Decomposition of Time Series Data

Nicholas Bradley

# Decomposition of Time Series Data

Here we will focus on the essential elements of time series analysis - the decomposition process of time series data to its components: the trend, seasonal and random components. We will examine the moving average function and see its applications for smoothing time series data, removing seasonality and estimating a series trend.

## The moving average function

The moving average (MA) is a function for smoothing time series data, which is based on averaging each observation of a series, when applicable, with its surrounding observations, that is, with a past, future or combination of both past and future observations in chronological order. The output of this transformation process is a smoothed version of the original series. The MA function has a variety of applications, such as data smoothing, noise reduction, and trend estimation. The main components of the MA are as follows:

* **The rolling window:** a generic function that slides across the series in a chronological order to extract sequential subsets.
* **Average function:** This is either a simple or weighted average, which is applied on each subset of the rolling window function

## The rolling window structure

The rolling window structure defines the sub-setting method of series observations and the most common types are:

* **The one-sided window:** This is a sliding window with a width of n, which groups each observation of the series, with its past consecutive n-1 observations.
* **The two-sided window:** This is a rolling window with a width of n, which groups each observation of the series.

## The average method

The second part of the MA is the averaging method of the window’s subset and there are two types:

* **The arithmetic average:** This is when the most common and basic method for averaging a sequence of observations. It sums all the observations and divides them by the number of observations.
* **The weighted average:** This appliess weight to each observation of the series.

## The MA attributes

Additionally, the MA has two primary attributes that are derived directly from the window structure.

* **Order:** The order defines the magnitude of the MA and is equal to the length of the window. Where a one-sided window is used, the width of the window is defined by n, as we are using the current observation, with the past n-1 consecutive observation. There is also a cost in using the MA function, as it causes a loss of observations during the transformation of the origin series to the smoothed series by the MA process.
* **Cost:** Once the parameters (window structure and average type) are set, the function starts to roll the window over the series observations in chronological order to calculate the series, MA’s. The observations of the new series represent the corresponding MAs of the original series in chronological order. Where the new series inherits the main characteristics of the origin series, such as the frequency and timestamp, the main applications of the MA function are:
* **Noise reduction:** This is when the use of MA method creates a smoothing effect that reduces the series variation, smoothing the random noise and outliers.
* **De-seasonalise:** MA’s can also be used to remove the seasonal component (if any).
* **Forecast:** With a few minor modifications, the MA can be used to forecast future series observations bey averaging the past observations to estimate future values.

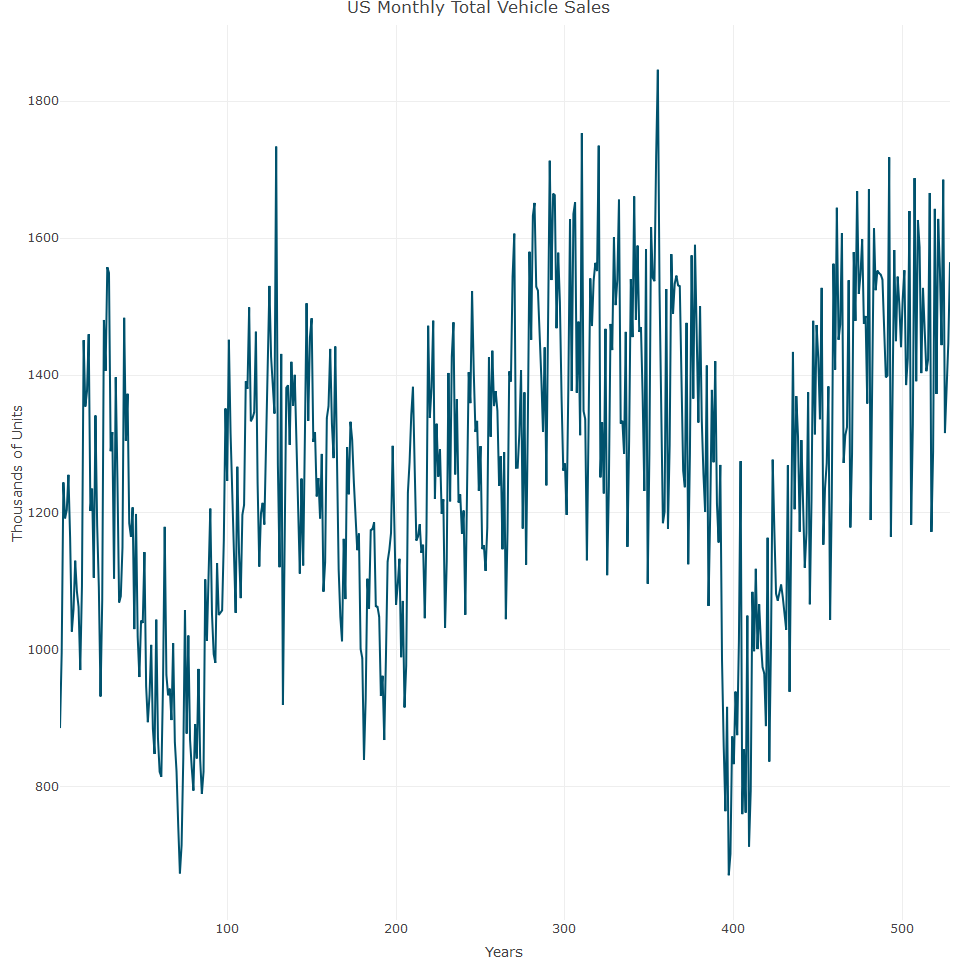
We can use the US Monthly Vehicle Sale data to demonstrate a one-sided MA, a two-sided MA and a weighted MA

## Time-Series [1:528] from 1 to 528: 885 995 1244 1191 1203 ...

Let’s see the characeristics of this time series

## The US\_Sales\_ts series is a ts object with 1 variable and 528 observations  
## Frequency: 1   
## Start time: 1 1   
## End time: 528 1

Let’s now plot it



Let’s now look at the simple moving average

## The Simple Moving Average

This transformation is a common MA function and is based on applying an arithmetic average on a one-sided rolling window. Hence, the rolling window groups each observation in the series(when applicable) with its previous n consecutive observations in order to calculate their arithematic average.We need to create the rolling window and the average function in order for the simple moving average to work. The function below requires a ts object and the number of lags to construct, It then utilises the ts.union and lag functions to create 1 lag and then unites them with the original series. The output of this function is a multiple time series object

## Time Series:  
## Start = 7   
## End = 12   
## Frequency = 1   
## y y\_1 y\_2 y\_3  
## 7 1162.3 1254.7 1203.2 1191.2  
## 8 1026.1 1162.3 1254.7 1203.2  
## 9 1057.9 1026.1 1162.3 1254.7  
## 10 1129.4 1057.9 1026.1 1162.3  
## 11 1084.4 1129.4 1057.9 1026.1  
## 12 1061.8 1084.4 1129.4 1057.9

## The lags(US\_Sales\_ts, l = 3) series is a mts object with 4 variables and 522 observations  
## Frequency: 1   
## Start time: 7 1   
## End time: 528 1

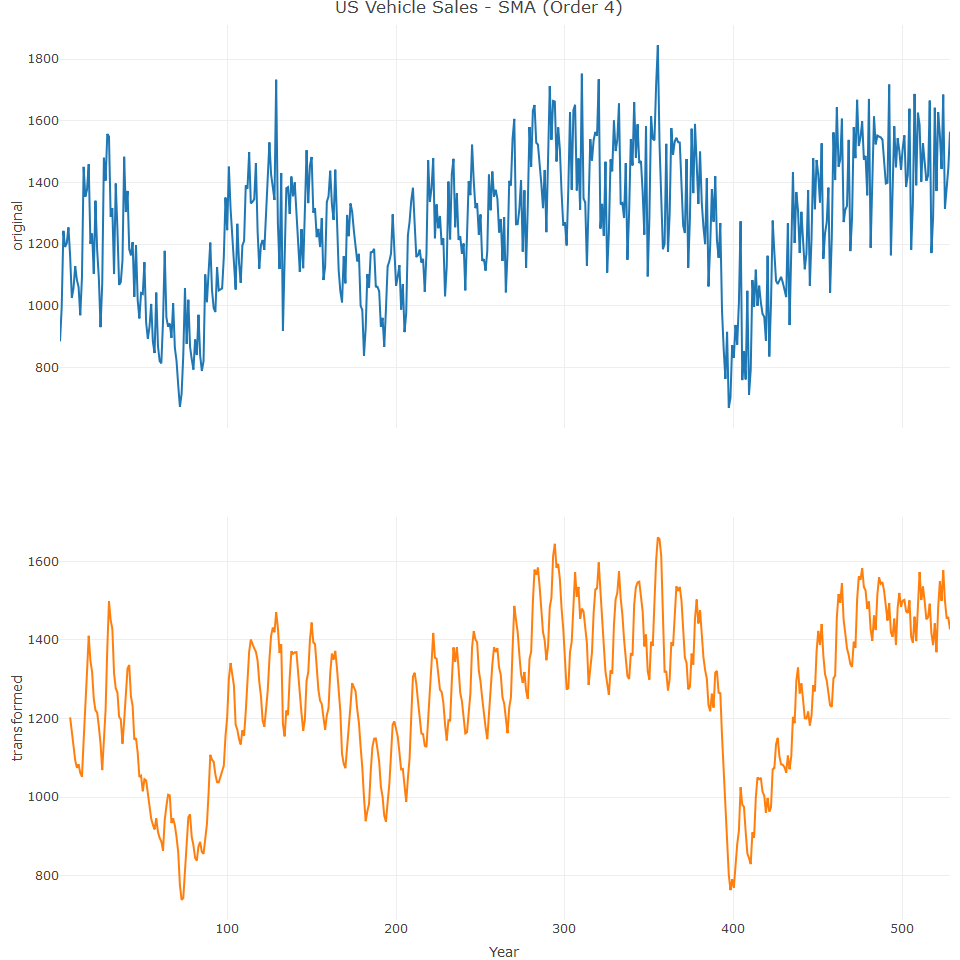
The output of the lags function illustrates the cost associated with the MA method. Here, we lost three observations as a result of creating three lags for the series. The lag function automatically drops the first observation (where l is the lag setting argument in the preceeding lags function) of the original series. Therefore, the first observation in the data is April 1974 and not January 1974 as in the original series.

