<https://www.askdavetaylor.com/create_custom_stock_market_index/>

Individual stocks essentially reflect investor expectations of the future value of a company. If you believe that a particular company is going to grow bigger, to earn more next year than it does this year, you therefore also believe that the company is going to be worth more as an entity. Its current value is essentially the stock price \* total number of shares of stock outstanding (e.g., issued and in the marketplace). This is what analysts call “market cap” and it can be calculated for any publicly traded firm quite easily.

the key indicator we want for a market index is the change in stock value over the desired period of time.

What period of time should we use? How about the usual 15-minute delayed ticker information of the value against its opening this morning?

For illustrative purposes, let’s grab a half-dozen additional tech stocks and see how they’re doing right now, as I write this blog entry: Microsoft (Nasdaq: MSFT) is +0.11, Apple (Nasdaq: AAPL) is +0.34, eBay (Nasdaq: EBAY) is +0.13, Yahoo (Nasdaq: YHOO) is -0.19 and Amazon (Nasdaq: AMZN) is -0.20.

A “favorite tech stocks index” can then be calculated by simply adding everything up: -0.04 + 0.11 + 0.34 + 0.13 + -0.19 + -0.20, giving us +0.15.

the S&P 500 is far more complicated than I’m suggesting, with a weighted floating value and all sorts of abstruse calculations to get its neat little number.

let’s just create our own index methodology based on the simplistic ideas presented herein. We’ll just add one more: a numeric index that’s actually the sum value of one share of each of the companies we’re tracking. If we go back to our tech stocks, here’s what we find out: Cisco is trading at $26.29, Microsoft at $31.02, Apple at $109.38, eBay at $33.99, Yahoo at $29.21 and Amazon at $61.49. Add them up and our base index number is 291.38. ADT Tech Six Index: 291.38 (+0.15).

#!/bin/sh

stocks=”CSCO MSFT EBAY AAPL AMZN YHOO”

getvalue=”/home/taylor/scripts/067-getstock.sh”

sumvalue=0

for stock in $stocks

do

value=”$($getvalue $stock)”

valuex100=”$(echo $value \\* 100 | bc | cut -d. -f1)”

echo “$stock is currently trading at: $value”

sumvalue=$(( $sumvalue + $valuex100 ))

done

indexvalue=”$(echo “scale=2; $sumvalue / 100″ | bc)”

echo “ADT Tech Six Index: $indexvalue”

exit 0

When run, the output is clean and interesting:

$ sh calculate-index.sh

CSCO is currently trading at: 26.39

MSFT is currently trading at: 31.00

EBAY is currently trading at: 34.01

AAPL is currently trading at: 109.76

AMZN is currently trading at: 61.68

YHOO is currently trading at: 29.14

ADT Tech Six Index: 291.98

Imagine now that you modify the script to only output that last value by itself, and that you now recalculate this every sixty minutes with a cron job (I assume that you’re running a Linux server for your Web site, as most people are). It’d look like this:

sh build-index > $webhome/current.index.value

Now all you need to do in your actual Web page is include that value where you want it displayed.

<https://www.coindesk.com/sp-cryptocurrency-contextualizing-bitcoins-price-explosion/>

Ash Bennington; 8/15/2017

As a baseline way of thinking about the rise in the price of bitcoin consider this: Over the last 90 years, the average annual rate of return on the S&P 500 index has been just 9.8 percent.

If we compare bitcoin's performance this year to the average return of the S&P 500, it's immediately clear bitcoin's moonshot rise has outperformed this benchmark by a stratospheric 6,000 percent.

For a slightly different proxy of price movement, let's take a look at how long it took the S&P 500 to double in value to its current level. At press time, the S&P 500 Index is trading around 2,466. Cutting that number in half, we get 1,233.

The last time the S&P 500 traded below that level was August of 2010. So it took the S&P 500 almost seven years to double – 30 times longer than it took bitcoin to do the same.

bitcoin's very nature makes it quite challenging to compare it to stocks in an apples-to-apples way.

One of the virtues of investing broadly in stocks via the S&P 500 Index is diversification.

The S&P 500 is made up of a basket of 500 companies, comprising many of the largest publicly traded corporations in America.

