data?

- Average de-trended values.
- Differencing a time-series.

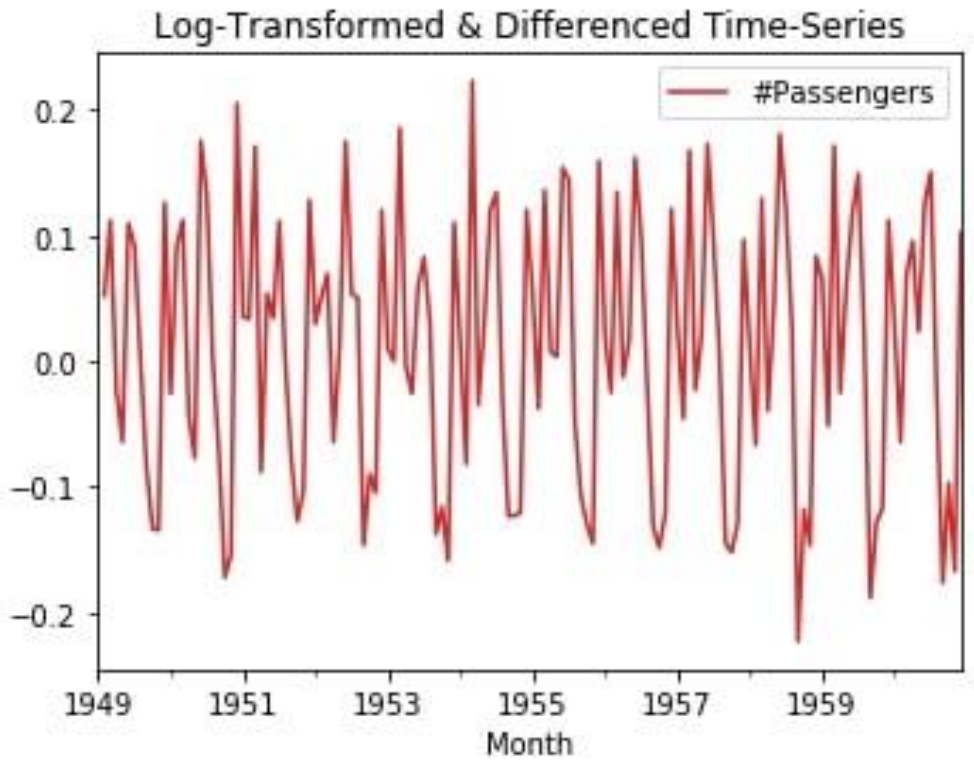
Average de-trended values

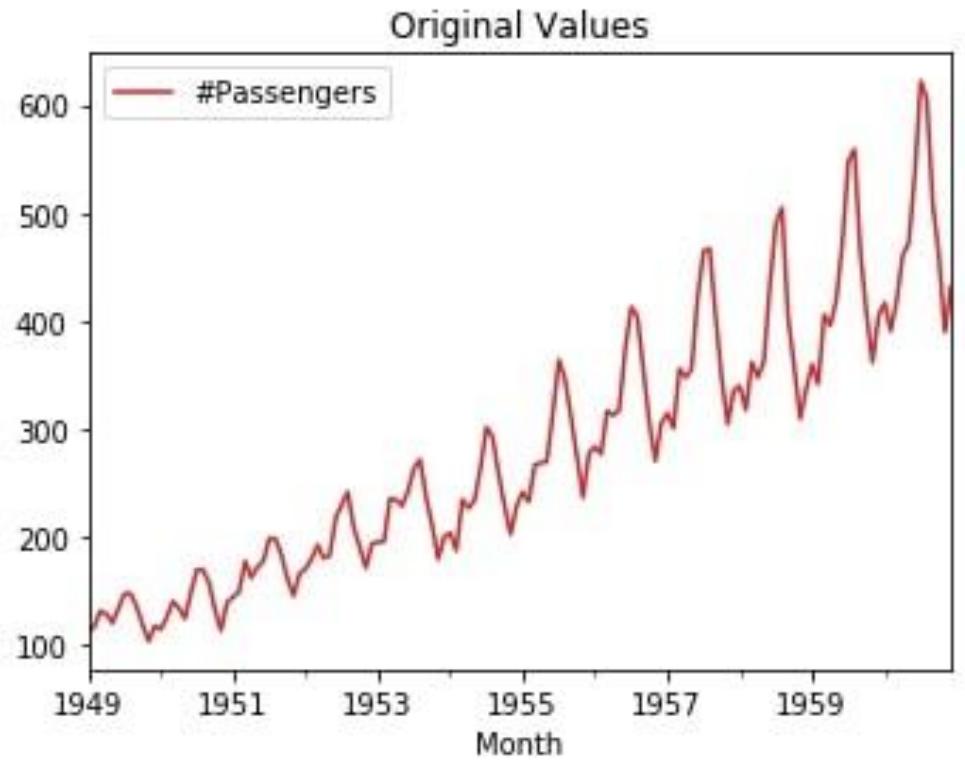
Average de-trended values is a method for removing seasonality from time series data. It involves the following steps:

- Detrend the time series: This means removing the trend from the data. The trend can be removed using a variety of methods, such as linear regression or polynomial regression.
- Calculate the average of the de-trended values: This gives the average value of the data after the trend has been removed.
- Subtract the average value from the de-trended values: This leaves the seasonal component of the data..

Differencing Over Log Transformed Time-Series

Apply differencing to log-transformed time-series by shifting its value by 1 period and subtracting it from original log-transformed time-series.





Exploratory Data Analysis for Time Series

Familiar Methods

- Plotting
- Histograms
- Scatter Plots

Time series specific exploratory methods

- Understanding Stationarity,
- Window Functions,
- Self correlation,
- Spurious correlations

Exploratory Data Analysis for Time Series

Familiar Methods

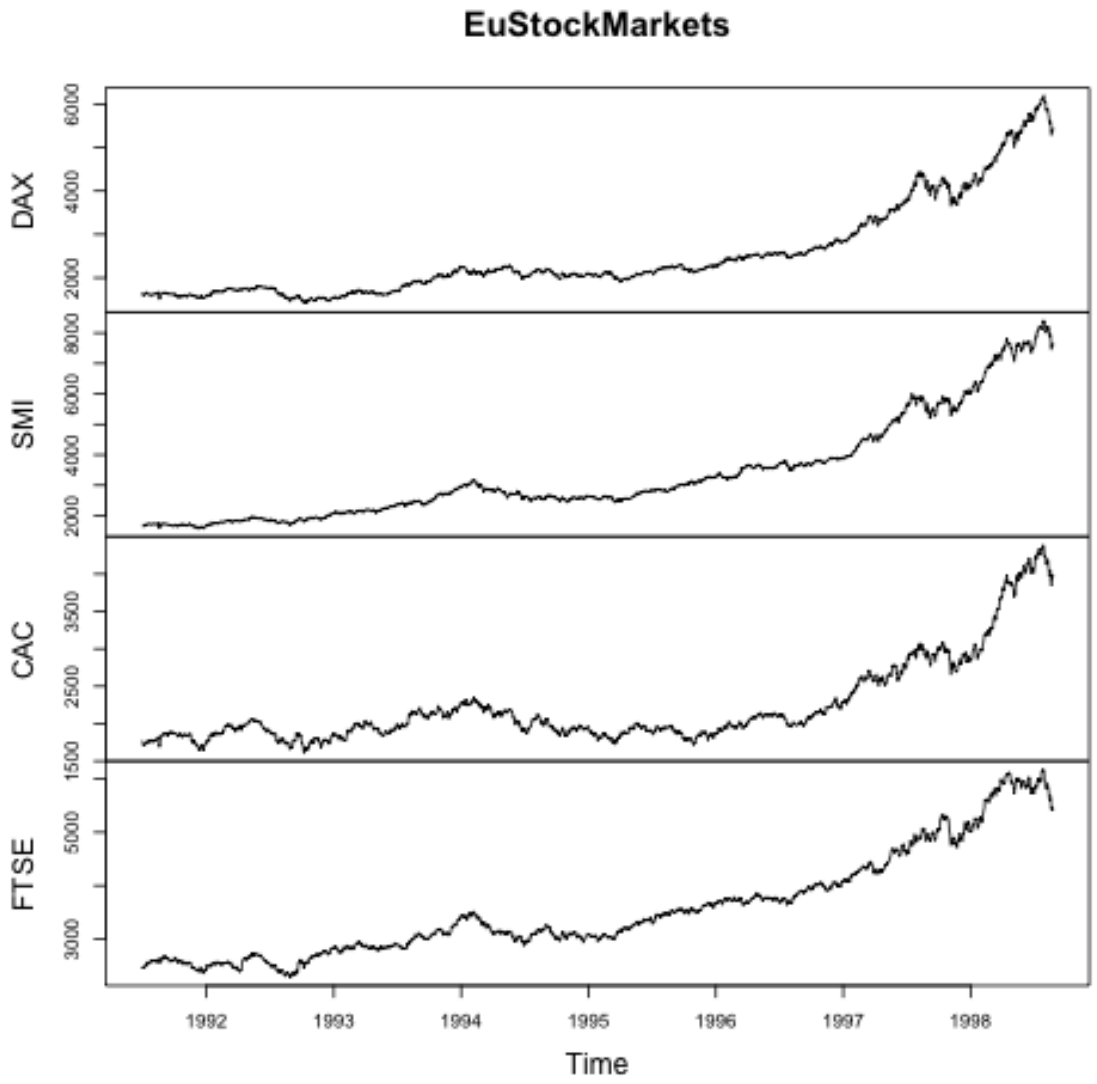
- Plotting
- Histograms
- Scatter Plots

Time series specific exploratory methods

- Understanding Stationarity,
- Window Functions,
- Self correlation,
- Spurious correlations

Plotting

- Line graph or line plot, uses lines to connect individual data points. A line graph displays quantitative values over a specified time interval.