The second and final step of this process is to calculate the artimetic average of the series with its n lags. We utilise the ts\_sum function, which sums the rows of the mts object and returns a ts object.

We now finalise the sma function by linking the lags and average functions. Note that the input parameters of the sma function are the series and order of the SMA

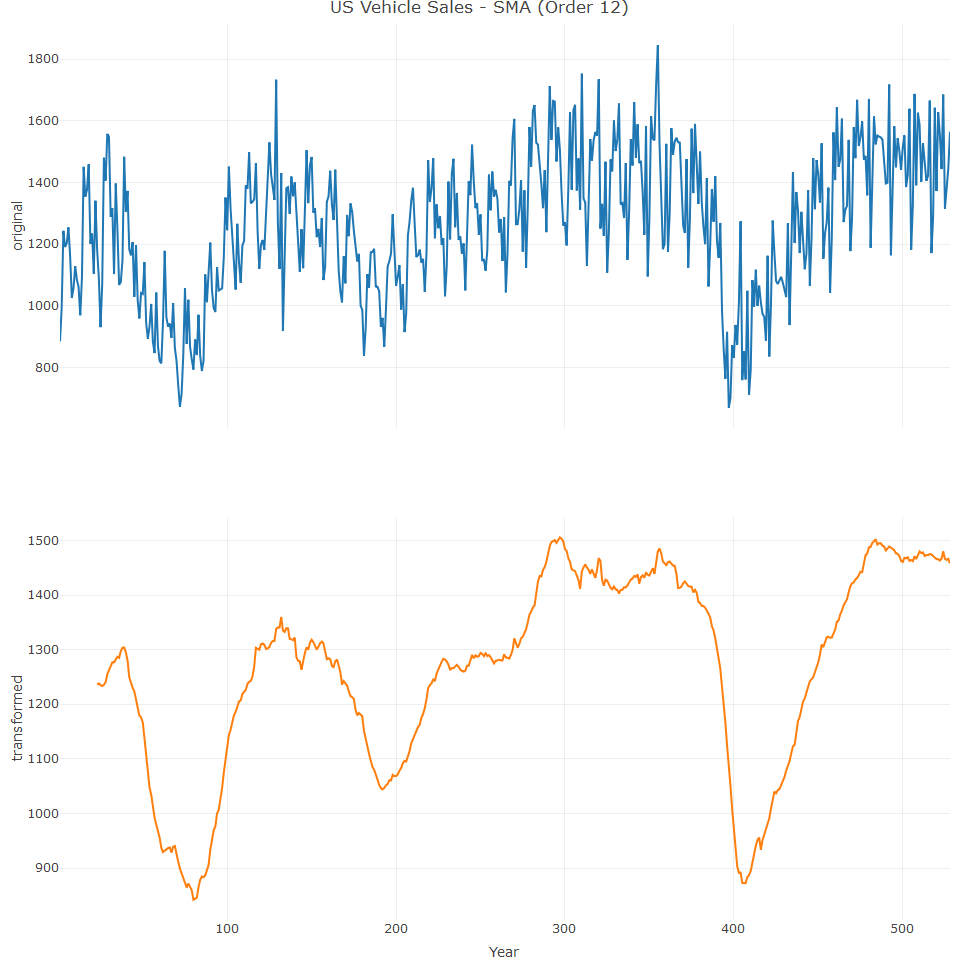
We can now use this function to smooth the US vehicl sales series. We set the order of the sma function to 4, hence, we will calculate the average price of each month with the three past consecutive months

## Time Series:  
## Start = 1   
## End = 10   
## Frequency = 1   
## original transformed  
## 1 885.2 NA  
## 2 994.7 NA  
## 3 1243.6 NA  
## 4 1191.2 NA  
## 5 1203.2 NA  
## 6 1254.7 NA  
## 7 1162.3 1202.850  
## 8 1026.1 1161.575  
## 9 1057.9 1125.250  
## 10 1129.4 1093.925



As can be seen in the proceeding graph is the noise reduction. By using a relatively lower order (with respect to the frequency of the series), SMA reduces some of the oscillation in the series that is related to random noise. The remaining oscillation of the series is mainly related to the seasonality pattern of the series and it’s trend and cycle of the series. Let’s increase the order to reduce oscillation even further, by for example increasing it to 12:

## Time Series:  
## Start = 1   
## End = 30   
## Frequency = 1   
## original transformed  
## 1 885.2 NA  
## 2 994.7 NA  
## 3 1243.6 NA  
## 4 1191.2 NA  
## 5 1203.2 NA  
## 6 1254.7 NA  
## 7 1162.3 NA  
## 8 1026.1 NA  
## 9 1057.9 NA  
## 10 1129.4 NA  
## 11 1084.4 NA  
## 12 1061.8 NA  
## 13 969.9 NA  
## 14 1092.1 NA  
## 15 1451.1 NA  
## 16 1354.4 NA  
## 17 1377.1 NA  
## 18 1459.8 NA  
## 19 1202.0 NA  
## 20 1234.5 NA  
## 21 1104.6 NA  
## 22 1341.3 NA  
## 23 1181.6 1235.850  
## 24 1090.6 1238.250  
## 25 931.3 1235.033  
## 26 1071.4 1233.308  
## 27 1480.6 1235.767  
## 28 1406.5 1240.108  
## 29 1557.6 1255.150  
## 30 1549.4 1262.617



The change in the order of the SMA function from four to twelve reduces the remaining oscillation of the series, which is mainly related to the seasonal patterns of the data. This use of MA to estimate the trend and seasonal pattern is part of the decomposition of a time series.

## Two Sided MA

The two-sided MA is based on a two-sided rolling window function. The term two-sided refers to the use of a two-sided window function with an arithmetic average unless mentioned otherwise. The output of the two-sided MA function could be either of the following:

**Centered:** This is when and are equal, which means that the function output at time *t* is centered around the *t* observation of the original *y* series. **Uncentered:** This is when the length of is different from the length of , or the MA order is an even number

We can use the US vehicle sales data to examine the effect of a different order two sided MA using the ts\_ma function, which enables a user to generate and plot multiple MA outputs at the same time, whilst using different methods and orders (SMA, two-sided MA etc). The function’s main parameters are:

**n:** sets the length of the past and future observations to be used in a two-sided MA function. If *n* is set to three, the window function will group each observation with its past and future three consecutive observations, which will yield a 7 order MA. It is possible to set this parameter with multiple values to generate multiple two-sided MA functions concurrently using a different order. **n-left/n-right:** these are used to customise the MA function by setting the length of the past (n-left) manually and/or the future (n\_right) side of the window function. If both parameters are defined the output is a two-sided MA function, either centered or uncentered. If only one parameter is set, the function output is a one-sided MA.