The stocks in the S&P 500 are drawn from 11 different sectors across 24 different industry groups. That kind of diversification means people are shielded, in large measure, from risks to any one company and, to a lesser extent, shielded from risks to any one industry group or market sector.

Bitcoin, on the other hand, is an investment in a single asset. In a certain sense, investing in bitcoin is roughly analogous to investing in a single stock.

Although that metaphor probably doesn't go far enough, still. What you're really investing in is a single implementation of one technology, a single instance of code.

This means that investing in bitcoin rather than the S&P massively concentrates your risk, and as we've seen today, that risk can equate to hundreds of dollars per bitcoin.

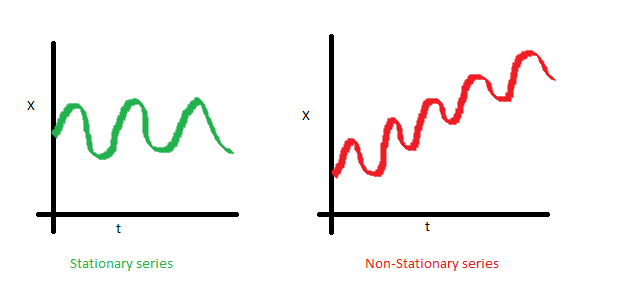
Sent LinkedIn request to Ash asking him to be a sponsor.

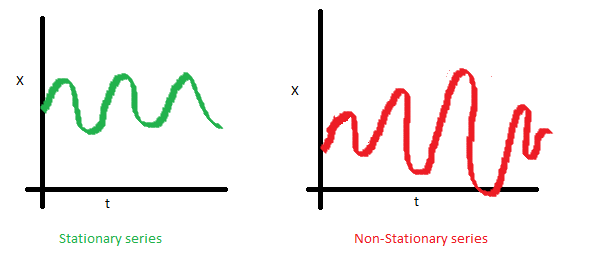
TimeSeries Modeling

<https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>

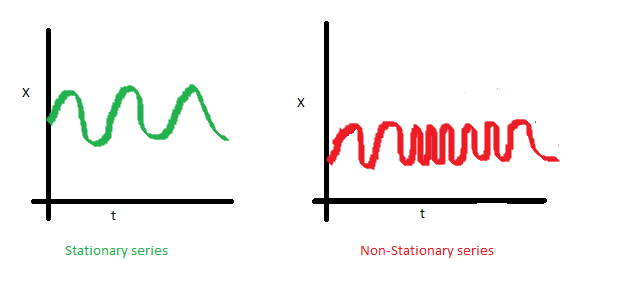
Is it stationary or non-stationary?

The mean of the series should not be a function of time rather should be a constant. The image below has the left hand graph satisfying the condition whereas the graph in red has a time dependent mean.



The variance of the series should not a be a function of time. This property is known as homoscedasticity. Following graph depicts what is and what is not a stationary series. (Notice the varying spread of distribution in the right hand graph)

The covariance of the i th term and the (i + m) th term should not be a function of time. In the following graph, you will notice the spread becomes closer as the time increases. Hence, the covariance is not constant with time for the ‘red series’.



The reason I took up this section first was that until unless your time series is stationary, you cannot build a time series model. In cases where the stationary criterion are violated, the first requisite becomes to stationarize the time series and then try stochastic models to predict this time series. There are multiple ways of bringing this stationarity. Some of them are Detrending, Differencing etc.

Dickey Fuller Test of Stationarity

=> X(t) - X(t-1) = (Rho - 1) X(t - 1) + Er(t)

We have to test if Rho – 1 is significantly different than zero or not. If the null hypothesis gets rejected, we’ll get a stationary time series.

oading the Data Set

Following is the code which will help you load the data set and spill out a few top level metrics.

> data(AirPassengers)

> class(AirPassengers)

[1] "ts"

#This tells you that the data series is in a time series format

> start(AirPassengers)

[1] 1949 1

#This is the start of the time series

> end(AirPassengers)

[1] 1960 12

#This is the end of the time series

> frequency(AirPassengers)

[1] 12

#The cycle of this time series is 12months in a year

> summary(AirPassengers)

Min. 1st Qu. Median Mean 3rd Qu. Max.