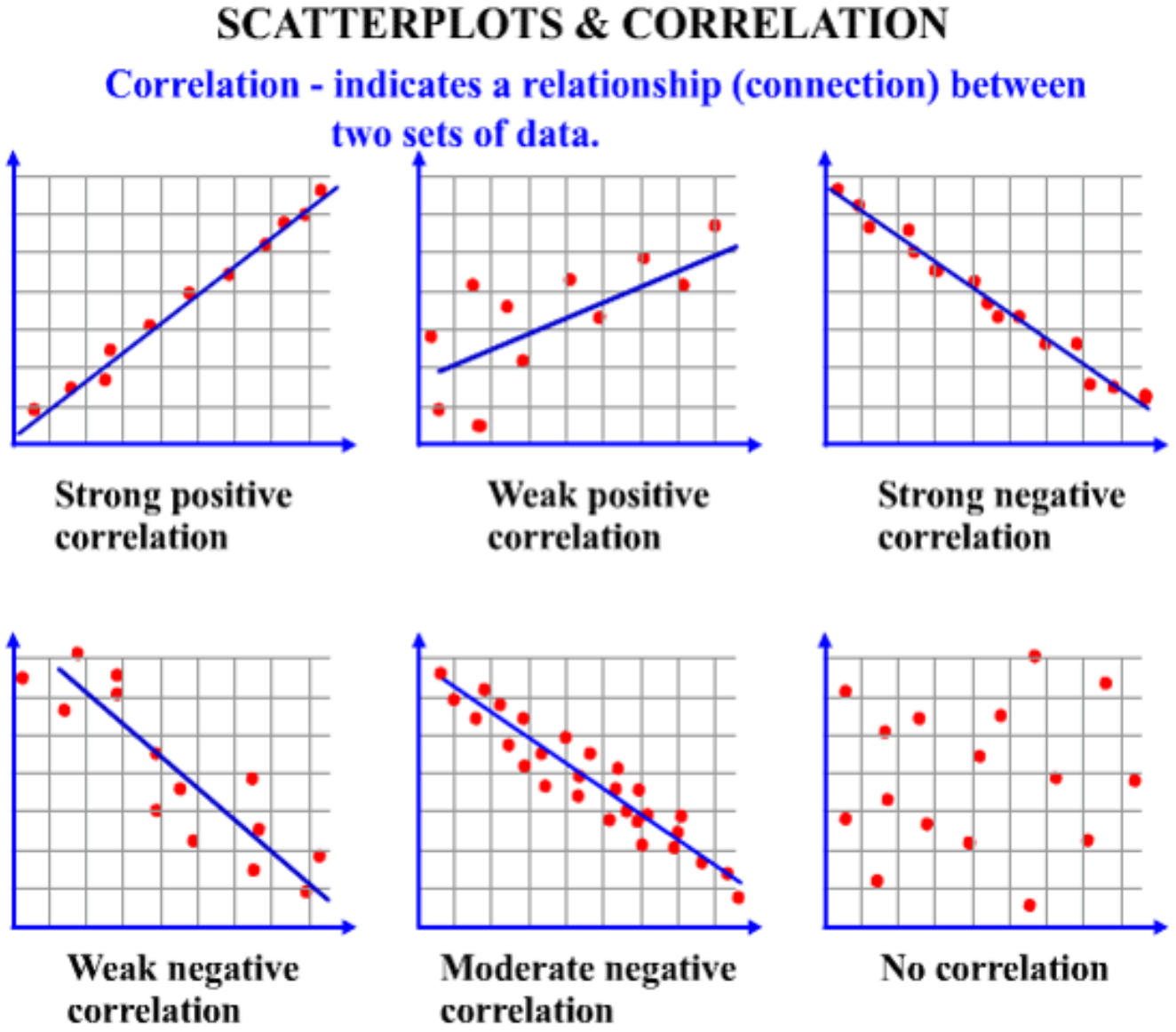
A simple plot of the time series data.

Histograms

- It is used to analyze the density of underlying distribution of data more precisely probability distribution of data. The plotting of the histogram depends upon ‘bins’ i.e dividing the entire range into series of interval then based upon the number of values present inside a range of a bin the height of the bar of that bin is determined.

Scatter Plots

- A scatterplot displays the strength, direction, and form of the relationship between two quantitative variables. A correlation coefficient measures the strength of that relationship.
- Types
  - No correlation
  - Positive correlation
  - Negative correlation



Correlation Coefficents

Formula

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

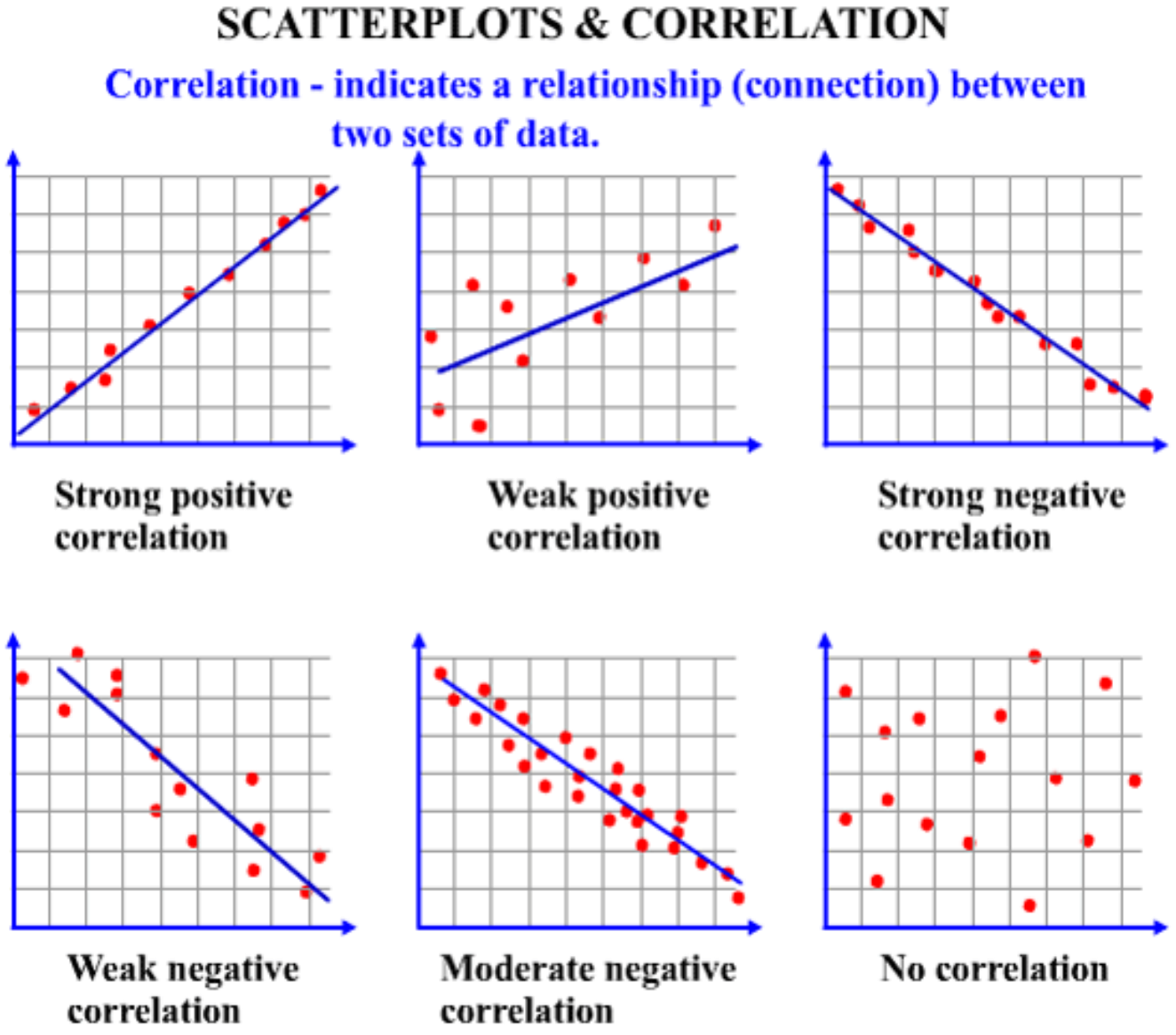
r = correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable



Solve the following:

Subject	Age x	Glucose Level y
1	43	99
2	21	65
3	25	79
4	42	75
5	57	87
6	59	81

- **Question**: Find the value of the correlation coefficient from the following table:

Autocorrelation

The correlation of a series with its own lagged values is called ***autocorrelation*** or ***serial correlation***.

- The first autocorrelation of Y_t is $\text{corr}(Y_t, Y_{t-1})$
- The first ***autocovariance*** of Y_t is $\text{cov}(Y_t, Y_{t-1})$
- Thus

$$\text{corr}(Y_t, Y_{t-1}) = \frac{\text{cov}(Y_t, Y_{t-1})}{\sqrt{\text{var}(Y_t)\,\text{var}(Y_{t-1})}} = \rho_1$$

Autocorrelation (Serial Correlation) and Autocovariance

The j^{th} autocovariance of a series Y_t is the covariance between Y_t and its j^{th} lag, Y_{t-j}, and the j^{th} autocorrelation coefficient is the correlation between Y_t and Y_{t-j}. That is,

$$j^{\text{th}} \text{ autocovariance} = \text{cov}(Y_t, Y_{t-j})$$

$$j^{\text{th}} \text{ autocorrelation} = \rho_j = \text{corr}(Y_t, Y_{t-j}) = \frac{\text{cov}(Y_t, Y_{t-j})}{\sqrt{\text{var}(Y_t)\text{var}(Y_{t-j})}}.$$

The j^{th} autocorrelation coefficient is sometimes called the j^{th} serial correlation coefficient.

Exploratory Data Analysis for Time Series

Familiar Methods

- Plotting
- Histograms
- Scatter Plots

Time series specific exploratory methods

- Understanding Stationarity,
- Window Functions,
- Self correlation,
- Spurious correlations

Understanding Stationarity

- **What Is a Stationary Series?**

A Stationary series is one whose statistical properties such as mean, variance, covariance, and standard deviation do not vary with time, or these stats properties are not a function of time. In other words, stationarity in Time Series also means series without a Trend or Seasonal components.

- **Why non-stationary time series data is difficult to analyze**:
  - **Non-stationary data can be more sensitive to noise.** This means that small changes in the data can have a large impact on the analysis.
  - **Non-stationary data can be more difficult to model.** This is because the statistical properties of the data are changing over time.
  - **Non-stationary data can be more difficult to forecast.** This is because the future values of the data are not likely to be constant.

Types of Stationary Series

1. **Strict Stationary** – Satisfies the mathematical definition of a stationary process. Mean, variance & covariance are not a function of time.
2. **Seasonal Stationary** – Series exhibiting seasonality.
3. **Trend Stationary** – Series exhibiting trend.

- *Note: Once the seasonality and trend are removed, the series will be strictly stationary*

How to Check Stationarity?