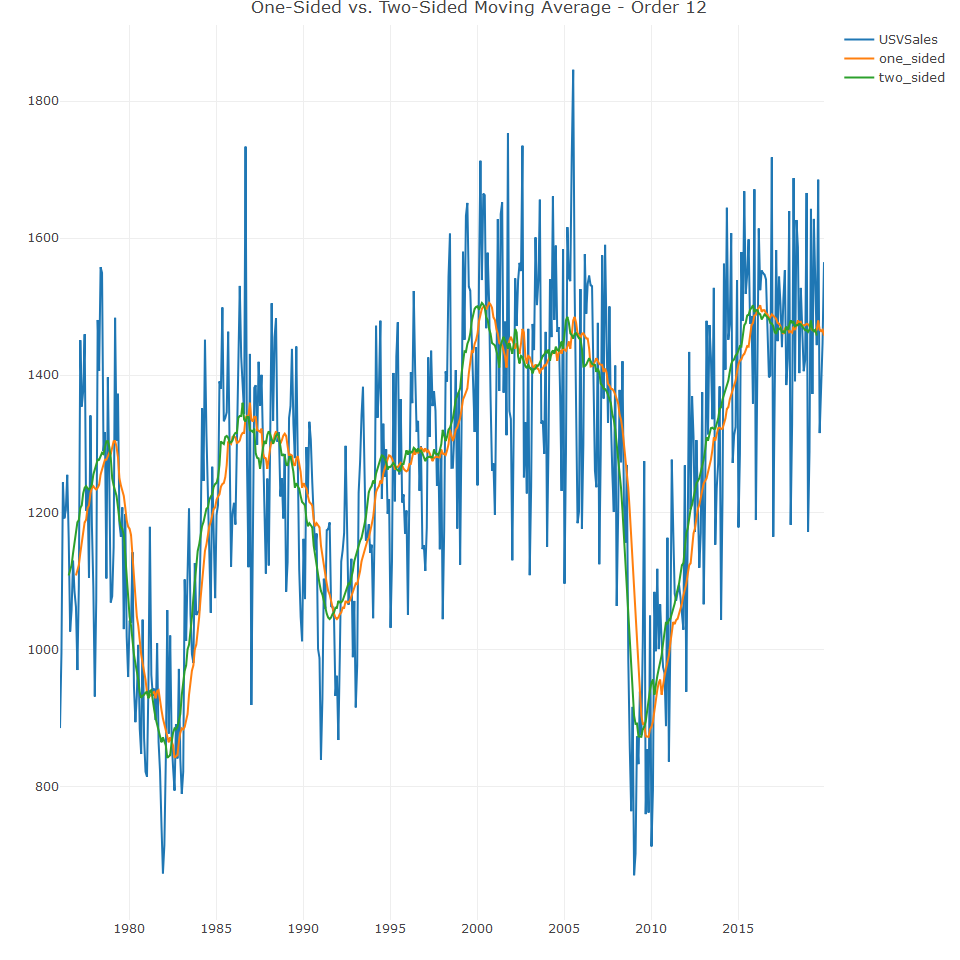
The following code generates three two-sided MA outputs with an order of 5, 11, 12 using the ts\_ma function. The first two outputs with an order of 5 and 11, are defined by setting the *n* parameter to 2 and 5. This yields a centered output, as the window function is symmetric.

As can be seen, the smoother the output, the higher the loss of observations - just compare the last plot with the first plot. The fifth order function performed well on smoothing some of the series oscillation, but left the seasonal and trend patterns. While both the eleventh and twelfth order functions removed the seasonal pattern, some of the noise that was left in the 11th one was smoothed by the 12th one. The selection of the order of the functions depends on the level of smoothing required and the cost that is acceptable

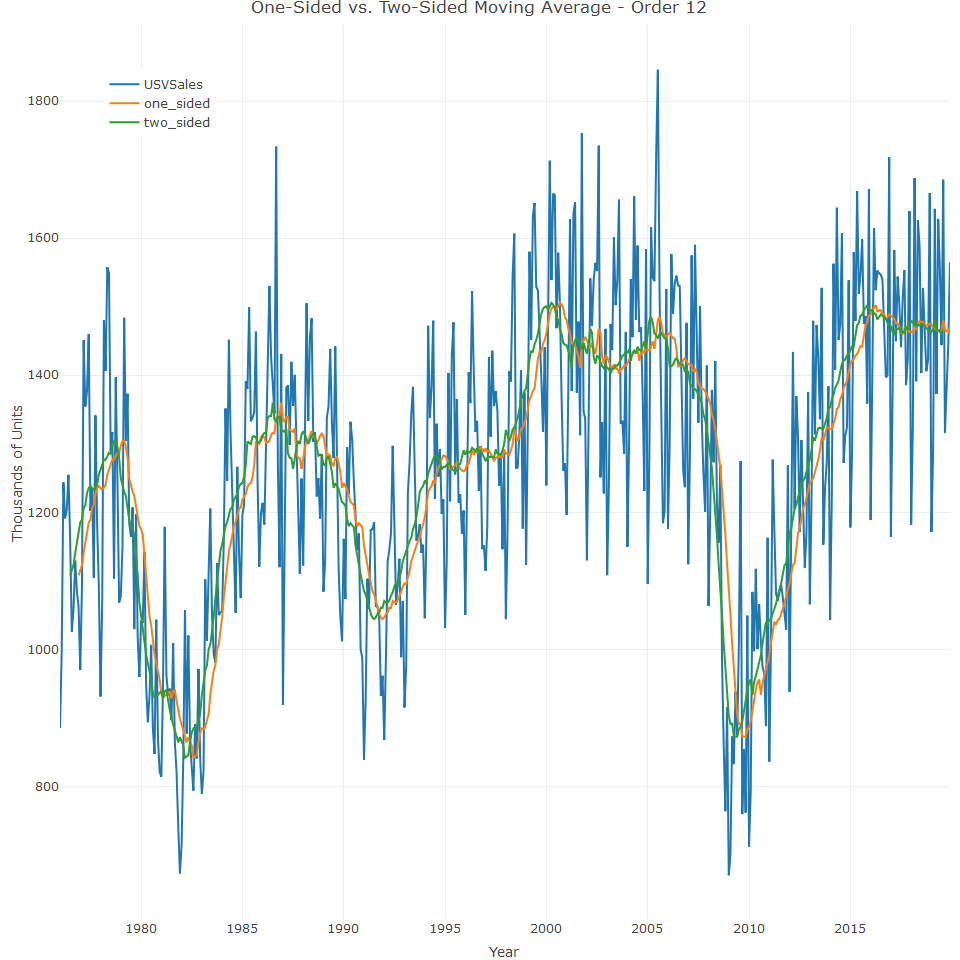
## Comparison between a simple MA and a two-sided MA

Generally, when applying both simple and two-sided MA’s with arithmetic average using the same order, as demonstrated in the following example, the output of both methods will be identical but lagged. The two-sided window at time *t* will group observations until inclusively, which is the same group of observations when using the one-sided window at time (assuming both windows have the same order).

We can now bind the output of the one-sided and two-sided MA functions with the USVSales series and plot it with the ts\_plot function:



We can also use the layout function from plotly to set the plot legend and labels.



In the above plot, we can see that the output of the two-sided MA fits better with the overall changes in the series trend with respect to the one-sided output, which is mainly caused by a delay in five periods between the two methods. This does however come at the cost of losing the first series’ last 5 observations in addition to the first six, as opposed to the last of the first 11 observations in the one sided MA. The loss of observations by both functions is 11, but generally, the value of the last observations is higher than the first observations in the series. This makes the use of the two-sided MA more expensive if one cares about the most recent observations of the series.

## Time Series Components

One of the primary goals of time series is to identify patterns in the data, which can then be utilized to provide meaningful insights about past and future events such as seasonal, outliers or unique events. Patterns can be categorised into one of the following:

**Structural patterns:** These are known as series components, representing the core structure of the series. There are three types, namely trend, cycle and seasonal.

**Non-Structural patterns:** This refers to any other patterns in the data that are not related to structural patterns.

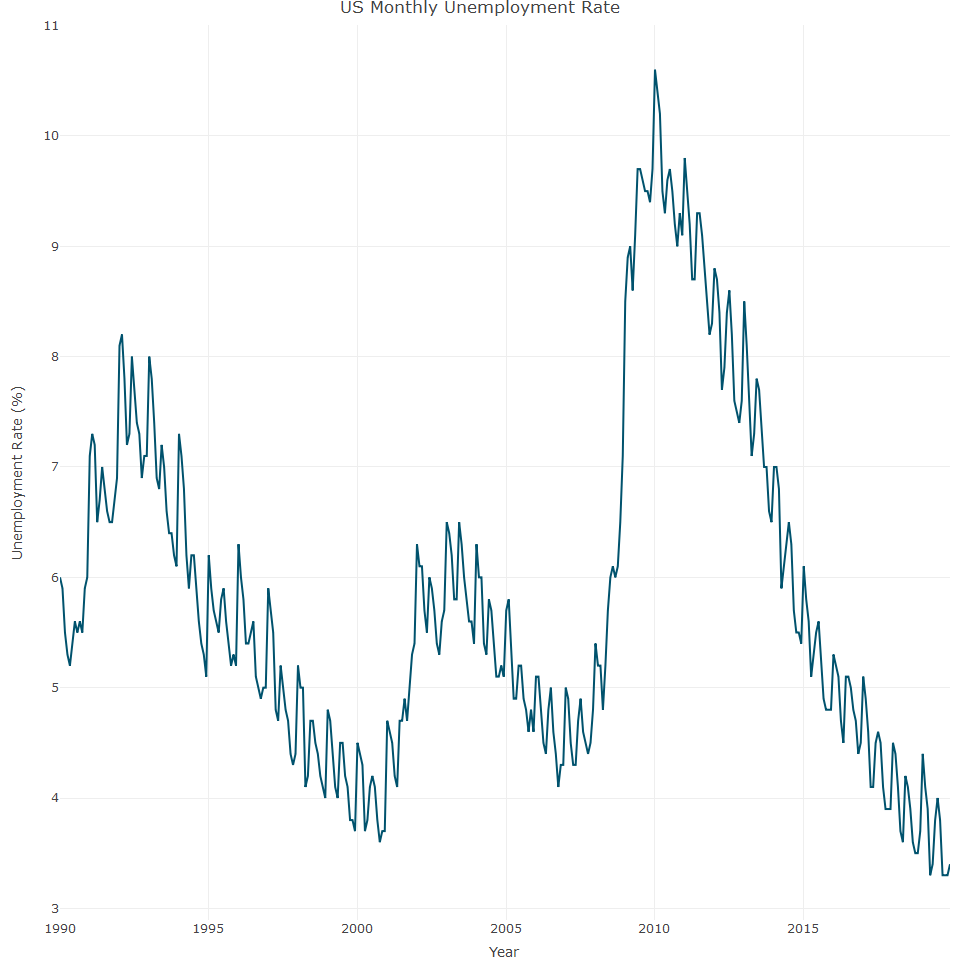
## The Cycle Component

A cycle can be defined as a sequence of repeatable events over time, where the starting point of a a cycle is at a local minimum of the series and the ending point is at the next one, and the ending point of one cycle is the starting point of the following cycle.