104.0 180.0 265.5 280.3 360.

Detailed Metrics

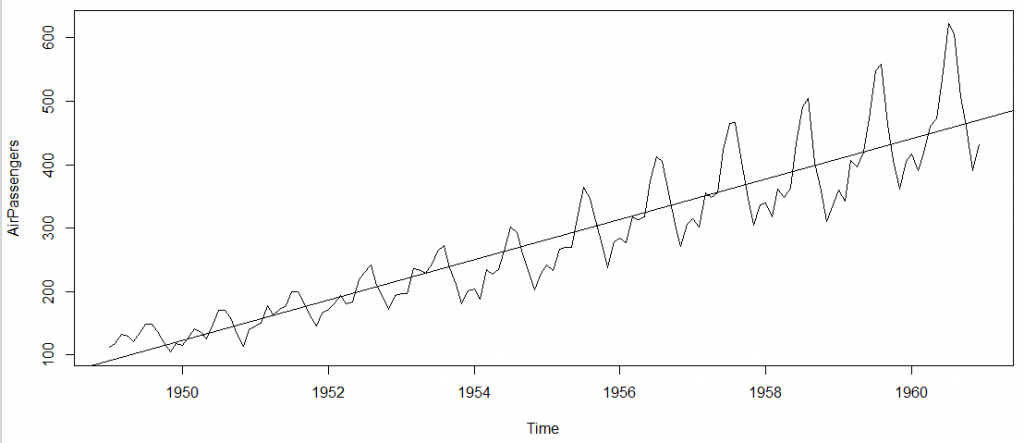
#The number of passengers are distributed across the spectrum

> plot(AirPassengers)

#This will plot the time series

>abline(reg=lm(AirPassengers~time(AirPassengers)))

# This will fit in a line



Here are a few more operations you can do:

> cycle(AirPassengers)

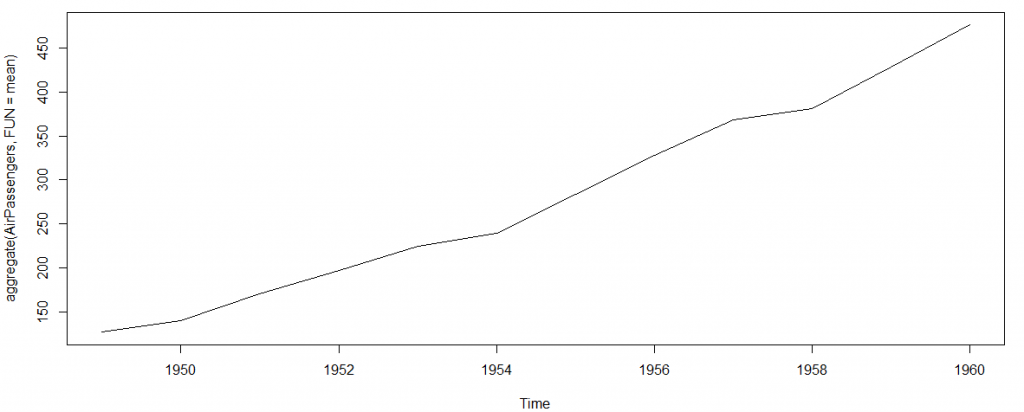
#This will print the cycle across years.

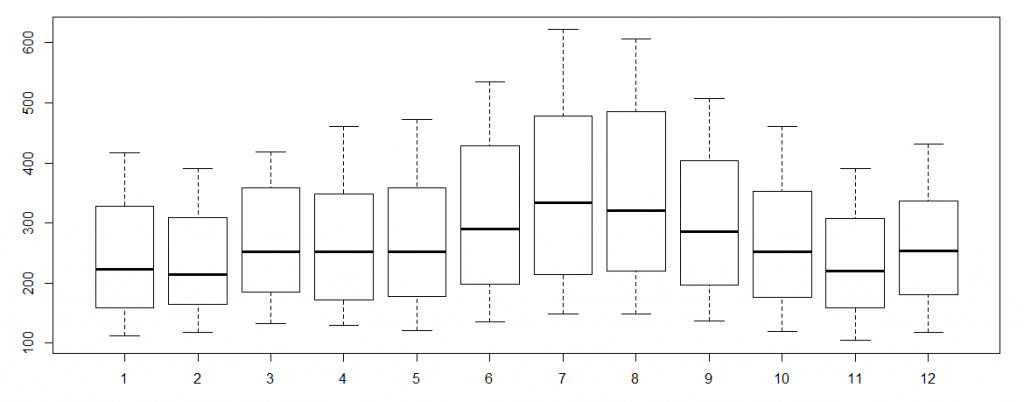
>plot(aggregate(AirPassengers,FUN=mean))

