LETTERKENNY INSTITUTE OF TECHNOLOGY

ASSIGNMENT COVER SHEET

Lecturer's Name: **James Connolly**

Assessment Title: **Prediction**

Work to be submitted to: **08/01/19**

Date for submission of work: **James Connolly**

Place and time for submitting work: ______

To be completed by the Student

Student's Name: **PRATEEK PARASHER**

Class: **MSc Big Data Analytics**

Subject/Module: **DATA SCIENCE**

Word Count (where applicable): ______

I confirm that the work submitted has been produced solely through my own efforts.

Student's signature: **PRATEEK PARASHER** Date: **08/01/19**

Notes

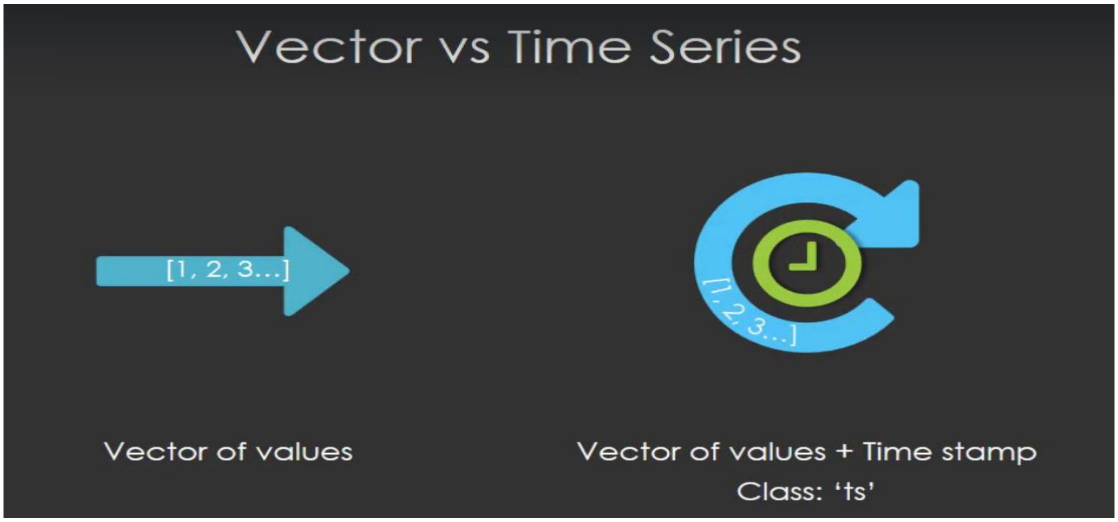
Penalties: The total marks available for an assessment is reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. Assessment work received more than two weeks late will receive a mark of zero. [Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute's Assessment Regulations.]

Plagiarism: Presenting the ideas etc. of someone else without proper acknowledgement (see section L1 paragraph 8).

Cheating: The use of unauthorised material in a test, exam etc., unauthorised access to test matter, unauthorised collusion, dishonest behaviour in respect of assessments, and deliberate plagiarism (see section L1 paragraph 8).

Continuous Assessment: For students repeating an examination, marks awarded for continuous assessment, shall normally be carried forward from the original examination to the repeat examination.

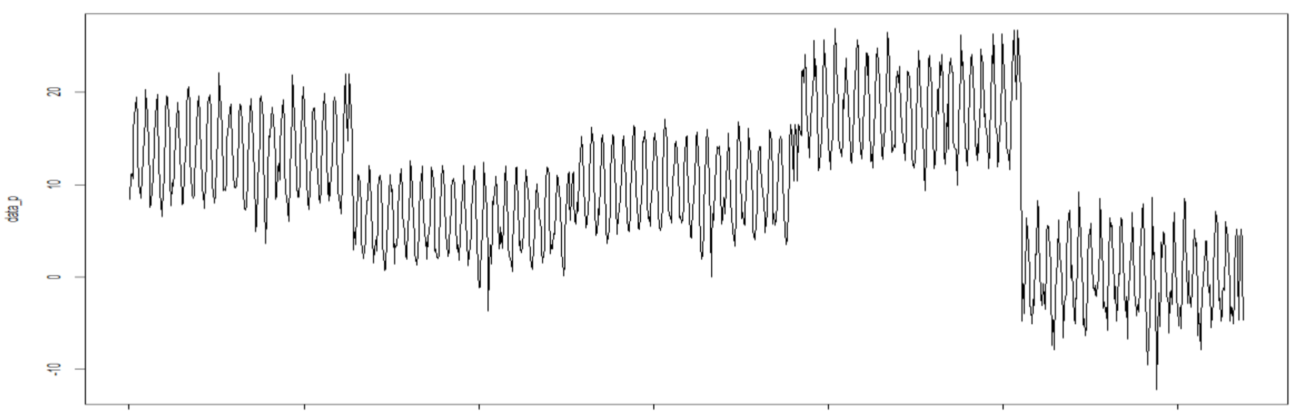
Time series specific functions require the data to be of a specific class. Without that class models visualizations and all sorts of functions do not work.




```
data_p <- read.csv("C:/Users/PRATEEK PARASHER/Downloads/TS_1.csv")

data_p = scan()

plot.ts(data_p)
```