- **Visualization/ Intuition**

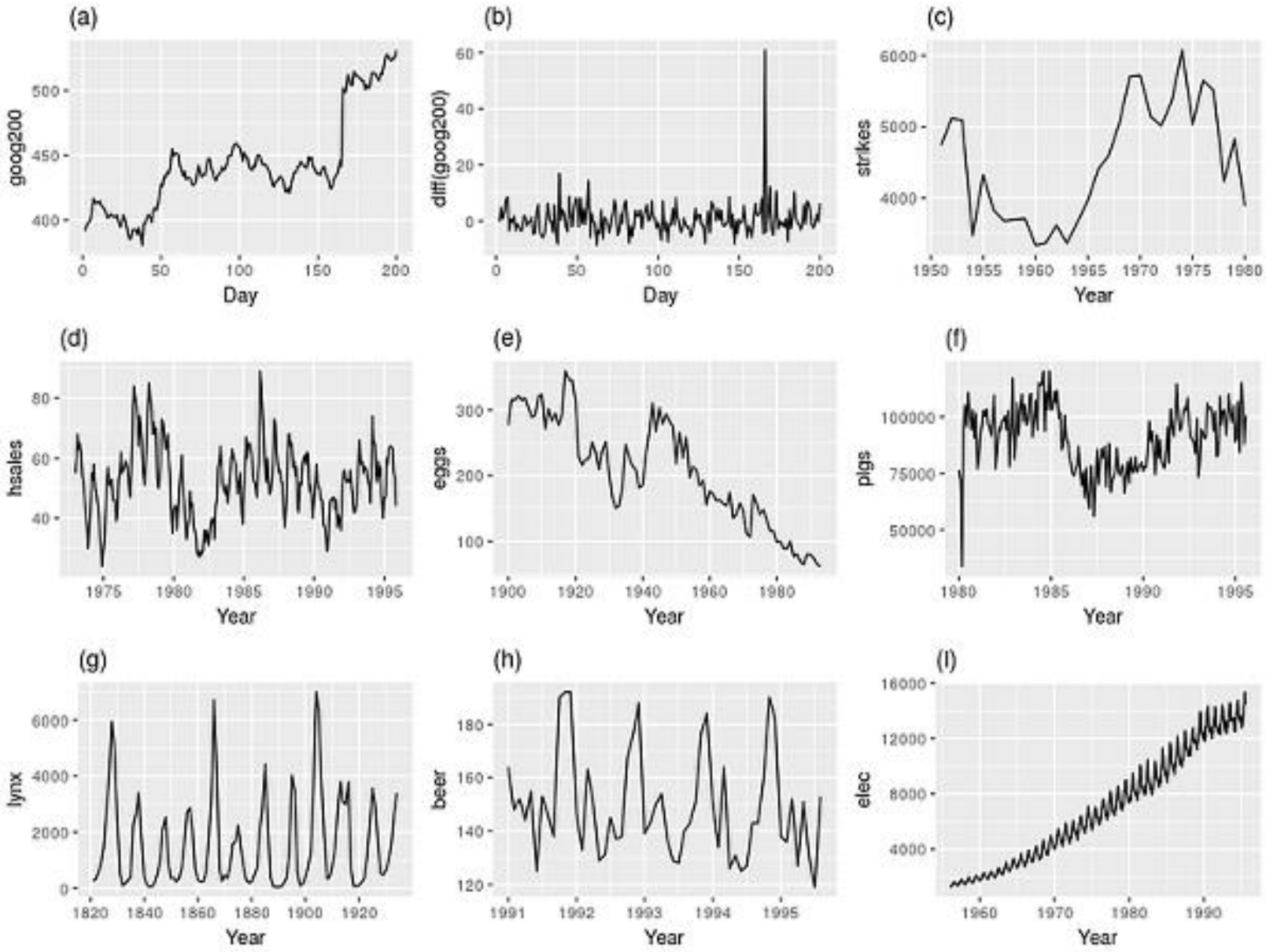


Figure 1: Nine examples of time series data; (a) Google stock price for 200 consecutive days; (b) Daily change in the Google stock price for 200 consecutive days; (c) Annual number of strikes in the US; (d) Monthly sales of new one-family houses sold in the US; (e) Annual price of a dozen eggs in the US (constant dollars); (f) Monthly total of pigs slaughtered in Victoria, Australia; (g) Annual total of lynx trapped in the McKenzie River district of north-west Canada; (h) Monthly Australian beer production; (i) Monthly Australian electricity production. [Hyndman & Athanasopoulos, 2018]

How to Check Stationarity?

- **Visualization/ Intuition**

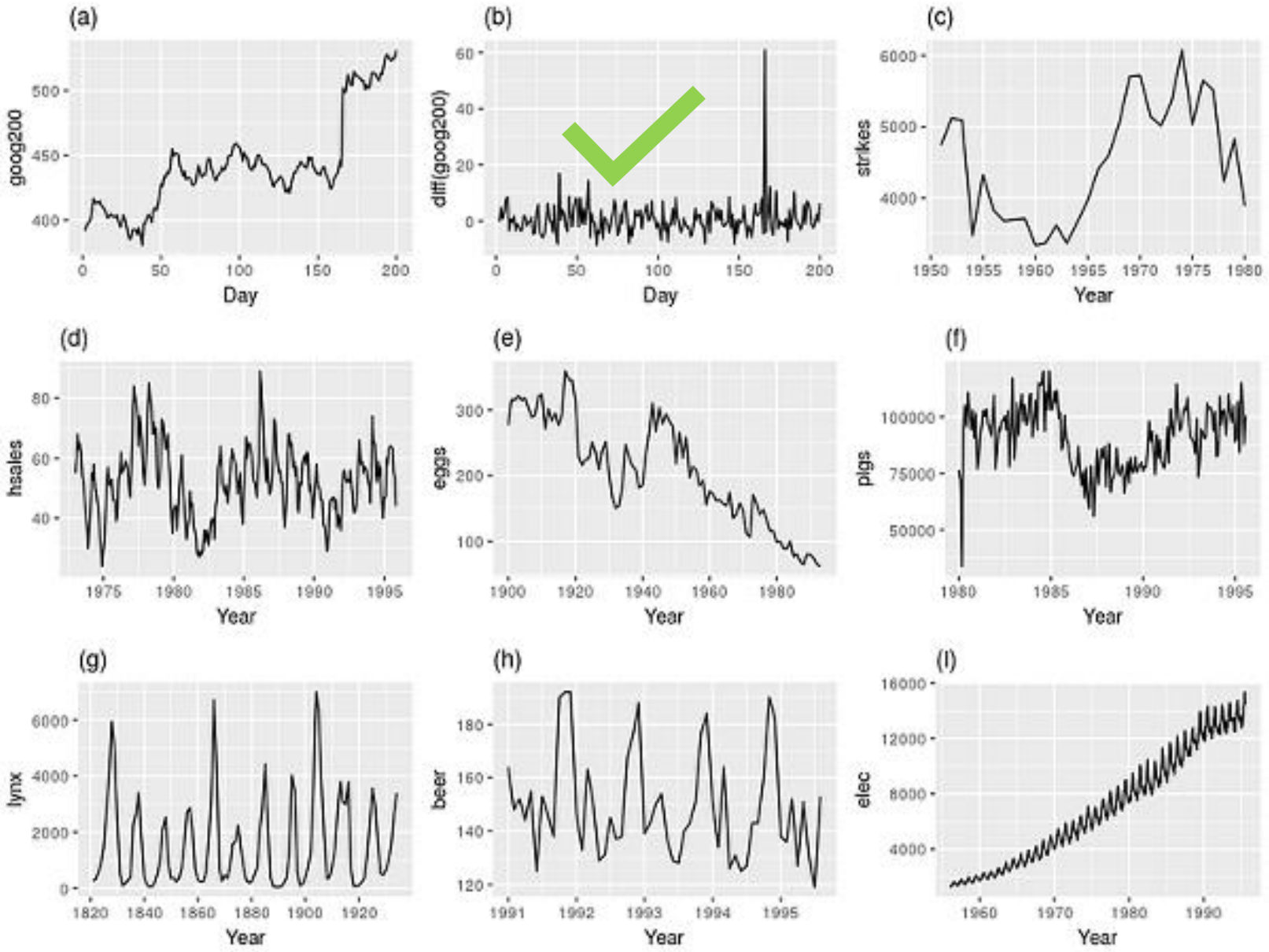


Figure 1: Nine examples of time series data; (a) Google stock price for 200 consecutive days; (b) Daily change in the Google stock price for 200 consecutive days; (c) Annual number of strikes in the US; (d) Monthly sales of new one-family houses sold in the US; (e) Annual price of a dozen eggs in the US (constant dollars); (f) Monthly total of pigs slaughtered in Victoria, Australia; (g) Annual total of lynx trapped in the McKenzie River district of north-west Canada; (h) Monthly Australian beer production; (i) Monthly Australian electricity production. [Hyndman & Athanasopoulos, 2018]

Popular statistical tests

Augmented Dickey-Fuller (ADF) Test

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

Augmented Dickey-Fuller (ADF) Test

- ADF test belongs to a category of tests called '*Unit Root Test*', which is the proper method for testing the stationarity of a time series.
- In probability theory and statistics, a **unit root** is a feature of some stochastic processes (such as random walks) that can cause problems in statistical inference involving time series models. ***In simple terms, the unit root is non-stationary*** but does not always have a trend component.

Random walk model

The differenced series is the *change* between consecutive observations in the original series, and can be written as

$$y'_t = y_t - y_{t-1}.$$

The differenced series will have only $T - 1$ values, since it is not possible to calculate a difference y'_1 for the first observation.

When the differenced series is white noise, the model for the original series can be written as

$$y_t - y_{t-1} = \varepsilon_t,$$

where ε_t denotes white noise. Rearranging this leads to the "random walk" model

$$y_t = y_{t-1} + \varepsilon_t.$$

Random walk models are widely used for non-stationary data, particularly financial and economic data. Random walks typically have:

- long periods of apparent trends up or down
- sudden and unpredictable changes in direction.

The forecasts from a random walk model are equal to the last observation, as future movements are unpredictable, and are equally likely to be up or down.

A closely related model allows the differences to have a non-zero mean. Then

$$y_t - y_{t-1} = c + \varepsilon_t \quad \text{or} \quad y_t = c + y_{t-1} + \varepsilon_t \ .$$

The value of c is the average of the changes between consecutive observations. If c is positive, then the average change is an increase in the value of y_t. Thus, y_t will tend to drift upwards. However, if c is negative, y_t will tend to drift downwards.

This is the model behind the drift method,

Second-order differencing

Occasionally the differenced data will not appear to be stationary and it may be necessary to difference the data a second time to obtain a stationary series:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

In this case, y''_t will have $T - 2$ values. Then, we would model the "change in the changes" of the original data. In practice, it is almost never necessary to go beyond second-order differences.

Seasonal differencing

A seasonal difference is the difference between an observation and the previous observation from the same season. So

$$y'_t = y_t - y_{t-m},$$

where $m =$ the number of seasons. These are also called "lag-m differences", as we subtract the observation after a lag of m periods.

If seasonally differenced data appear to be white noise, then an appropriate model for the original data is

$$y_t = y_{t-m} + \varepsilon_t.$$

Forecasts from this model are equal to the last observation from the relevant season. That is,

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.

Autoregressive models

In a multiple regression model, we forecast the variable of interest using a linear combination of predictors. In an autoregression model, we forecast the variable of interest using a linear combination of *past values of the variable*. The term *auto*regression indicates that it is a regression of the variable against itself.

Thus, an autoregressive model of order p can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

where ε_t is white noise. This is like a multiple regression but with *lagged values* of y_t as predictors. We refer to this as an **AR(p) model**, an autoregressive model of order p.

Autoregressive models are remarkably flexible at handling a wide range of different time series patterns

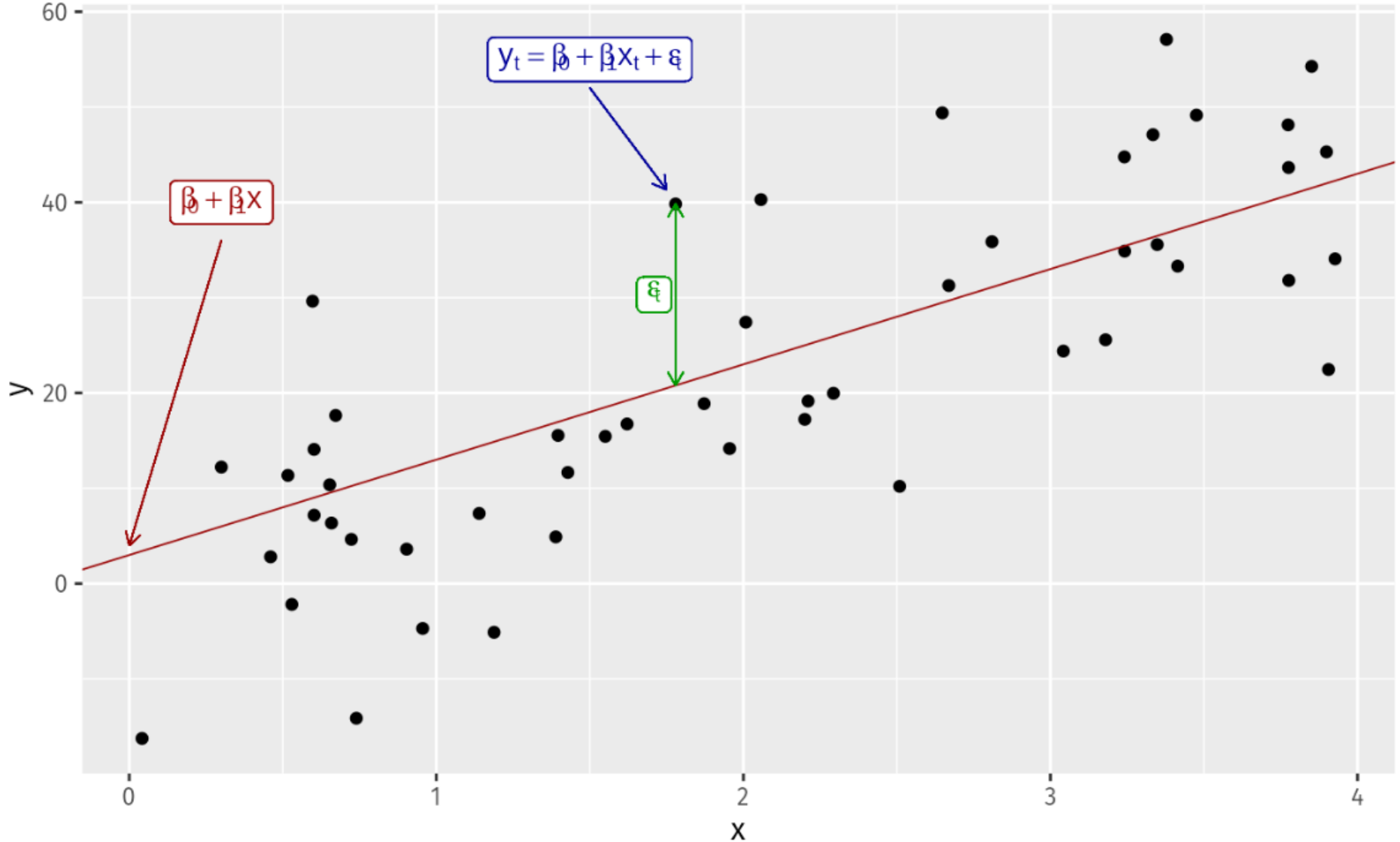
Changing the parameters $\phi_1, \ldots, \phi_p$ results in different time series patterns. The variance of the error term ε_t will only change the scale of the series, not the patterns.