#This will aggregate the cycles and display a year on year trend

> boxplot(AirPassengers~cycle(AirPassengers))

#Box plot across months will give us a sense on seasonal effect





Important Inferences

The year on year trend clearly shows that the #passengers have been increasing without fail.

The variance and the mean value in July and August is much higher than rest of the months.

Even though the mean value of each month is quite different their variance is small. Hence, we have strong seasonal effect with a cycle of 12 months or less.

Exploring data becomes most important in a time series model – without this exploration, you will not know whether a series is stationary or not. As in this case we already know many details about the kind of model we are looking out for.

Introduction to ARMA Time Series Modeling

ARMA models are commonly used in time series modeling. In ARMA model, AR stands for auto-regression and MA stands for moving average.

In case you get a non stationary series, you first need to stationarize the series (by taking difference / transformation) and then choose from the available time series models.

Auto-Regressive Time Series Model

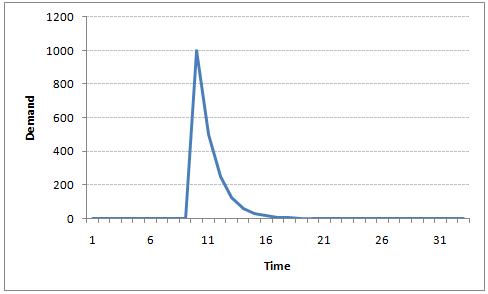
The current GDP of a country say x(t) is dependent on the last year’s GDP i.e. x(t – 1). The hypothesis being that the total cost of production of products & services in a country in a fiscal year (known as GDP) is dependent on the set up of manufacturing plants / services in the previous year and the newly set up industries / plants / services in the current year. But the primary component of the GDP is the former one.

Hence, we can formally write the equation of GDP as:

x(t) = alpha \* x(t – 1) + error (t)

This equation is known as AR(1) formulation. The numeral one (1) denotes that the next instance is solely dependent on the previous instance. The alpha is a coefficient which we seek so as to minimize the error function. Notice that x(t- 1) is indeed linked to x(t-2) in the same fashion. Hence, any shock to x(t) will gradually fade off in future.

For instance, let’s say x(t) is the number of juice bottles sold in a city on a particular day. During winters, very few vendors purchased juice bottles. Suddenly, on a particular day, the temperature rose and the demand of juice bottles soared to 1000. However, after a few days, the climate became cold again. But, knowing that the people got used to drinking juice during the hot days, there were 50% of the people still drinking juice during the cold days. In following days, the proportion went down to 25% (50% of 50%) and then gradually to a small number after significant number of days. The following graph explains the inertia property of AR series:



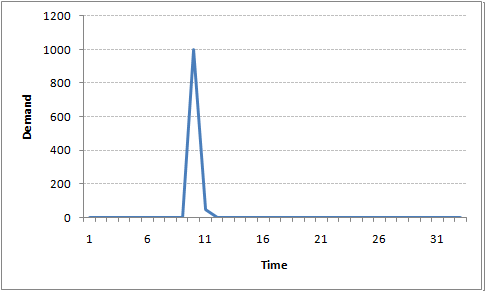
Moving Average Time Series Model

Let’s take another case to understand Moving average time series model.

A manufacturer produces a certain type of bag, which was readily available in the market. Being a competitive market, the sale of the bag stood at zero for many days. So, one day he did some experiment with the design and produced a different type of bag. This type of bag was not available anywhere in the market. Thus, he was able to sell the entire stock of 1000 bags (lets call this as x(t) ). The demand got so high that the bag ran out of stock. As a result, some 100 odd customers couldn’t purchase this bag. Lets call this gap as the error at that time point. With time, the bag had lost its woo factor. But still few customers were left who went empty handed the previous day. Following is a simple formulation to depict the scenario :

x(t) = beta \* error(t-1) + error (t)

If we try plotting this graph, it will look something like this :



Did you notice the difference between MA and AR model? In MA model, noise / shock quickly vanishes with time. The AR model has a much lasting effect of the shock.