Unlike the seasonal pattern, cycles do not necessarily occur at equally spaced time intervals and their length changes from cycle to cycle. The US monthly unemployment time series is an example of a series with a cycle pattern. Let’s examine it’s main characteristics.

## The USUnRate series is a ts object with 1 variable and 864 observations  
## Frequency: 12   
## Start time: 1948 1   
## End time: 2019 12

Thi time series has a monthly frequency and begins in January 1948 and we don’t need the full series, wo we use the window function



The above plot indicates that there have been three cycles since 1990:

* The first occurred between 1990 and 2000, close to an 11 year cycle.
* The second started 2000 and ended in 2007
* The third cycle began in 2007 and had not ended

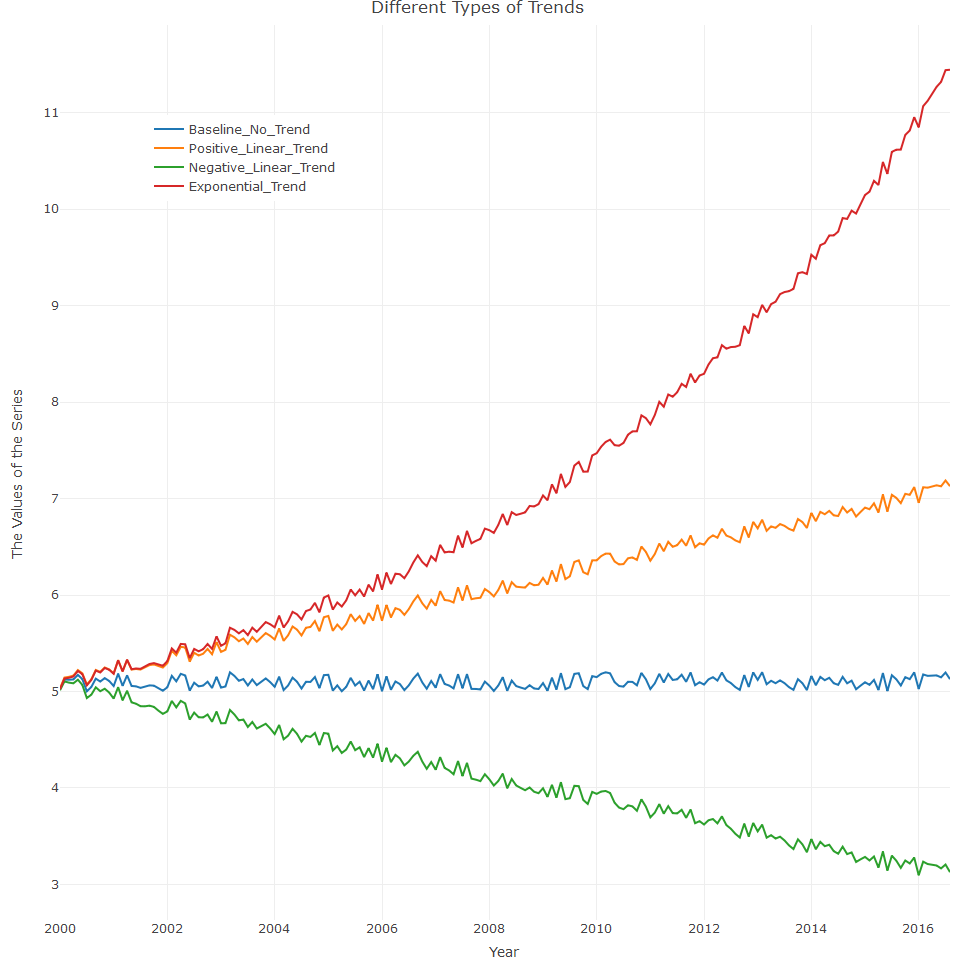
## The trend component

A trend represents the general direction of the series, either up or down, over time. Let’s generate some none trend series as our baseline data. We use the runif function to create a monthly series with 2000 minimum

The code above can be explained as follows:

* ts\_linear\_trend\_p: is a series with a positive linear trend, which adds increasing arithmetic progression sequence as a function of time
* ts\_linear\_trend\_n: is a series with a negative linear trends, which adds decreasing arithmetic progression sequence as a function of time
* ts\_exp\_trend: is a series with an exponential trend, which adds an increasing geometric progression sequence as a function of time

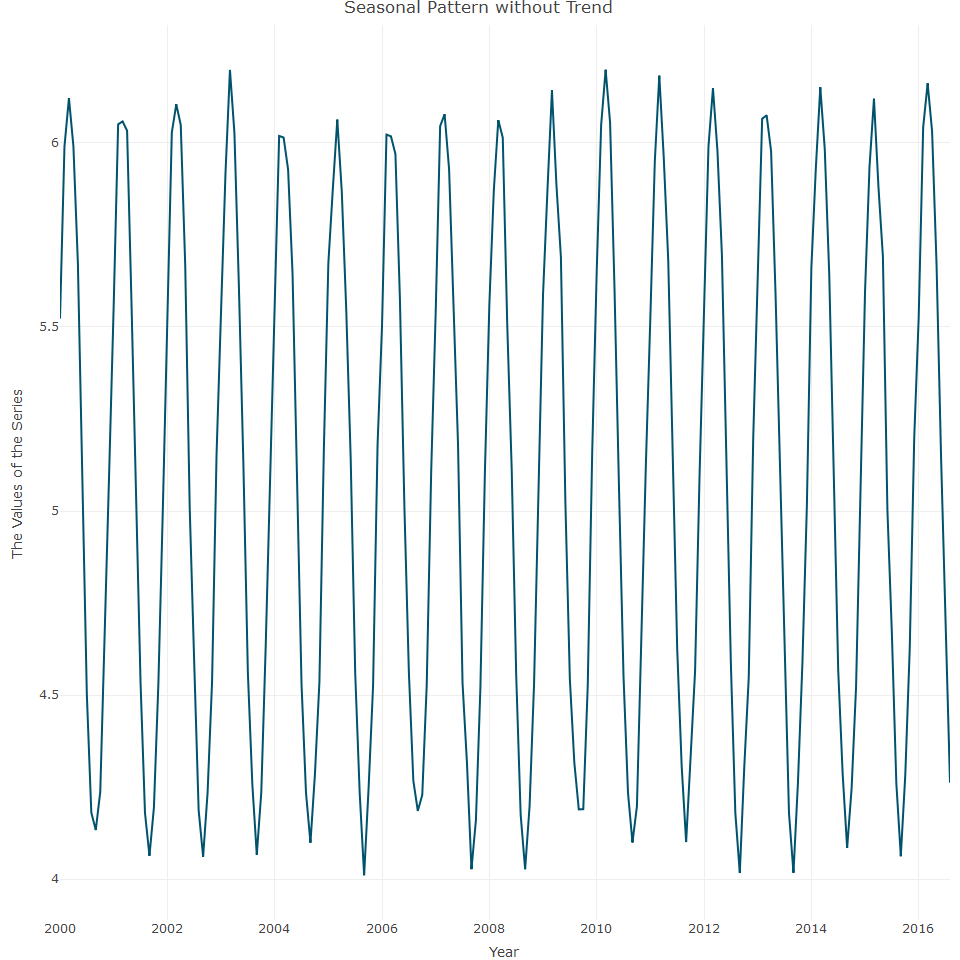
Let’s plot the series and examine the different trends and we first need to merge the four series