Simple linear regression

In the simplest case, the regression model allows for a linear relationship between the forecast variable y and a single predictor variable x:

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t.$$

The coefficients β_0 and β_1 denote the intercept and the slope of the line respectively. The intercept β_0 represents the predicted value of y when $x = 0$. The slope β_1 represents the average predicted change in y resulting from a one unit increase in x.



Notice that the observations do not lie on the straight line but are scattered around it. We can think of each observation y_t as consisting of the systematic or explained part of the model, $\beta_0 + \beta_1 x_t$, and the random "error", ε_t. The "error" term does not imply a mistake, but a deviation from the underlying straight line model. It captures anything that may affect y_t other than x_t.

Multiple linear regression

When there are two or more predictor variables, the model is called a **multiple regression model**. The general form of a multiple regression model is

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t,$$

where y is the variable to be forecast and $x_1, \ldots, x_k$ are the k predictor variables. Each of the predictor variables must be numerical. The coefficients $\beta_1, \ldots, \beta_k$ measure the effect of each predictor after taking into account the effects of all the other predictors in the model. Thus, the coefficients measure the *marginal effects* of the predictor variables.

Least squares estimation

In practice, of course, we have a collection of observations but we do not know the values of the coefficients $\beta_0, \beta_1, \ldots, \beta_k$. These need to be estimated from the data.

The least squares principle provides a way of choosing the coefficients effectively by minimising the sum of the squared errors. That is, we choose the values of $\beta_0, \beta_1, \ldots, \beta_k$ that minimise

$$\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 x_{1,t} - \beta_2 x_{2,t} - \cdots - \beta_k x_{k,t})^2.$$

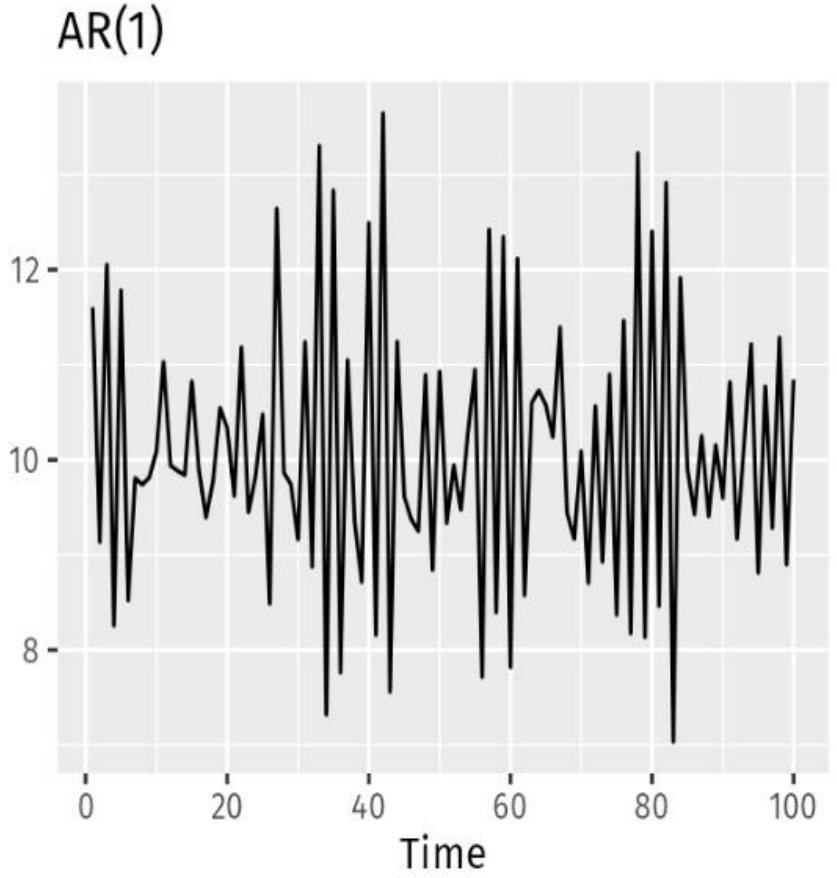
This is called **least squares** estimation because it gives the least value for the sum of squared errors. Finding the best estimates of the coefficients is often called "fitting" the model to the data, or sometimes "learning" or "training" the model.

For an AR(1) model:

- when $\phi_1 = 0$, y_t is equivalent to white noise;
- when $\phi_1 = 1$ and $c = 0$, y_t is equivalent to a random walk;
- when $\phi_1 = 1$ and $c \neq 0$, y_t is equivalent to a random walk with drift;
- when $\phi_1 < 0$, y_t tends to oscillate around the mean.

We normally restrict autoregressive models to stationary data, in which case some constraints on the values of the parameters are required.

- For an AR(1) model: $-1 < \phi_1 < 1$.



$$y_t = 18 - 0.8y_{t-1} + \varepsilon_t.$$

Non-stationary time series are related to random walks and unit roots in a few ways.

- **Random walks are non-stationary.** A random walk is a time series where the current value is equal to the previous value plus a random noise term. This means that the mean and variance of the random walk will change over time, making it non-stationary.
- **Unit roots can cause non-stationarity.** A unit root is a characteristic of some stochastic processes (such as random walks) that can cause problems in statistical inference involving time series models. A linear stochastic process has a unit root if 1 is a root of the process's characteristic equation. Such a process is non-stationary but does not always have a trend. If the other roots of the characteristic equation lie inside the unit circle—that is, have a modulus (absolute value) less than one—then the first difference of the process will be stationary; otherwise, the process will need to be differenced multiple times to become stationary.
- **Non-stationary time series can be transformed to be stationary by differencing.** This means taking the difference between the current value of the series and the previous value. Differencing the series removes the trend and seasonality from the series, making it stationary.

Dickey-Fuller Test

Definition

$$y_t = \phi y_{t-1} + u_t$$

- The test examines the value of ϕ. In particular, it tests the null hypothesis that $\phi = 1$ against the alternative that $\phi < 1$. In practice, the test implores the use of the differenced form.

$H_0 : \phi = 1$ (Non-stationary)

$H_a : \phi < 1$ (Stationary Series)

Dickey-Fuller Test

Definition

- The test examines the value of ϕ. In particular, it tests the null hypothesis that $\phi = 1$ against the alternative that $\phi < 1$. In practice, the test implores the use of the differenced form.
  - $\Delta y_t = \psi y_{t-1} + u_t$
  - We derive this by using the first $AR(1)$ ($y_t = \phi y_{t-1} + u_t$) and it's immediate lag. If we subtract the immediate lag of the y_t, i.e. y_{t-1} to the both sides of the equation, we are essentially getting this difference equation.
  - In this case, $\psi = \phi - 1$

Dickey-Fuller Test

Definition

- Likewise, the test can be extended further to accommodate the inclusion of an intercept and a deterministic time trend.
  - $\Delta y_t = \psi y_{t-1} + \mu + \beta t + u_t$
- As with the base difference equation, the null and alternative hypothesis are formulated in the manner
  - $H_0 : \psi = 0$
  - $H_a : \psi \neq 0$

$\psi = \phi - 1$

if $H_0 : \psi = 0$

$0 = \phi - 1$

$\phi = 1$ (N.S)

if $H_a : \psi \neq 0$

$-0.5 = \phi - 1$

$0.5 = \phi$ (S)

Augmented Dickey Fuller Test

Definition

- So far, the Dickey Fuller test assumes that u_t is a white noise error term. However, if u_t is autocorrelated, we would need a drift version of the test which allows for higher order lags.
- Running the original Dickey Full test in this case would result in an oversized test suggesting that the true size of the test which is the proportion of times a correct null hypothesis is incorrectly rejected would be higher than the normal sized used.
- As such, we 'augment' the test using p lags of the original series
- $$\Delta y_t = \psi y_{t-1} + \mu + \alpha t + \sum_{i=1}^{p} \beta_i \Delta y_{t-i} + u_t$$

Kwiatkowski Phillips Schmidt and Shin Test (KPSS)

- The test assumes that the time series can be divided into a deterministic trend, a random walk and a stationary error. This means time series can be illustrated as :

$$Y_t = \beta\, t + (r_t + \alpha) + e_t$$

where:

- $r_t = r_{t-1} + u_t$ is a random walk, the initial value $r_0 = \alpha$ serves as an intercept,
- t is the time index,
- u_t are independent identically distributed $(0, \sigma_u^2)$.

r_t

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

Definition

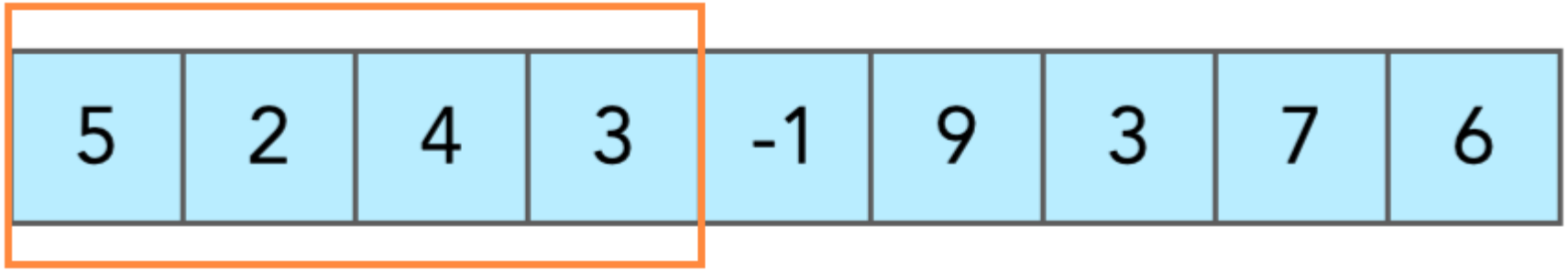
- The null and alternative hypothesis are
  - $H_0 : \sigma_u^2 = 0$
  - $H_a : \sigma_u^2 > 0$
- If it was found that $\sigma_u^2 = 0$, it means that r_t is just a constant and reduces to a trend r. Therefore, y_t is trend stationary. If however, the variance is significantly different from zero, then r varies over time suggesting that y_t is not stationary.