Difference between AR and MA models

The primary difference between an AR and MA model is based on the correlation between time series objects at different time points. The correlation between x(t) and x(t-n) for n > order of MA is always zero. This directly flows from the fact that covariance between x(t) and x(t-n) is zero for MA models (something which we refer from the example taken in the previous section). However, the correlation of x(t) and x(t-n) gradually declines with n becoming larger in the AR model. This difference gets exploited irrespective of having the AR model or MA model. The correlation plot can give us the order of MA model.

Exploiting ACF and PACF plots

Once we have got the stationary time series, we must answer two primary questions:

Q1. Is it an AR or MA process?

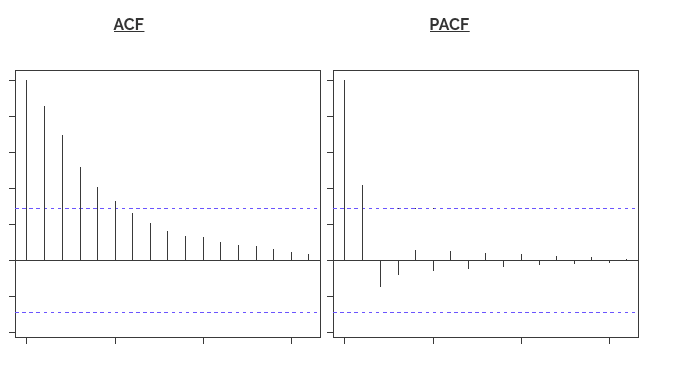
Q2. What order of AR or MA process do we need to use?

The trick to solve these questions is available in the previous section. Didn’t you notice?

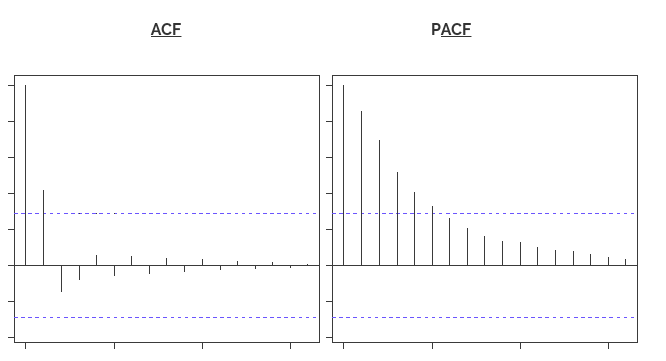
The first question can be answered using Total Correlation Chart (also known as Auto – correlation Function / ACF). ACF is a plot of total correlation between different lag functions. For instance, in GDP problem, the GDP at time point t is x(t). We are interested in the correlation of x(t) with x(t-1) , x(t-2) and so on. Now let’s reflect on what we have learnt above.

In a moving average series of lag n, we will not get any correlation between x(t) and x(t – n -1) . Hence, the total correlation chart cuts off at nth lag. So it becomes simple to find the lag for a MA series. For an AR series this correlation will gradually go down without any cut off value. So what do we do if it is an AR series?

Here is the second trick. If we find out the partial correlation of each lag, it will cut off after the degree of AR series. For instance,if we have a AR(1) series, if we exclude the effect of 1st lag (x (t-1) ), our 2nd lag (x (t-2) ) is independent of x(t). Hence, the partial correlation function (PACF) will drop sharply after the 1st lag. Following are the examples which will clarify any doubts you have on this concept :

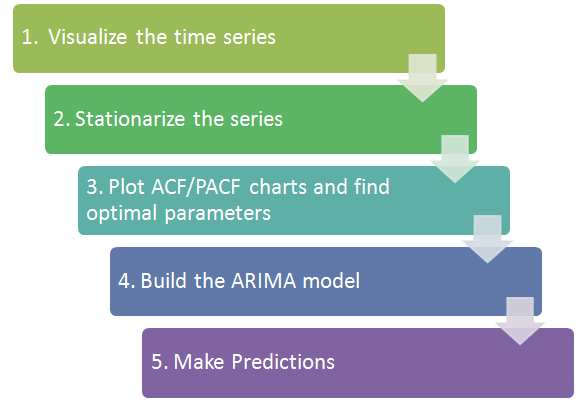


The blue line above shows significantly different values than zero. Clearly, the graph above has a cut off on PACF curve after 2nd lag which means this is mostly an AR(2) process.