## The Seasonal Component

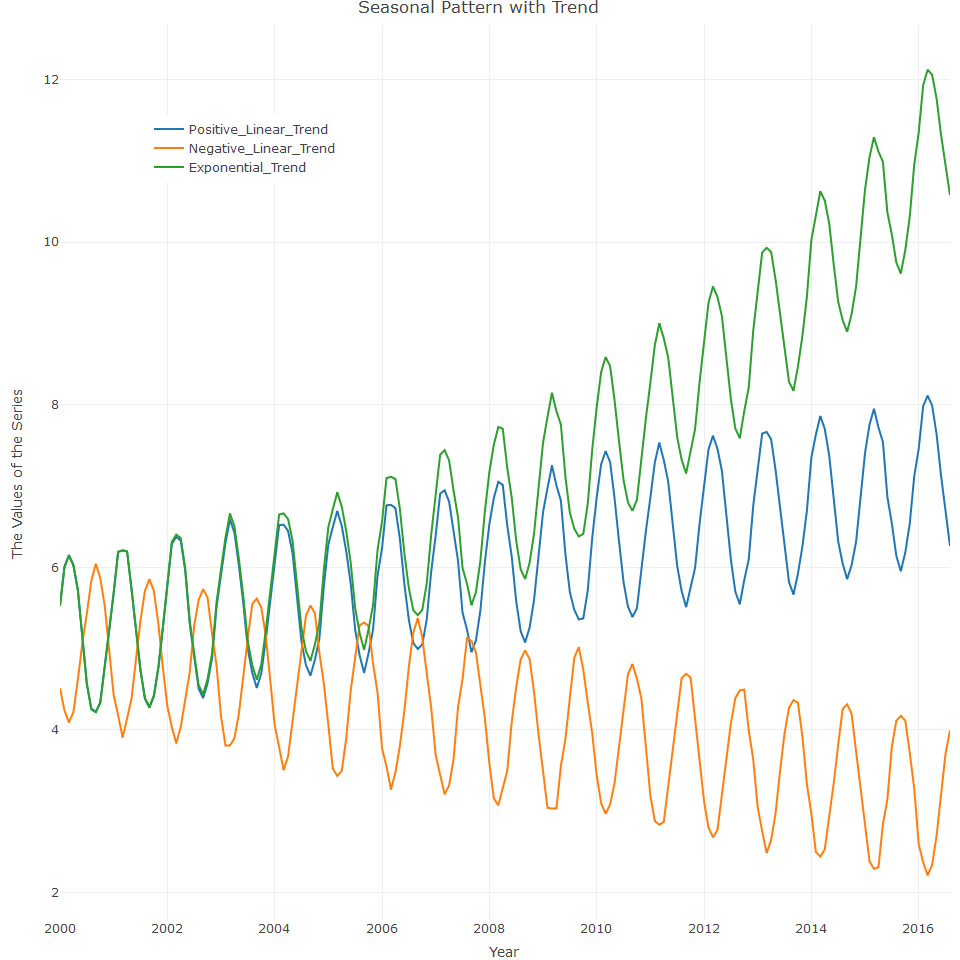
This component is another common pattern in time series data, but it is not found in every time series.It represents a repeated variation in the series, which is related to the frequency units of the series (for example, the months of the year for a monthly series). A series with common seasonality is that of gas or electricity with its patterns of demand based on weather, season of the year and sunlight hours. It is also possible for a series to have several seasonal patterns - such as the hourly demand for electricity, which could have three seasonal patterns:

-Hourly seasonality, which is derived from parameters such as sunlight hours and temperatures throughout the day -Weekly seasonality, depends on the day of the week (weekday versus weekend) -Monthly seasonality, related to season of the year (high winter consumption versus low summer consumption.



The above plot is very simplistic example of a series with a seasonal pattern, so in reality, it is most likely that series will combine multiple patterns which create a more complex structure. A series with both seasonal and trend patterns is a good example of a mixture of patterns. We can now add the patterns to the rest of the trend series we created before (both linear and exponential) and merge it into a single series.

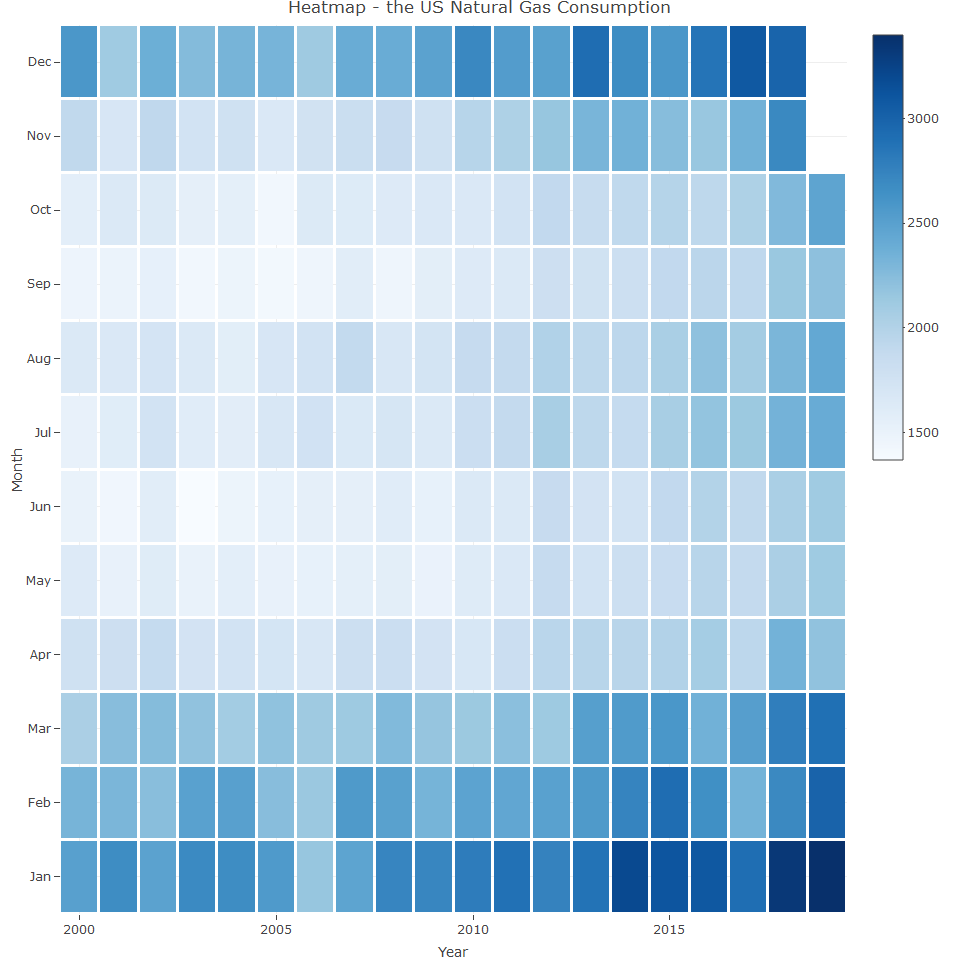
We can now create the plot



Let’s now examine the difference between the seasonal component and the cycle component

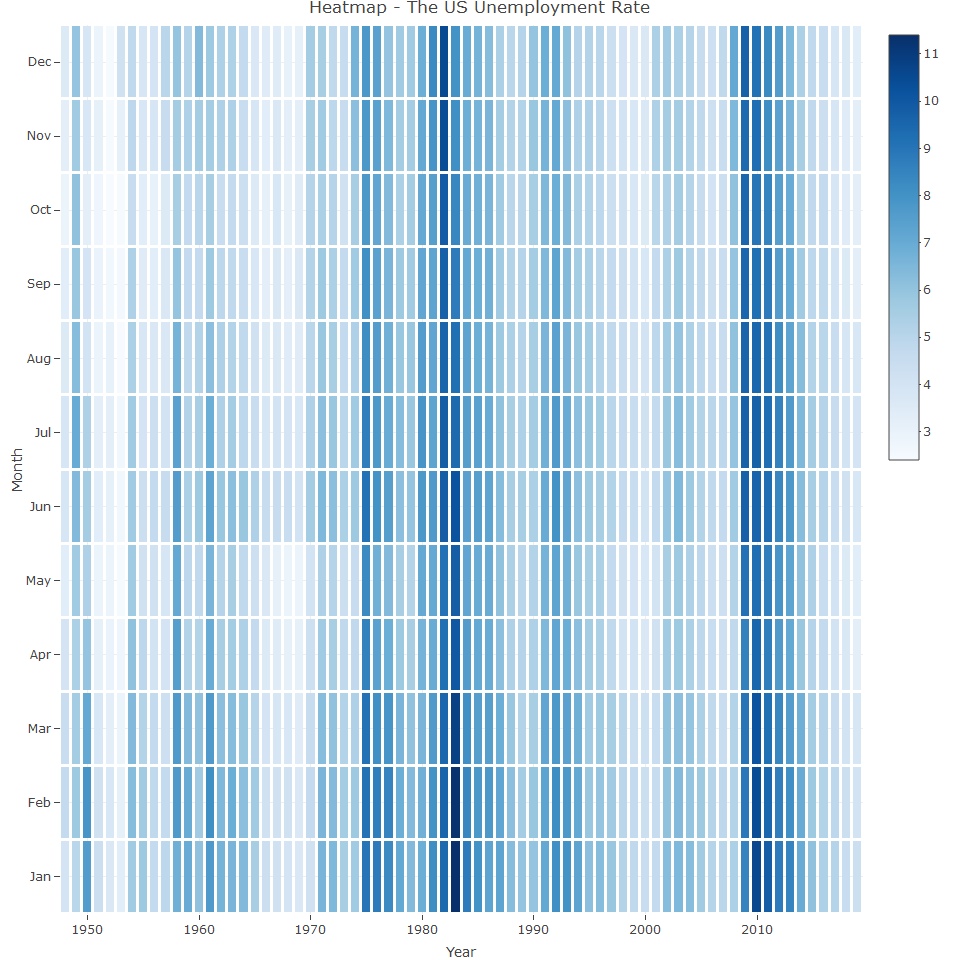
## The seasonal component versus the cycle component

Seasonal and cycle components both describe cyclic events over time where the length of their cycle distinguish the two. The cycle component has a constant cycle which is derived from and is tied to the series frequency, while the cycle length of the cycle component is no always constant and can vary from one cycle to the next. A heatmap is a simplistic way to identify whether a cycle pattern exists in a series



As can be seen, the the color flow derives from the frequency units. The winter months of December and January usually have the darkest colour and May, June, September have the brightest colours.