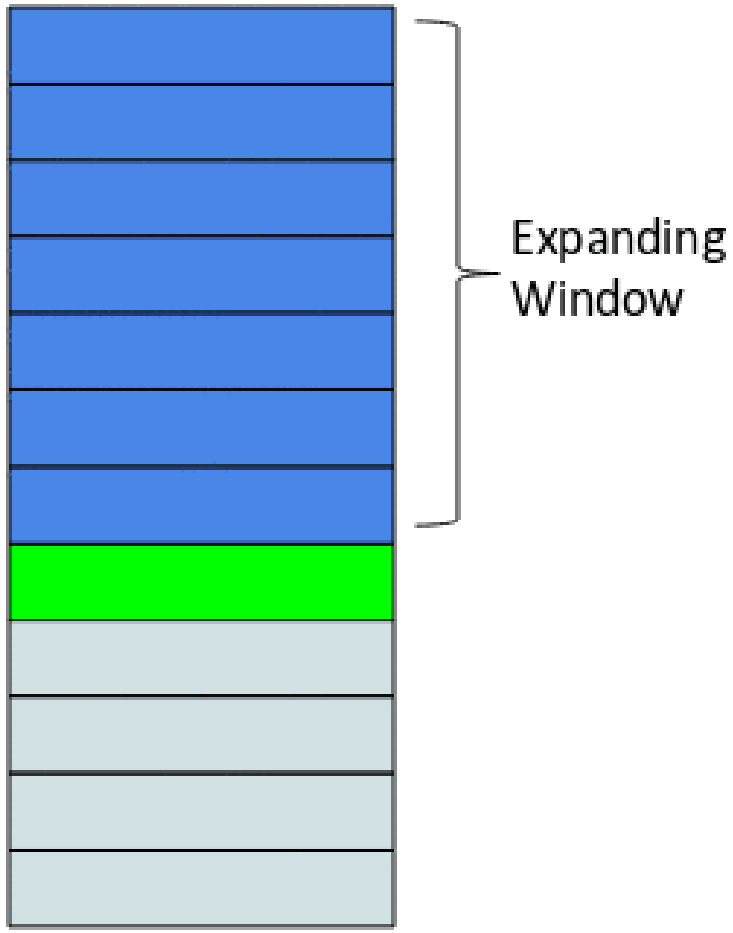
Applying Window Functions

- Rolling Window
- Expanding window
- Custom rolling functions

Rolling windows

- A window that is sliding with every next point, the features generated using this method are called the ‘rolling window’ features





Expanding window

- with every step, the size of the window increases by one as it takes into account every new value in the series.

Custom rolling functions

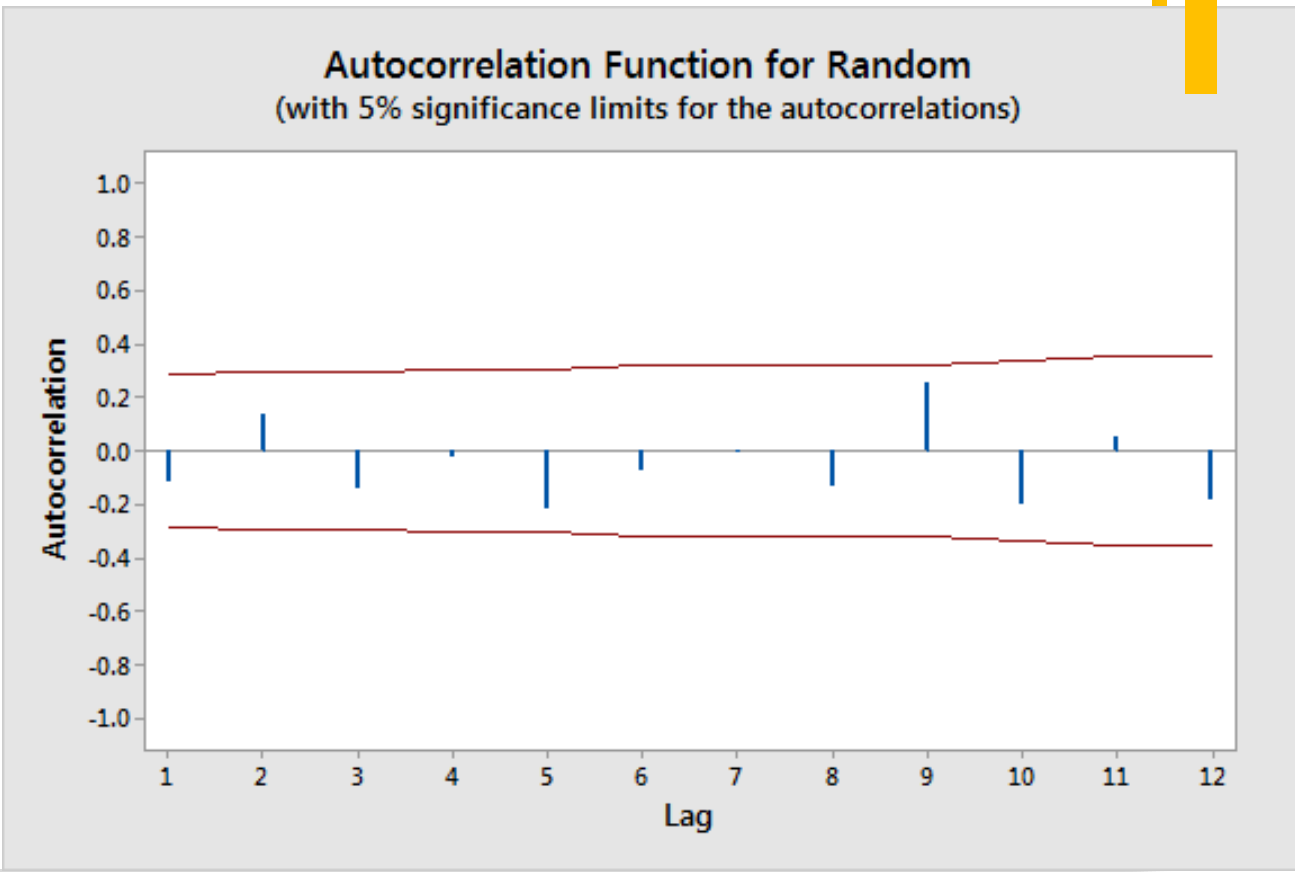
- In practice, this is something you are likely to see when analyzing time series domains that have known underlying fundamental laws of behavior or useful heuristics that are necessary for proper analysis.

Self correlation/Autocorrelation Function (ACF)

- ACF is used to identify which lags have significant correlations, understand the patterns and properties of the time series, and then use that information to model the time series data. From the ACF, you can assess the randomness and stationarity of a time series. You can also determine whether trends and seasonal patterns are present.
- In an ACF plot, each bar represents the size and direction of the correlation. Bars that extend across the red line are statistically significant.

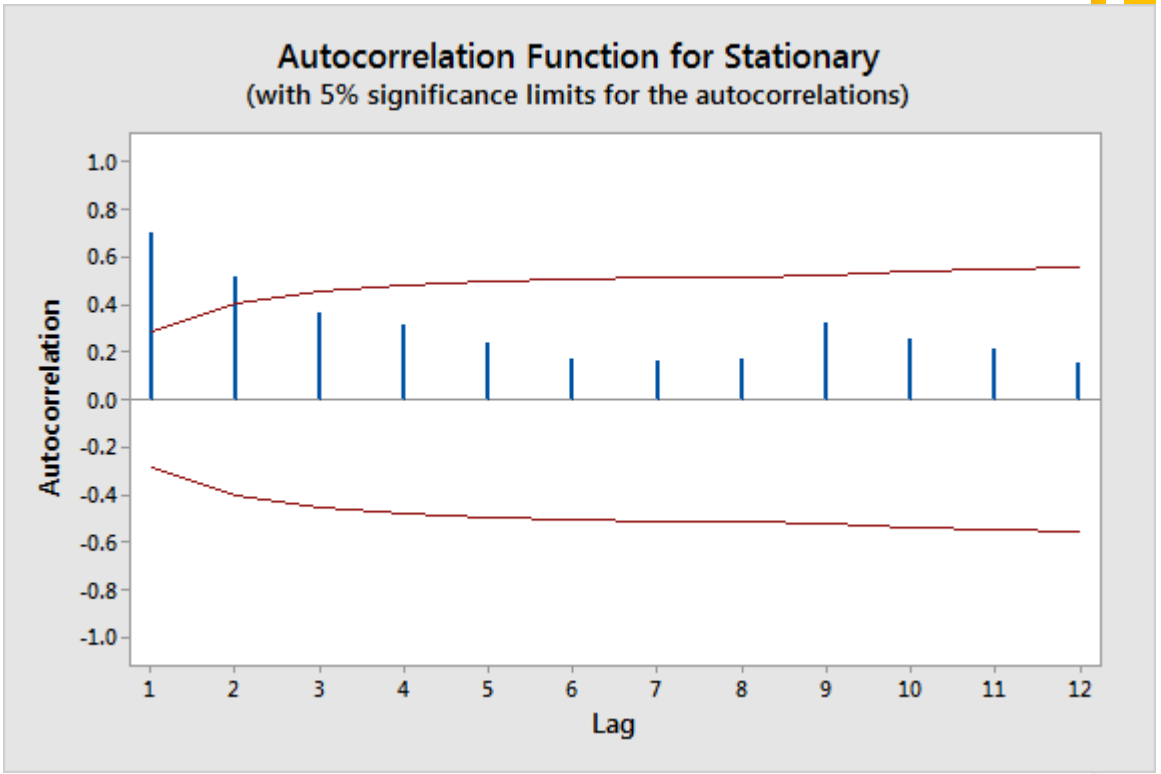
ACF of Randomness/White Noise

- For random data, autocorrelations should be near zero for all lags. Analysts also refer to this condition as white noise. Non-random data