Clearly, the graph above has a cut off on ACF curve after 2nd lag which means this is mostly a MA(2) process.

Till now, we have covered on how to identify the type of stationary series using ACF & PACF plots. Now, I’ll introduce you to a comprehensive framework to build a time series model.  In addition, we’ll also discuss about the practical applications of time series modelling.



Step 1: Visualize the Time Series

It is essential to analyze the trends prior to building any kind of time series model. The details we are interested in pertains to any kind of trend, seasonality or random behaviour in the series. We have covered this part in the second part of this series.

Step 2: Stationarize the Series

Once we know the patterns, trends, cycles and seasonality , we can check if the series is stationary or not. Dickey – Fuller is one of the popular test to check the same. We have covered this test in the first part of this article series. This doesn’t ends here! What if the series is found to be non-stationary?

There are three commonly used technique to make a time series stationary:

1. Detrending : Here, we simply remove the trend component from the time series. For instance, the equation of my time series is:

x(t) = (mean + trend \* t) + error

We’ll simply remove the part in the parentheses and build model for the rest.

2. Differencing : This is the commonly used technique to remove non-stationarity. Here we try to model the differences of the terms and not the actual term. For instance,

x(t) – x(t-1) = ARMA (p , q)

This differencing is called as the Integration part in AR(I)MA. Now, we have three parameters

p : AR

d : I

q : MA

3. Seasonality : Seasonality can easily be incorporated in the ARIMA model directly. More on this has been discussed in the applications part below.

Step 3: Find Optimal Parameters

The parameters p,d,q can be found using ACF and PACF plots. An addition to this approach is can be, if both ACF and PACF decreases gradually, it indicates that we need to make the time series stationary and introduce a value to “d”.

Step 4: Build ARIMA Model

With the parameters in hand, we can now try to build ARIMA model. The value found in the previous section might be an approximate estimate and we need to explore more (p,d,q) combinations. The one with the lowest BIC and AIC should be our choice. We can also try some models with a seasonal component. Just in case, we notice any seasonality in ACF/PACF plots.

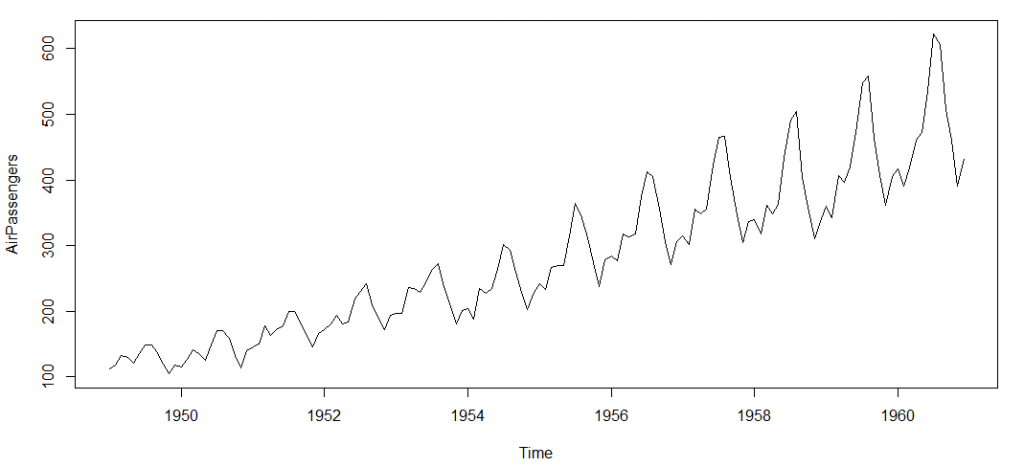
Step 5: Make Predictions

Once we have the final ARIMA model, we are now ready to make predictions on the future time points. We can also visualize the trends to cross validate if the model works fine.