We can examine another heatmap:

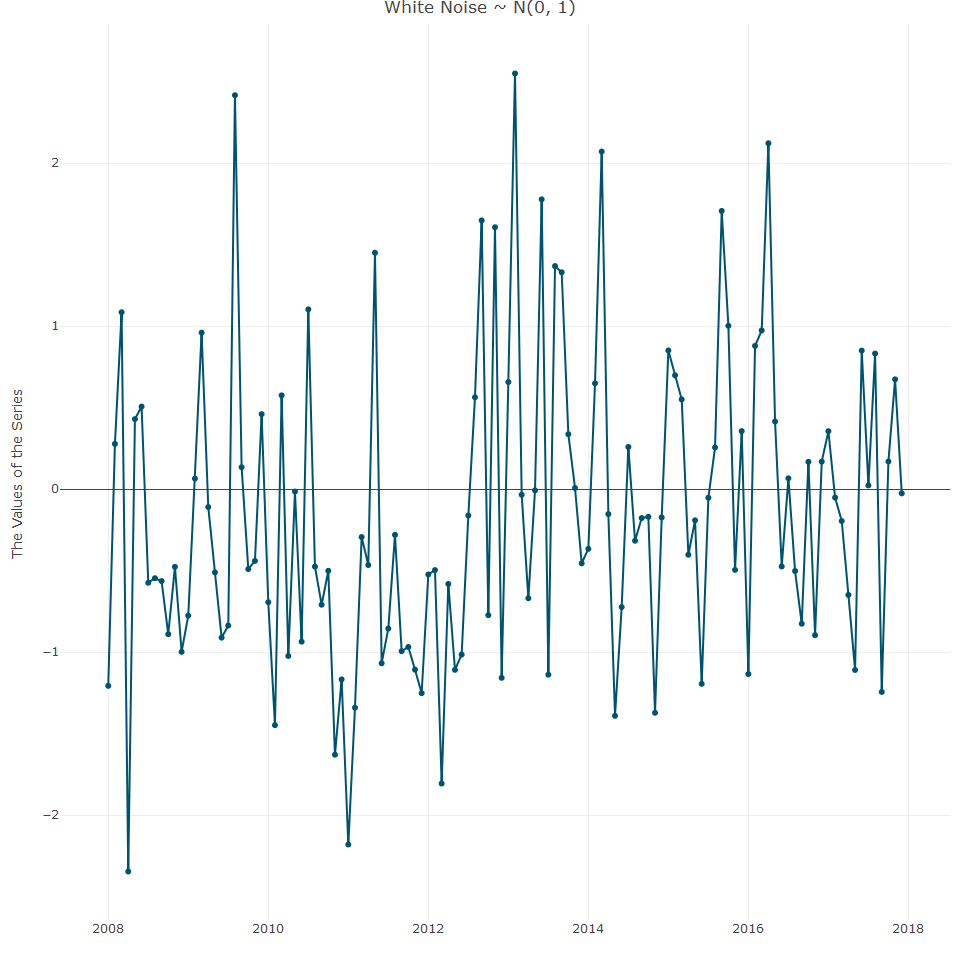


In the plot above, the colour flow of USUnRate is vertical, indicating the state of he cycle. The brightest vertical strips represent the ending of one cycle and the beginning of the new one. The darkest strips represent the cycle peaks.

## White Noise

Paradoxically, the main pattern of a white noise series is the lake of patterns. A series is defined as white noise when there is no correlation between the series observations or patterns. In other words, the relationship between the different observations is random. In many of the applications of white noise in time series, there are many assumptions made about the distribution of the white noise series. Typically, unless mentioned otherwise, we assume that white noise is an independent and typically distributed random variable with a mean of 0 and a variance of . For instance, we can simulate white noise with the rnorm function and generate random numbers with a normal distributionof mean 0 and variance of 1:

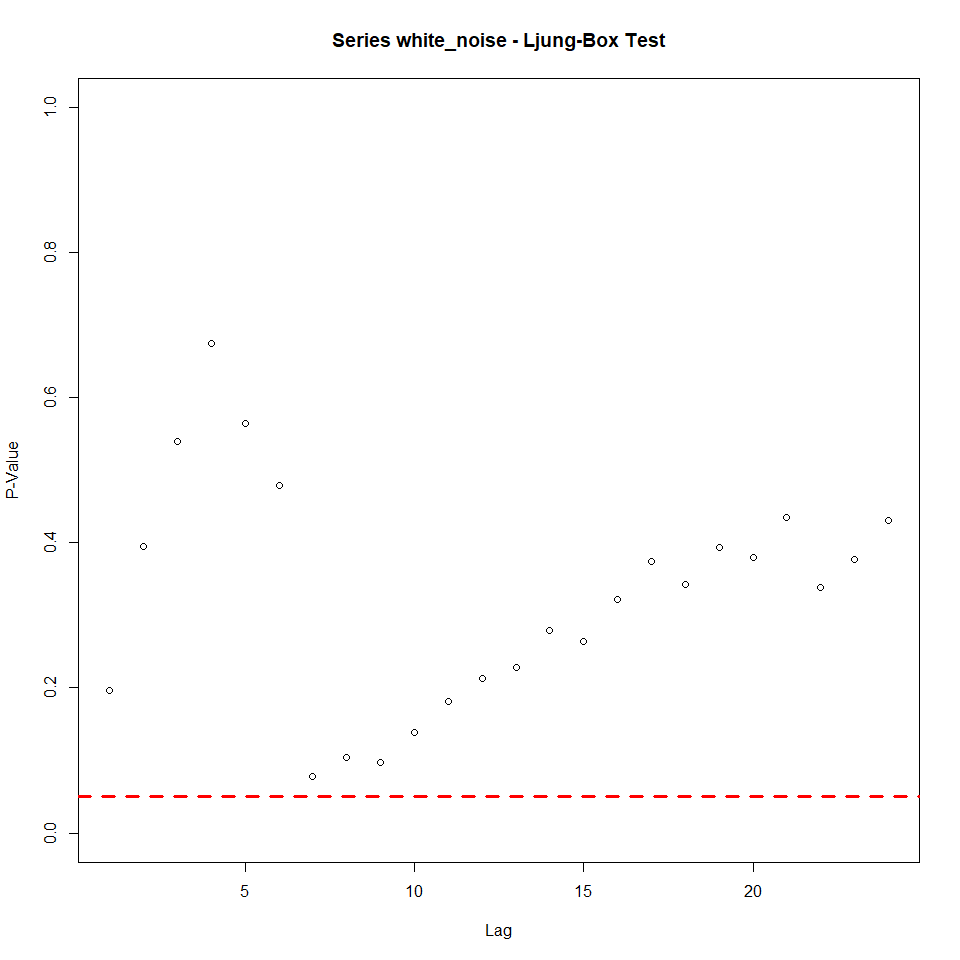
We can now plot the graph:



There are a few methods for testing whether a time series is white noise:

* The basic method is carried out by plotting and examining the series to identify whether the variation of the series is random or not.
* It is possible to measure the correlation between the series and its lags with the autocorrelation function (ACF). A series is considered to be white noise whenever there is no correlation between the series and its lag. Th acf from the stats package calculates the level of correlation between a series and its lags
* The Ljung-Box test evaluates whether the series is correlated with its lags. In this case, the null hypothesis assumes that the lags are not correlated, so therefore, lags with lower p-values would be considered as correlated with the series. The Box.test function is used to perform the Ljung-Box test.

We apply the Box.test function to apply the Ljung-Box test to the white noise series and it’s first 24 lags and then plot the results and add a horizon to define a signifance level of 0.05



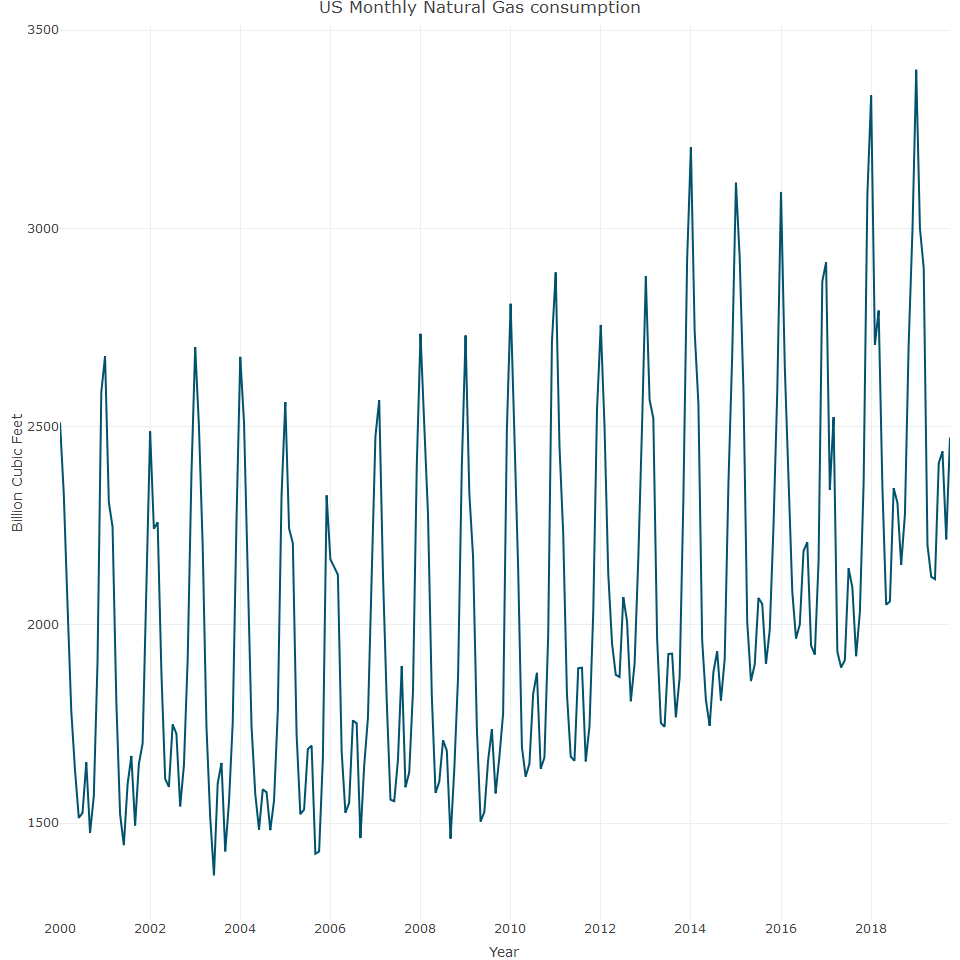
We can see that all lags are above the red line, which indicates that we failed to reject the null hypothesis for a significance of 0.05 and therefore, the series is not correlated with its first 24 lags and is thus a white noise series.