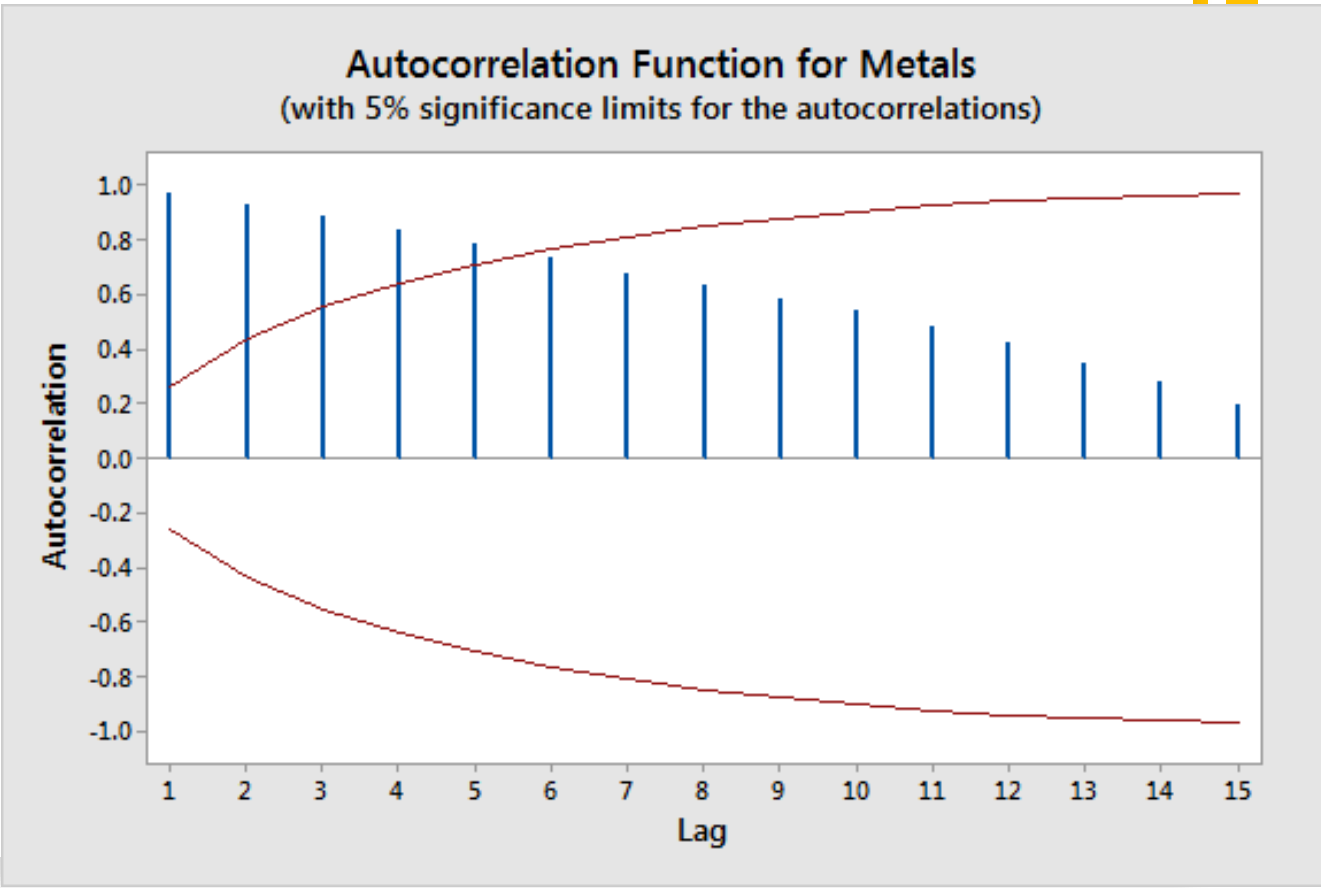
ACF of Stationarity

- Stationarity means that the time series does not have a trend, has a constant variance, a constant autocorrelation pattern, and no seasonal pattern. The autocorrelation function declines to near zero rapidly for a stationary time series. In contrast, the ACF drops slowly for a non-stationary time series



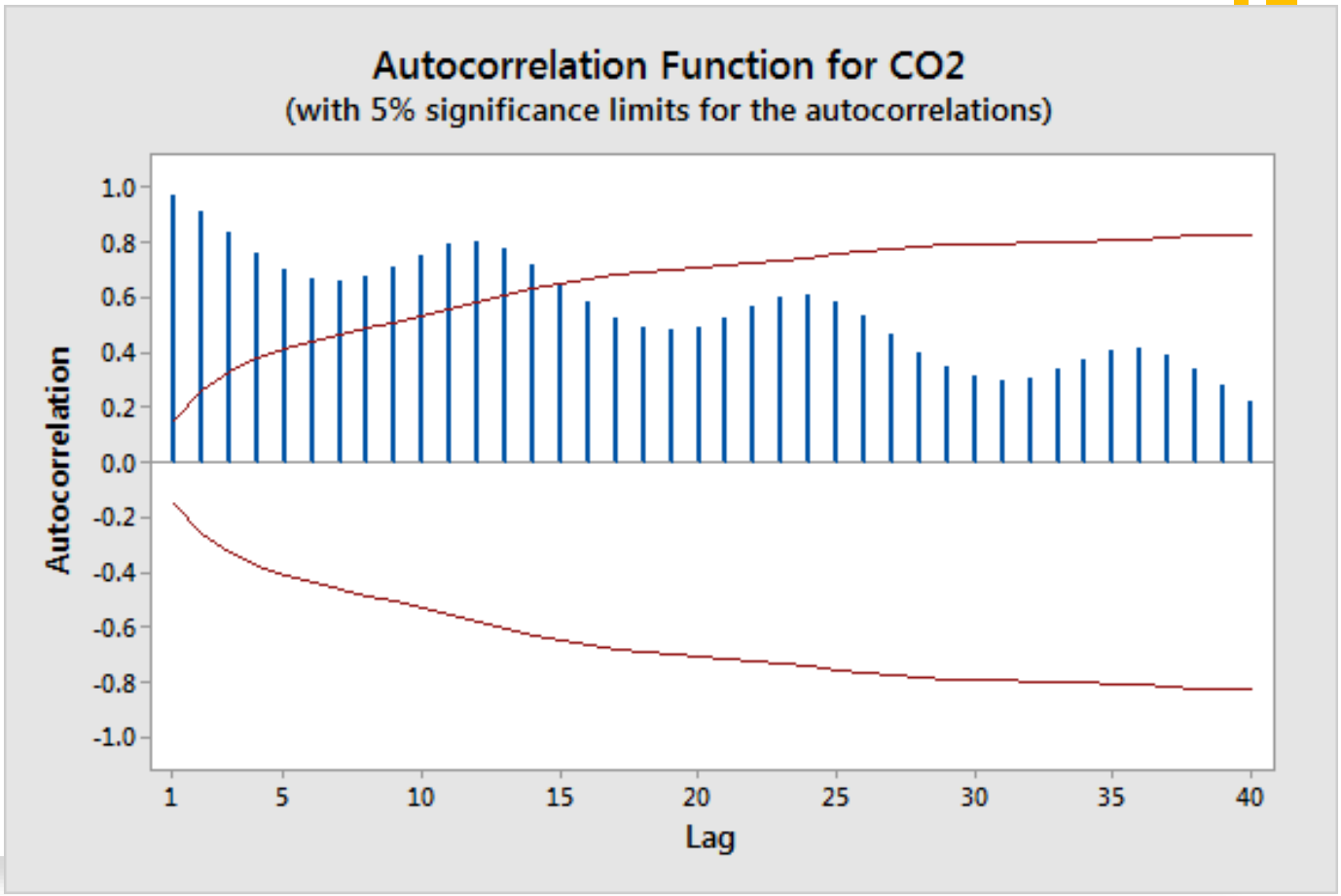
ACF of Trends

- When trends are present in a time series, shorter lags typically have large positive correlations because observations closer in time tend to have similar values. The correlations taper off slowly as the lags increase.



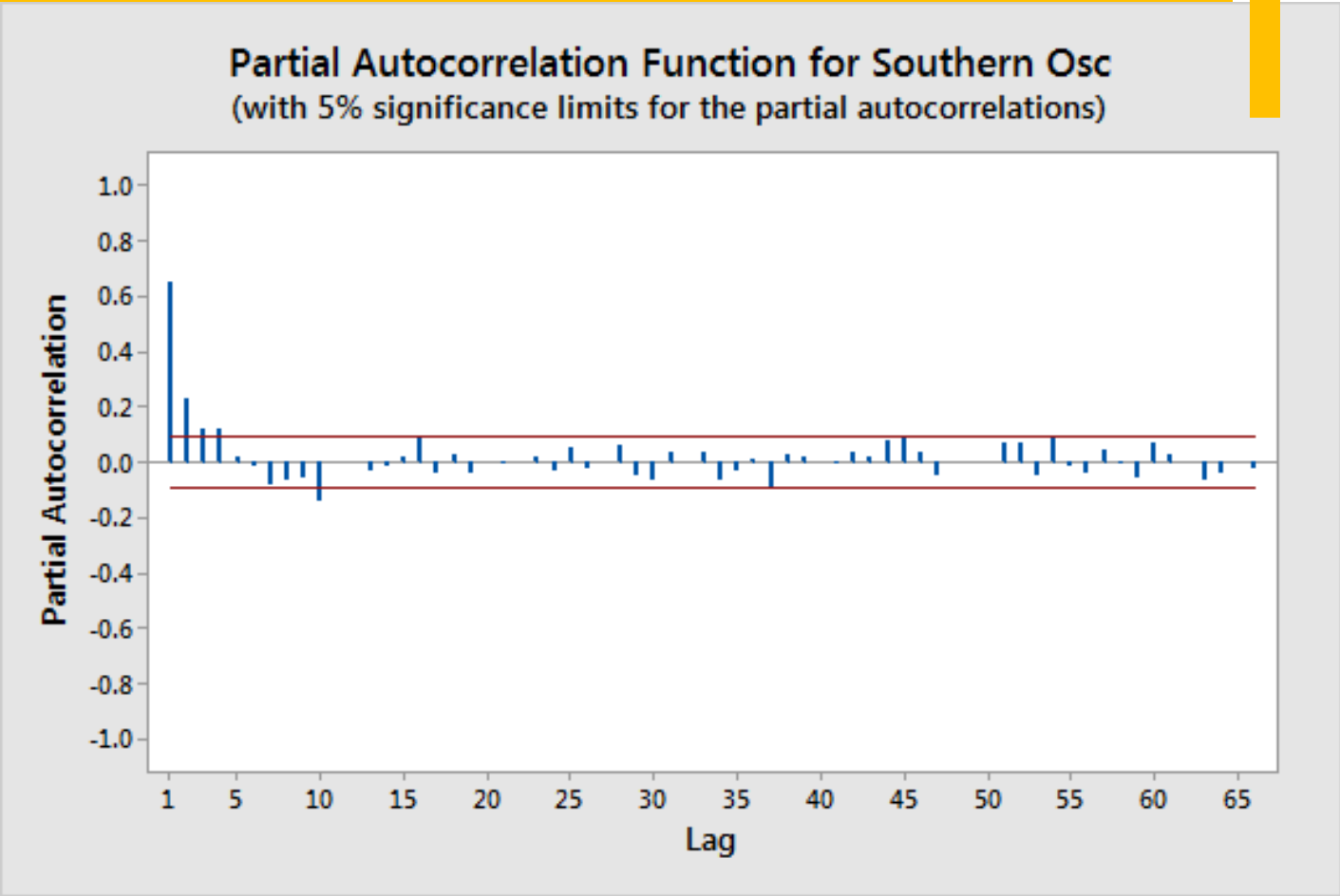
ACF of Seasonality

- When seasonal patterns are present, the autocorrelations are larger for lags at multiples of the seasonal frequency than for other lags..



Partial auto correlation function (PACF)

- It is similar to the ACF except that it displays only the correlation between two observations that the shorter lags between those observations do not explain.
- For example, the partial autocorrelation for lag 3 is only the correlation that lags 1 and 2 do not explain. In other words, the partial correlation for each lag is the unique correlation between those two observations after partialling out the intervening correlations.



Partial auto correlation function (PACF)

For a time series, the partial autocorrelation between x_t and x_{t-h} is defined as the conditional correlation between x_t and x_{t-h}, conditional on $x_{t-h+1}, \ldots, x_{t-1}$, the set of observations that come between the time points t and $t-h$.

- The 1st order partial autocorrelation will be defined to equal the 1st order autocorrelation.
- The 2nd order (lag) partial autocorrelation is

$$\frac{\text{Covariance}(x_t, x_{t-2}|x_{t-1})}{\sqrt{\text{Variance}(x_t|x_{t-1})\text{Variance}(x_{t-2}|x_{t-1})}}$$

This is the correlation between values two time periods apart conditional on knowledge of the value in between. (By the way, the two variances in the denominator will equal each other in a stationary series.)