Example:

Where did we start ?

Following is the plot of the number of passengers with years. Try and make observations on this plot before moving further in the article.



Here are my observations :

1. There is a trend component which grows the passenger year by year.

2. There looks to be a seasonal component which has a cycle less than 12 months.

3. The variance in the data keeps on increasing with time.

We know that we need to address two issues before we test stationary series. One, we need to remove unequal variances. We do this using log of the series. Two, we need to address the trend component. We do this by taking difference of the series. Now, let’s test the resultant series.

adf.test(diff(log(AirPassengers)), alternative="stationary", k=0)

Augmented Dickey-Fuller Test

data: diff(log(AirPassengers))

Dickey-Fuller = -9.6003, Lag order = 0,

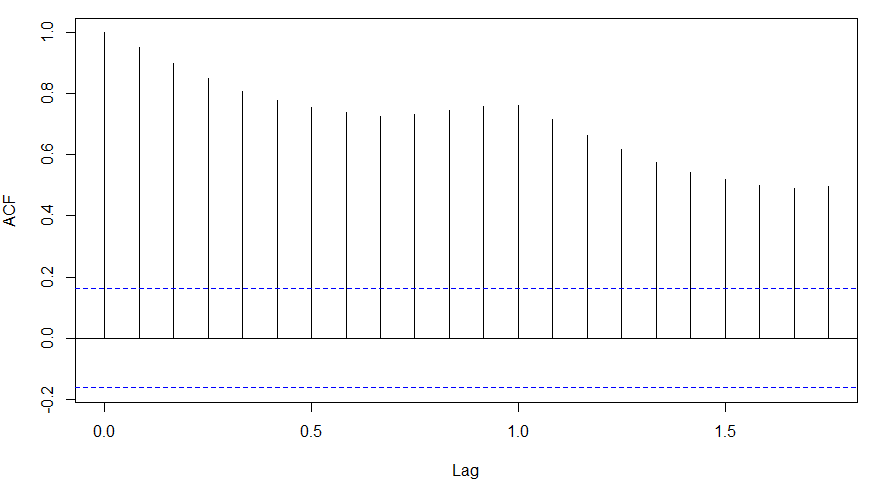
p-value = 0.01

alternative hypothesis: stationary

We see that the series is stationary enough to do any kind of time series modelling.

Next step is to find the right parameters to be used in the ARIMA model. We already know that the ‘d’ component is 1 as we need 1 difference to make the series stationary. We do this using the Correlation plots. Following are the ACF plots for the series :

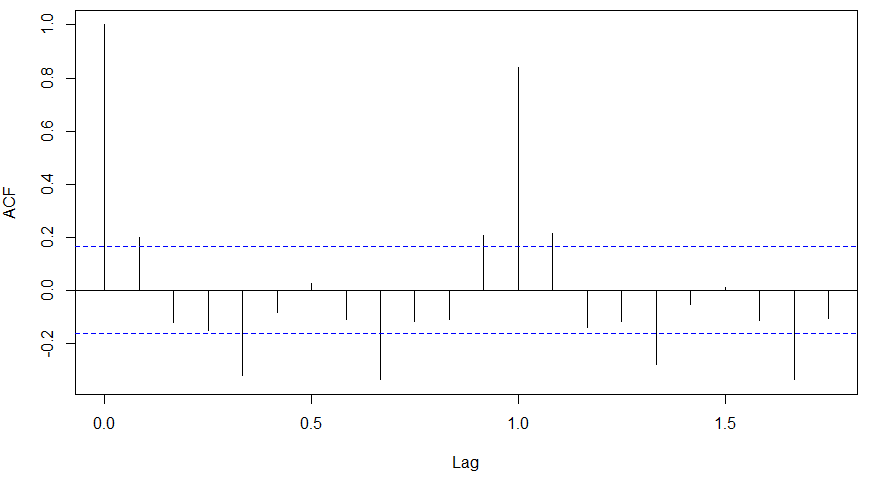
acf(log(AirPassengers))