## The irregular component

This is the fourth component, which is the remainder between the series and structural components, providing an indication of irregular events in the series, including non-systematic patterns or events in the data, which cause irregular fluctuation. This irregular component could also provide indication of the appropriate fit of the other components when using a decomposing method. A high correlation in this component is an indication that some patterns are related to one of the other components were left over due to an inaccurate estimate. If the irregular component is not correlated with its lags, we can assume (depending on series structure) that estimation of the trend and seasonal components captured the majority of the information about the series structure.

## The additive versus the muliplicative model

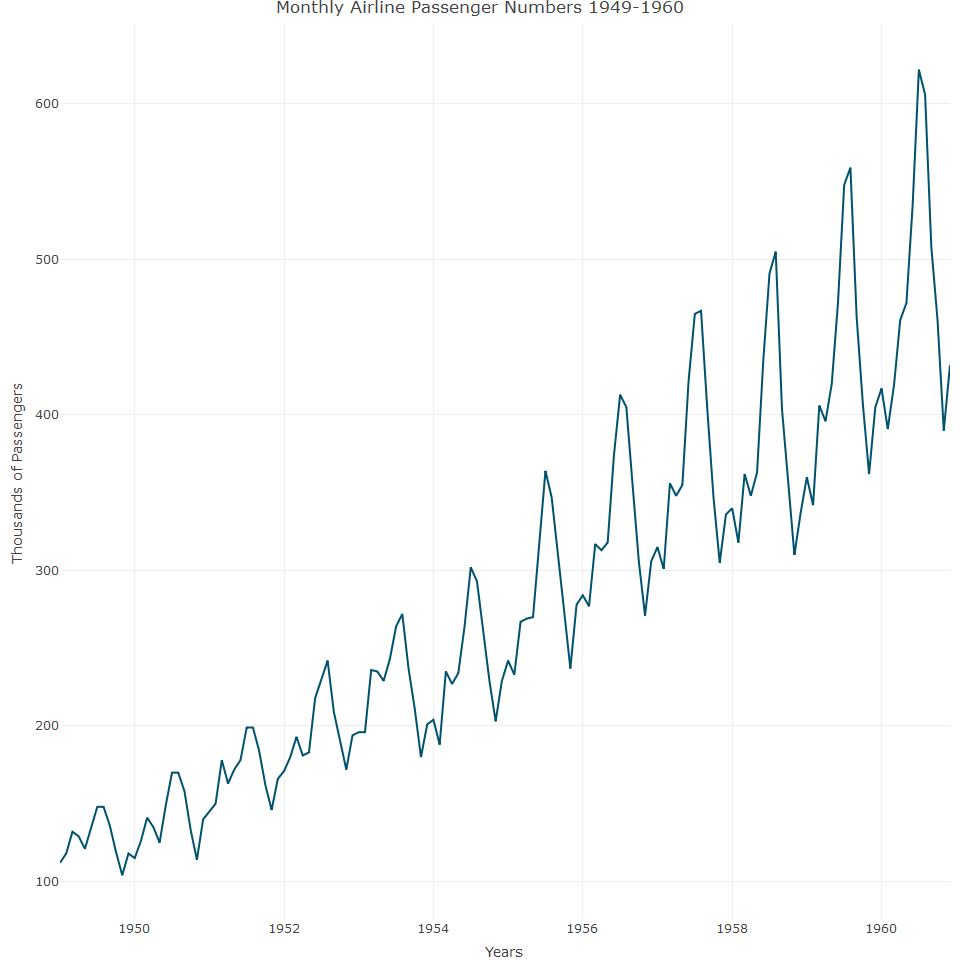
A additive series is whenever there is growth in the trend (with respect to the previous period) or if the amplitude of the seasonal component remains the same over time. A multiplicative series is whenever the growth of the trend or magnitude of the seasonal component increases or decreases by some multiplicity from period to period over time. The US monthly natural gas consumption is an example of additive series. It is easy to notice that the amplitude of the seasonal component remains the same (or close to the same) over time.



The amplitude of the USgas series seasonal component over the last 20 yeas did not change much (apart from some years, which may be related to some unusual weather patterns). Additionally, the series trend appears to be linear, with some structural breaks during 2010.

## The AirPassengers series is a ts object with 1 variable and 144 observations  
## Frequency: 12   
## Start time: 1949 1   
## End time: 1960 12

We can use data on air passengers, which describes the total monthly international air passengers between 1949 to 1960 as an example of a multiplicative series. As can be seen, the amplitude of the seasonal component increase from year to year.

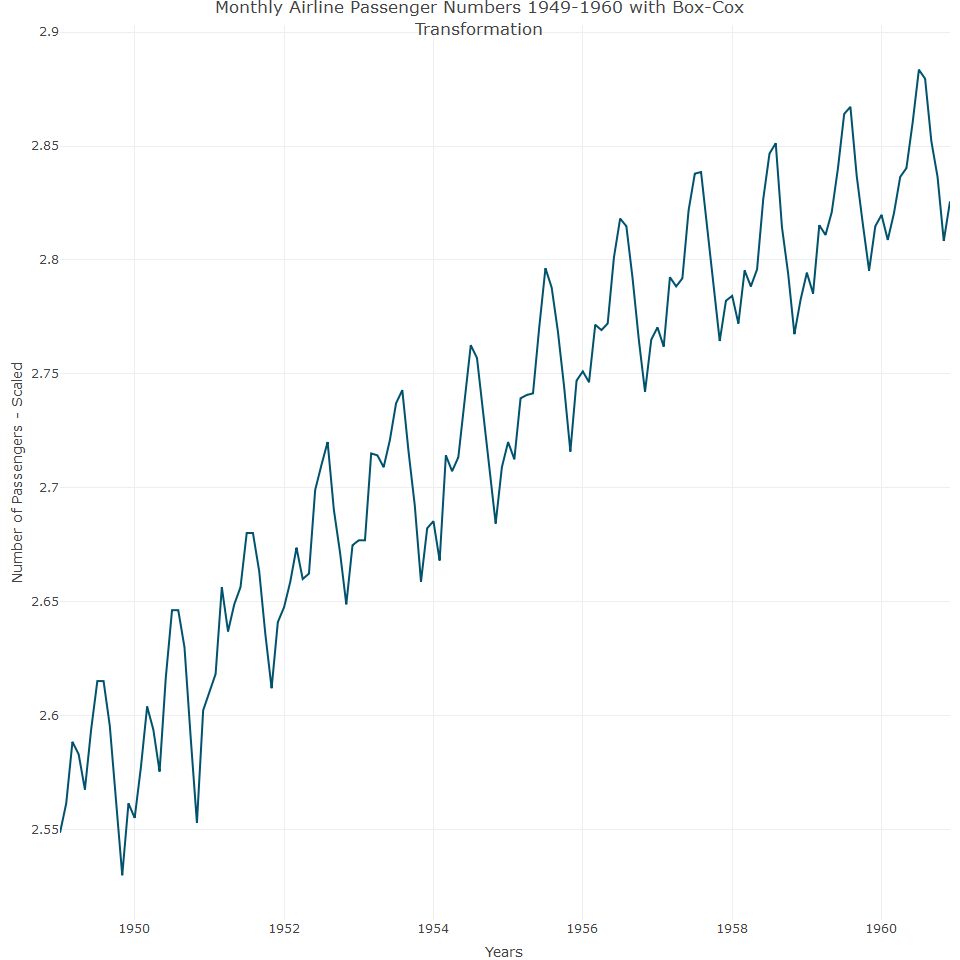


## Handling Multiplicative Series

Most forecasting models assume the variation of the input series remains constant over time, an assumption that generally holds for additive structure series, but not when it has multiplicative structure. The typical approach for handling a series with multiplicative structure is to apply data transformation on the input series. The most common transformation approaches are the log transformation and the box cox transformation. We can apply a box cox transformation and we first need to find the lamda () value, which is as below:

## [1] -0.2947156

We can now use the value to transform the input series with the Boxcox function and plot



As can be seen in the above plot, the values of the series are scaled.