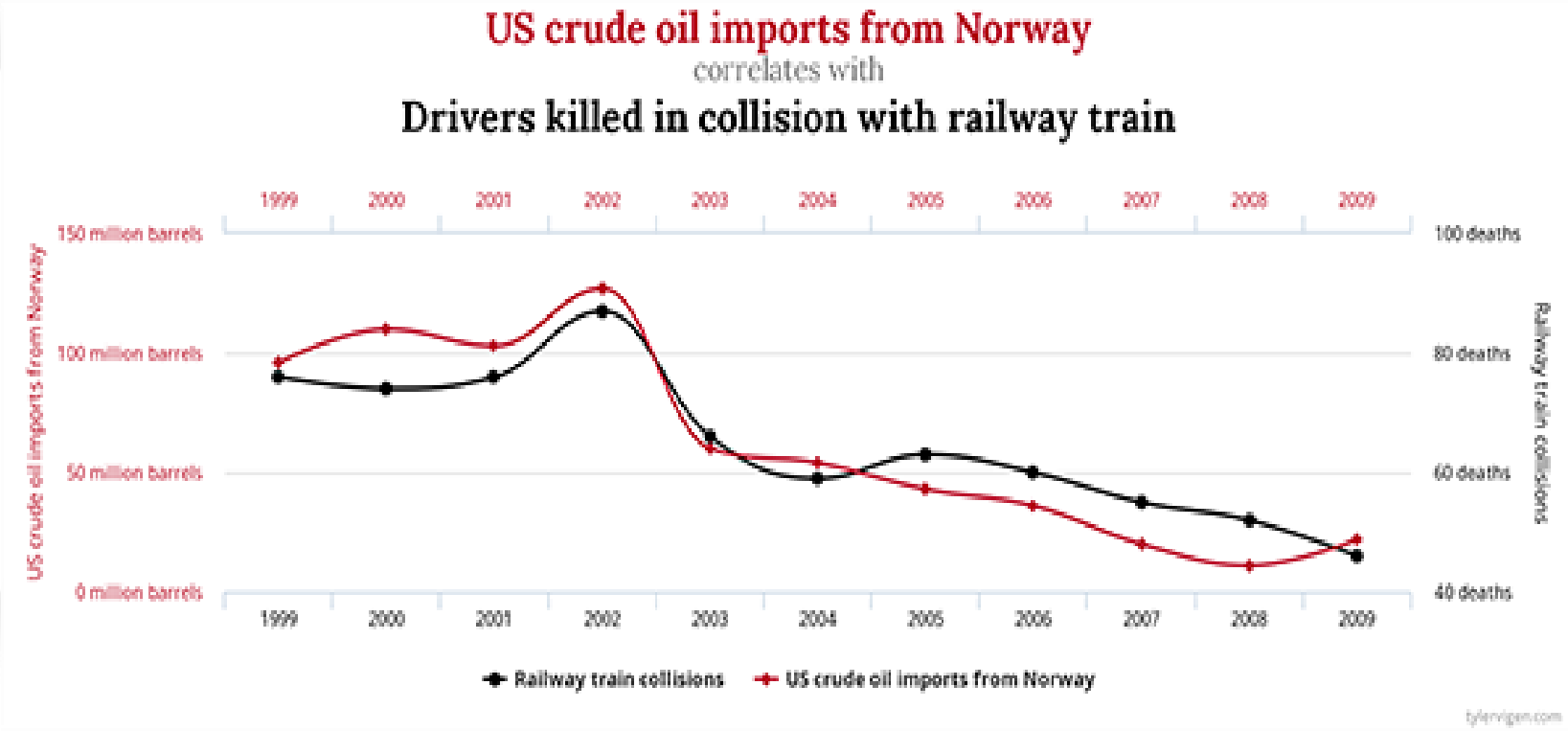
- The 3rd order (lag) partial autocorrelation is

$$\frac{\text{Covariance}(x_t, x_{t-3}|x_{t-1}, x_{t-2})}{\sqrt{\text{Variance}(x_t|x_{t-1}, x_{t-2})\text{Variance}(x_{t-3}|x_{t-1}, x_{t-2})}}$$

And, so on, for any lag.

Spurious correlations

- A spurious correlation occurs when two variables are correlated but don’t have a causal relationship. In other words, it appears like values of one variable cause changes in the other variable, but that’s not actually happening.



What Causes a Spurious Correlation?

Spurious correlation that produces a non-zero correlation coefficient and a graph that displays a relationship.

Confounding Variables

- Confounding occurs when a third variable causes changes in two other variables, creating a spurious correlation between the other two variables. For example, imagine that the following two positive causal relationships exist.
- A → B
- A → C

As A increases, both B and C will increase together. Hence, it appears that B → C.

- ## Mediating Variables

In other cases, a chain of correlations, or mediating variables, produces a spurious correlation. For example, imagine that both A & B and B & C have causal relationships, as shown below.

- A → B → C.

- ## Random Sampling Error

- Samples don’t always accurately reflect the population due to chance. Random sampling error can produce the appearance of effects in the sample that don’t exist in the population. A correlation is one possible effect .

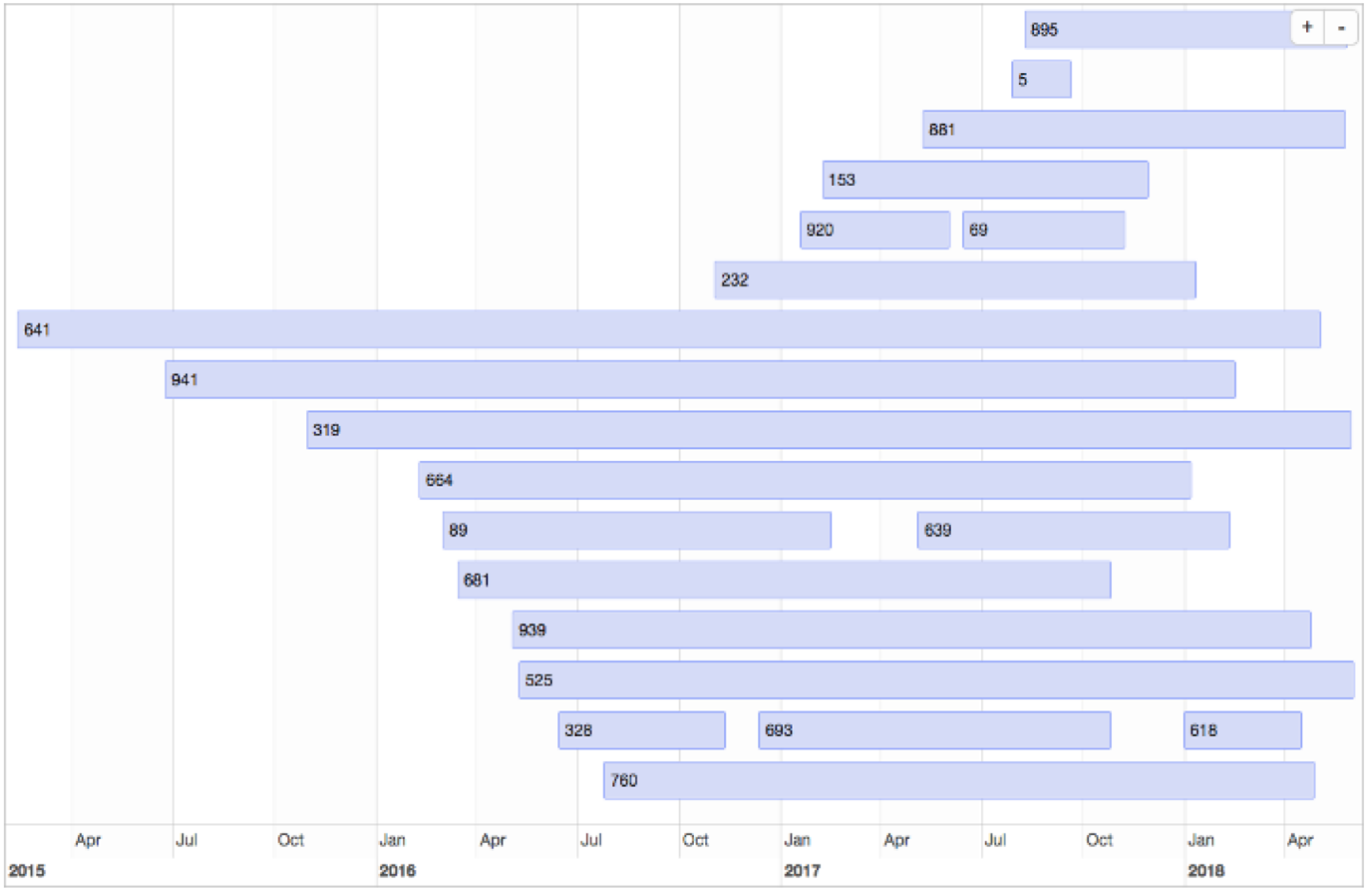
Visualizations

Visualizations of varying degrees of complexity:

- A one-dimensional visualization to understand the overall temporal distribution of individuals with a found time series
- A two-dimensional visualization to understand the typical trajectory of a value over time in the case of many parallel measurements
- A three-dimensional visualization where time can take up as many as two of the dimensions or as few as none of the dimensions, but still be implicitly present

1D Visualizations

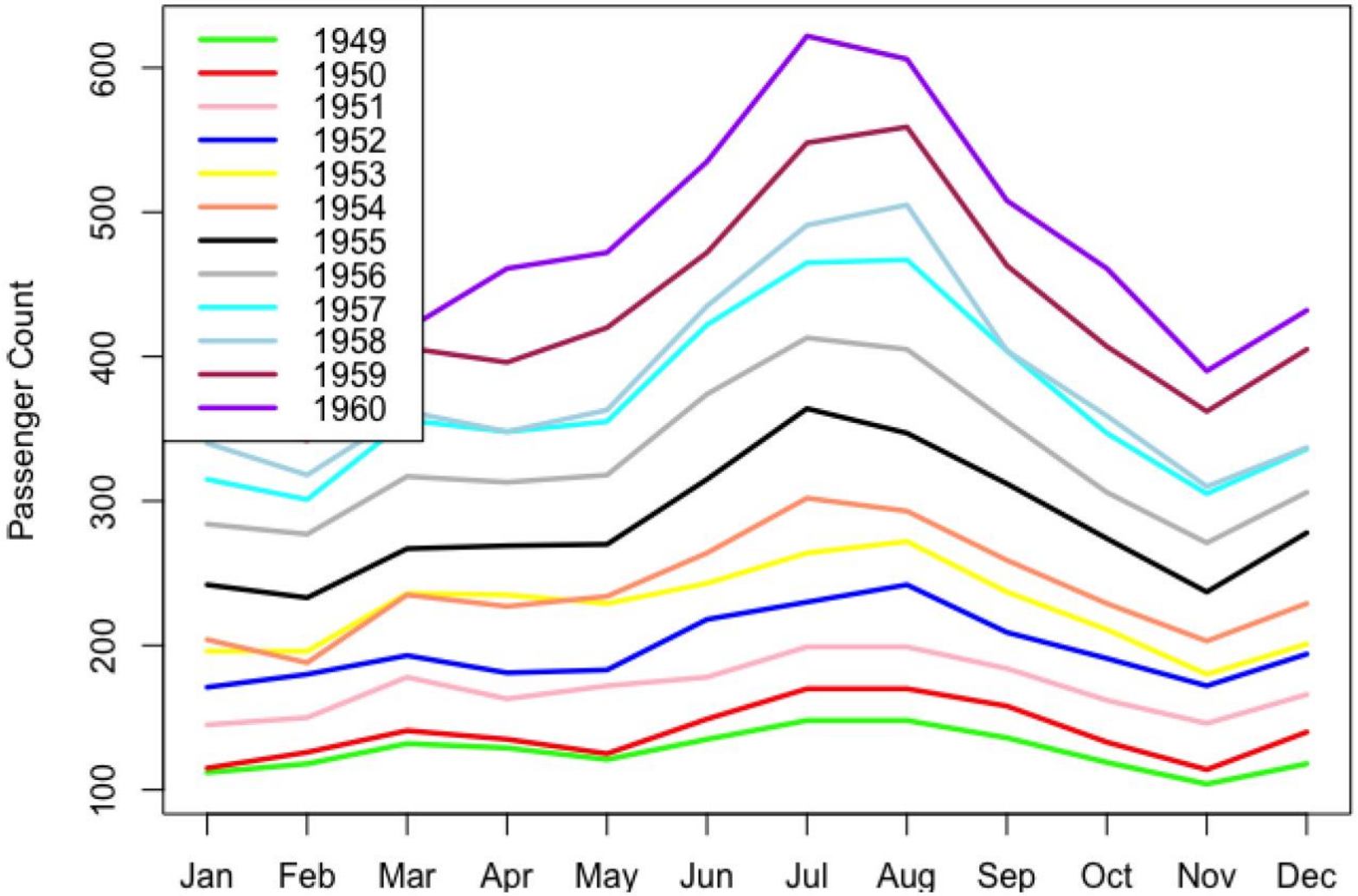
- In the cases of many units of measurement (many users, members, etc.) we consider multiple time series in parallel. It can be interesting to stack these visually, emphasizing individual units of analysis and their respective time frames.