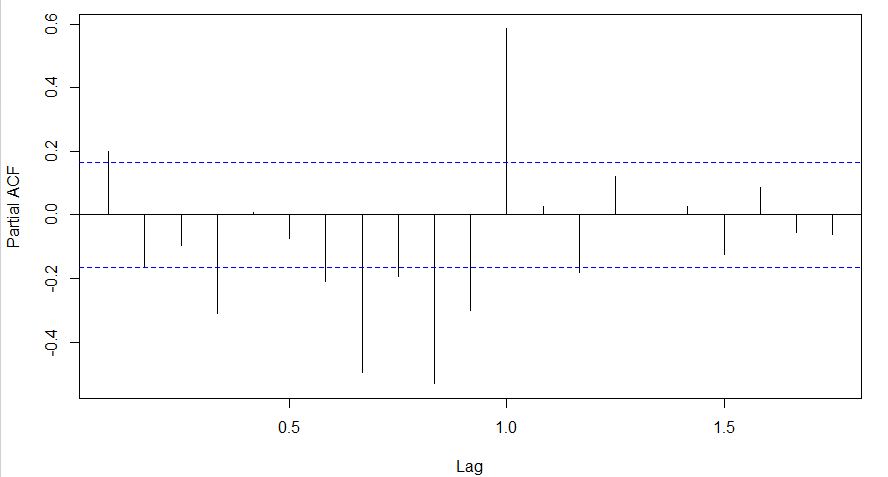
What do you see in the chart shown above?

Clearly, the decay of ACF chart is very slow, which means that the population is not stationary. We have already discussed above that we now intend to regress on the difference of logs rather than log directly. Let’s see how ACF and PACF curve come out after regressing on the difference.

acf(diff(log(AirPassengers)))



pacf(diff(log(AirPassengers)))



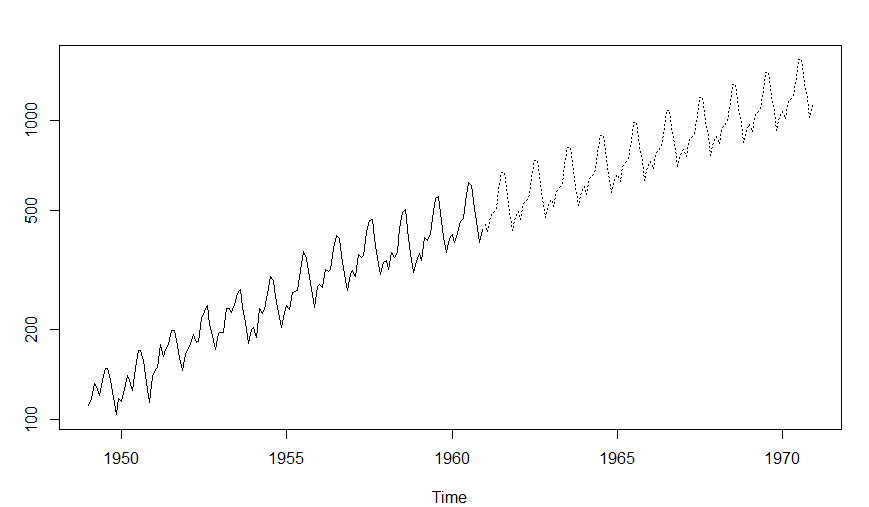
Clearly, ACF plot cuts off after the first lag. Hence, we understood that value of p should be 0 as the ACF is the curve getting a cut off. While value of q should be 1 or 2. After a few iterations, we found that (0,1,1) as (p,d,q) comes out to be the combination with least AIC and BIC.

Let’s fit an ARIMA model and predict the future 10 years. Also, we will try fitting in a seasonal component in the ARIMA formulation. Then, we will visualize the prediction along with the training data. You can use the following code to do the same :

(fit <- arima(log(AirPassengers), c(0, 1, 1),seasonal = list(order = c(0, 1, 1), period = 12)))

pred <- predict(fit, n.ahead = 10\*12)

ts.plot(AirPassengers,2.718^pred$pred, log = "y", lty = c(1,3))



<https://www.coindesk.com/10-must-read-cryptocurrency-research-papers-from-2015/>