## Decomposition of Time Series

Once the data has been cleaned and formated correctly, a first step is to identify the structure of the series components. Time series decomposition is a generic term for the process of separating a series into its components, and this process provides insights into the structural pattern of the series. Typically, these insights identify the most appropriate approaches to handle the series, based on the aim of the analysis (seasonality analysis, forecasting for example).

## Classical seasonal decomposition

This is one of the most common methods of decomposing a time series down to its components and is a three step process, where each step is dedicated to the estimation of one of the components in sequential order and hence the calculation of each component is derived from the estimation of the previous component:

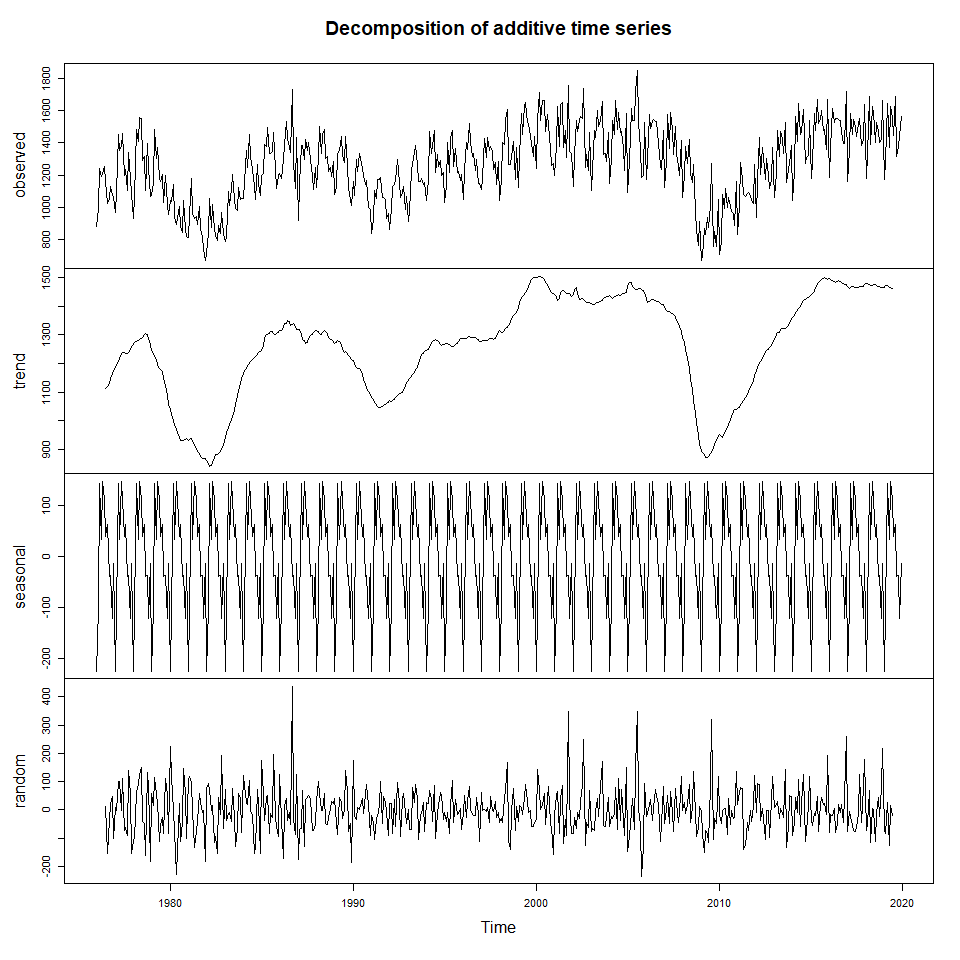
* **Trend Estimation:** This is the first stage of the decomposing process, by using MA function to remove the seasonal component from the series. The order of the MA function is defined by the frequency of the series. For instance, if the frequency of the input series is 12, then the MA order should also be 12. If a two-sided MA is used, then some of the first and last observations of trend estimation will be missing.
* **Seasonal Component estimation:** A two step process starting with detrending of the series by subtracting the trend estimation from the previous step from the series. After the series is detrended, the next step is to estimate the corresponding seasonal component for each frequency unit (for example, for a monthly series, the seasonal component for Janaury or February and so on). This calculation is done by grouping observations by their frequency unit and then averaging each group. The output of this process is a new series with a length that is equal to the series frequency and is ordered frequently. This series represents the seasonal component of each frequency unit, so that the estimation is one-to-many (one estimation has multiple observations). For example, if the input series has a monthly frequency, the seasonal component of observations that occurred in January will be the same across the series.
* **Irregular Component estimation:** This calculation involves subtracting the estimation of the trend and seasonal components from the original series. The decompose function carries out this estimation, so let’s review the structure of the decompose output:

## List of 6  
## $ x : Time-Series [1:528] from 1976 to 2020: 885 995 1244 1191 1203 ...  
## $ seasonal: Time-Series [1:528] from 1976 to 2020: -225.2 -102.4 143 34.5 147.9 ...  
## $ trend : Time-Series [1:528] from 1976 to 2020: NA NA NA NA NA ...  
## $ random : Time-Series [1:528] from 1976 to 2020: NA NA NA NA NA ...  
## $ figure : num [1:12] -225.2 -102.4 143 34.5 147.9 ...  
## $ type : chr "additive"  
## - attr(\*, "class")= chr "decomposed.ts"

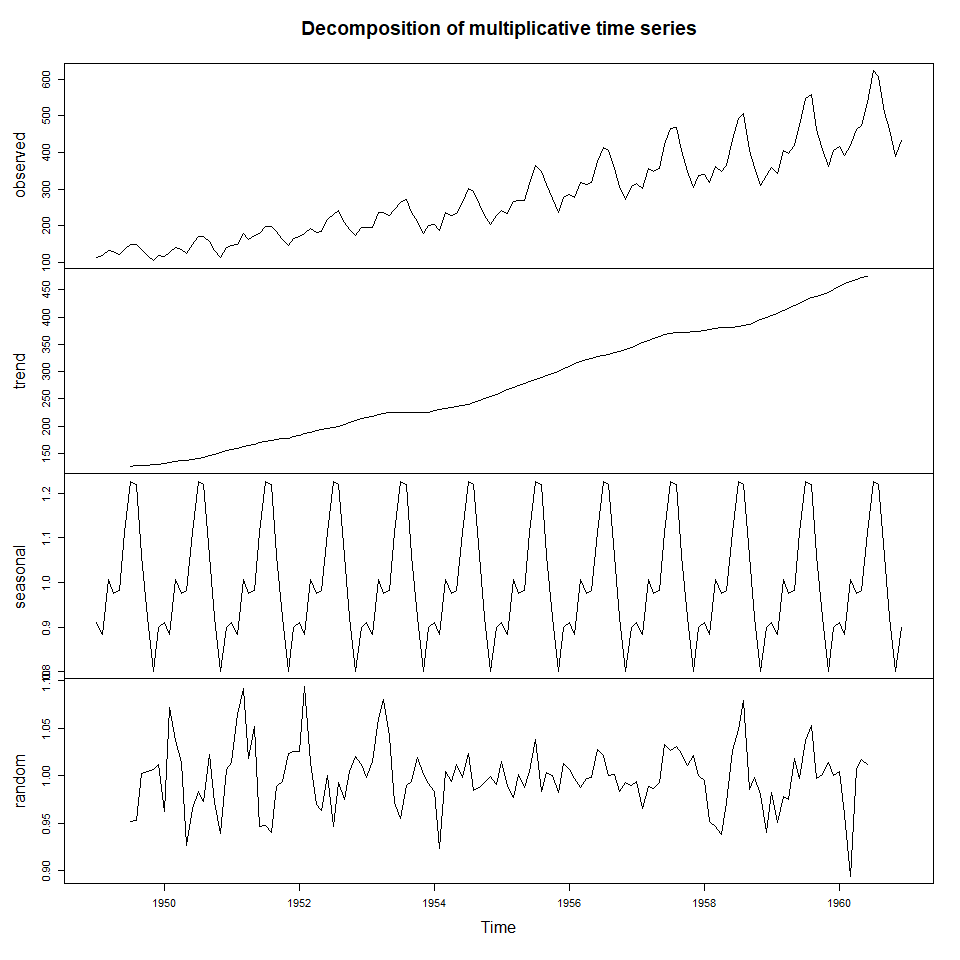
The preceding output returns a lis of six objects:

* **x:** the original series, a ts object
* **seasonal:** the estimate of the seasonal component, a ts object
* **trend:** estimate of the series trend. The first and last objects ae missing due to the use of two-sided MA function. The number of missing values is defined by the MA function
* **random:** the estimate of the irregular component, a ts object. This output is nothing but the remainder of the series and the preceding two components. The random object is missing whenever the trend estimation is missing.
* **figure:** estimated seasonal figure only
* **type:** type of decomposition, either additive (the default) or multiplicative

Let’s now plot the additive series



Now let’s plot the multiplicative model



A downside of classical decomposition is that seasonal component estimation is based on the arithmetic average, resulting in a one to many estimation, so there is a single seasonal componemnt estimation for each cycle unit (for example, all observations of the series that occurred in January will have the same seasonal component estimation if the series is monthly)