A Gantt chart of a random sample of data can offer some idea of the distributionof the range of "active" time periods for the users/donors

2D Visualizations

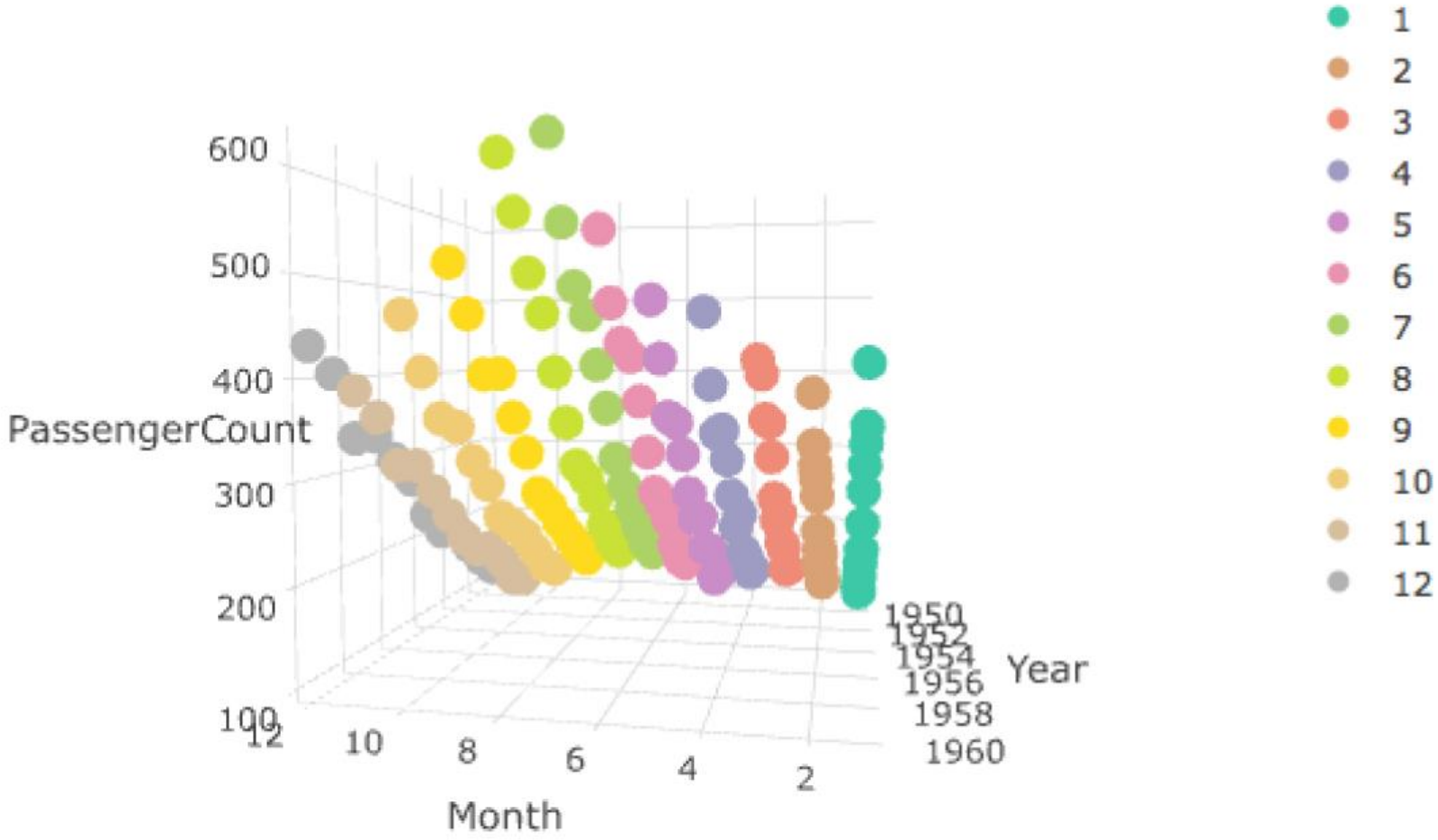
- Time happens on more than one axis. There is, of course, the axis of time going forward from day to day and year to year, but we can also consider laying time out along the axis of hour of the day or day of the week, and so on. In this way, we can more easily think about seasonality, such as certain behaviors happening at a certain time of the day or month of the year.



Per-year month-by-month counts

3D Visualizations

- 3D visualization helps us get a sense of the overall shape of the data expanding to a three-dimensional scatter plot proves to be notably better than a two-dimensional



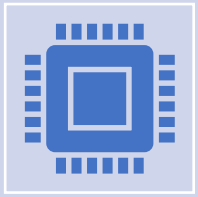
A 3D scatterplot of the AirPassenger data. This perspective highlights the seasonality.

Simulating Time Series Data

- Simulation of time series data is the process of creating artificial time series data that follows a particular pattern. This can be done for a variety of purposes, such as testing forecasting algorithms, evaluating the performance of statistical models, or generating synthetic data for training machine learning models.
- Different ways to simulate the time series data.
  - Random walk model
  - Statistical model
  - Machine learning model

Random walk model assumes that the next value in a time series is equal to the last observation plus a random noise term. This model can be used to simulate a variety of different time series patterns, such as trends, seasonality, and cyclicality.

Statistical models are more complex than random walk models, but they can be used to simulate more realistic time series patterns. For example, ARIMA models are a type of statistical model that can be used to simulate trends, seasonality, and cyclicality.



Machine learning models can be trained on real time series data to learn the patterns in the data. Once the model is trained, it can be used to generate synthetic data that follows the same patterns as the real data.

Simulating Time Series Data

Here are some of the benefits of simulating time series data:

- It can be used to test forecasting algorithms.
- It can be used to evaluate the performance of statistical models.
- It can be used to generate synthetic data for training machine learning models.
- It can be used to understand the underlying patterns in time series data.
- Simulations have lower stakes than forecasts; there are no lives and no resources on the line

Here are some of the challenges of simulating time series data:

- It can be difficult to generate realistic time series patterns.
- It can be time-consuming to simulate large amounts of time series data.
- It can be difficult to verify that the simulated data is accurate.

Storing temporal data:Defining requirements of Live vs Stored Data

- *How much time series data will you be storing?*
- *Do your measurements tend toward endless channels of updates (e.g., a constant stream of web traffic updates) or distinct events (e.g., an hourly air traffic time series for every major US holiday in the last 10 years)?* If
- *Will your data be regularly or irregularly spaced?*
- *Will you continuously collect data or is there a well-defined end to your project?*
- *What will you be doing with your time series? Do you need real-time visualizations? Preprocessed data for a neural network to iterate over thousands of times?*
- *How will you downsample data? How will you prevent infinite growth? What should be the lifecycle of an individual data point in a time series?*

Defining requirements of Live vs Stored Data

Here are some of the factors to consider when defining requirements for live vs stored data:

- **Latency:** Live data is typically more latency-sensitive than stored data. This is because live data is constantly being updated, and any delay in accessing the data can have a significant impact on the application. Stored data, on the other hand, can be accessed with a longer latency, as it is not constantly being updated.
- **Accuracy:** Live data is typically more accurate than stored data. This is because live data is collected from the source in real time, while stored data may be outdated or inaccurate. Stored data, on the other hand, can be more accurate for historical analysis, as it can be aggregated and filtered to remove noise.
- **Volume:** Live data is typically lower volume than stored data. This is because live data is only collected for a short period of time, while stored data can be collected for a long period of time. Stored data, on the other hand, can be higher volume, as it can contain data from a variety of sources and time periods.

- **Cost:** Live data is typically more expensive than stored data. This is because live data requires more infrastructure to collect and process, while stored data can be stored on less expensive storage. Stored data, on the other hand, can be more cost-effective for long-term storage, as it does not require the same level of infrastructure.
- **Security:** Live data is typically more sensitive than stored data. This is because live data may contain confidential information, such as financial data or customer data. Stored data, on the other hand, may be less sensitive, as it may not contain any confidential information.
- **Compliance:** Live data may need to comply with certain regulations, such as those governing financial data or healthcare data. Stored data, on the other hand, may not need to comply with the same regulations.
- **Scalability:** Live data may need to be scalable to handle large volumes of data. Stored data, on the other hand, may not need to be as scalable, as it is not constantly being updated.

Database vs file solution for storing time series data

Databases

- **Advantages:**
  - Efficient storage and retrieval of data. Databases are designed to store and retrieve data efficiently. They typically have a well-defined schema, which makes it easy to insert, update, and query data. Databases also offer a variety of features that are useful for time series data, such as indexing, aggregation, and time series analysis.
  - Well-defined schema. Databases have a well-defined schema, which makes it easy to insert, update, and query data. This is important for time series data, as it allows you to store the data in a structured way that makes it easy to analyze.
  - Indexes, aggregation, and time series analysis. Databases offer a variety of features that are useful for time series data, such as indexing, aggregation, and time series analysis. This makes it easy to find and analyze the data, which is important for applications that need to monitor and analyze time series data.

Databases

- Disadvantages:
  - More complex to set up and manage. Databases are more complex to set up and manage than files. This is because they require a more complex schema and they need to be configured to optimize for time series data.
  - Less flexible schema. Databases have a less flexible schema than files. This means that you need to define the schema before you start storing data, which can be limiting for some applications.
  - Not as efficient for large amounts of data. Databases are not as efficient for large amounts of data as files. This is because databases need to store the data in a structured way, which takes up more space.

Files

- Advantages:
  - Simple to use. Files are simple to use and can be stored on any filesystem. This makes them a good choice for applications that need to store time series data in a simple and flexible way.
  - Flexible schema. Files have a flexible schema, which means that you can store the data in any way that you want. This is useful for applications that need to store different types of time series data.
  - Efficient for large amounts of data. Files are efficient for large amounts of data, as they do not need to store the data in a structured way. This can save space and improve performance.

Files

- Disadvantages:
  - More difficult to query and analyze. Files are more difficult to query and analyze than databases. This is because the data is not stored in a structured way, which makes it more difficult to find and analyze the data.
  - Can be difficult to scale. Files can be difficult to scale, as they need to be stored on a filesystem. This can be a challenge for applications that need to store large amounts of time series data.

Popular Time series database solutions

- InfluxDB is an open-source time series database that is designed for high-performance, high-volume data ingestion. It is a good choice for applications that require real-time data ingestion and analysis.
- Prometheus is an open-source time series database that is designed for monitoring and alerting. It is a good choice for applications that need to collect and store metrics from a variety of sources.
- TimescaleDB is a time series database that is built on top of PostgreSQL. It is a good choice for applications that need to store historical data and perform complex queries.

File Solutions

- CSV files are a simple way to store time series data. They are easy to create and read, but they are not as efficient as time series databases.
- JSON files are a more flexible way to store time series data. They can be used to store both structured and unstructured data, but they are not as efficient as time series databases
- Parquet files are a binary format that is designed for efficient storage and querying of time series data. They are a good choice for applications that need to store large amounts of time series data and need to perform complex queries