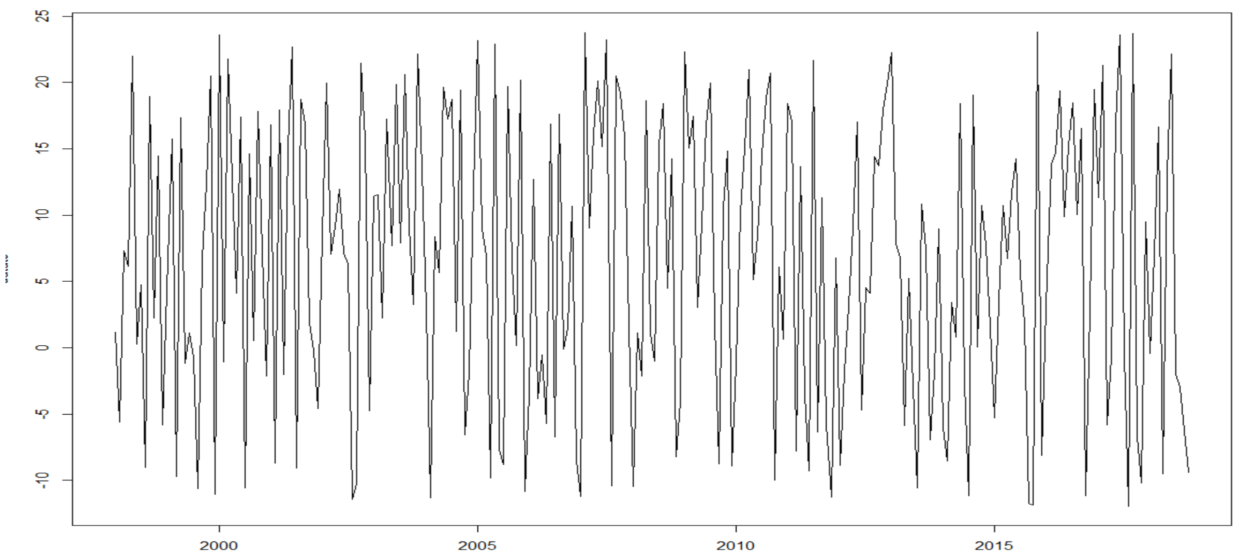

```
data_p =runif(n=251 , min= -12 , max= 24)
```

datats = ts(data = data_p,

c(start = 1998,1) , c(end = 2018,10) , frequency = 12)

- using the dublin weather data set which clearly has stable seasonality and no trend.

plot(datats)



class(datats)

[1] "ts"

Decompose One Time Series into Multiple Series

Time series decomposition is a mathematical procedure which transforms a time series into multiple different time series. The original time series is often split into 3 component series:

- Seasonal: Patterns that repeat with a fixed period of time. For example in this case weather temp have fixed patterns with different seasons summer and winter.
- Trend: The underlying trend of the metrics.
- Random: Also call “noise”, “irregular” or “remainder,” this is the residuals of the original time series after the seasonal and trend series are removed.

Decomposing Time Series (U)

frequency(datats)

[1] 12

length(datats)

[1] 250

decompose(datats, type = "additive")

```
> decompose(datats, type = "additive")
$`x`
             Jan          Feb          Mar          Apr          May          Jun          Jul          Aug          Sep
1998   1.15635370  -5.59150267   7.29793394   6.14215363  22.02126373   0.28756622   4.72969587  -9.02887746  18.96384646
1999   4.37141873  15.72737134  -9.74460006  17.34293382  -1.15886912   1.12815701  -0.76393949 -10.65468126   6.67697724
2000  23.60681775  -1.09604551  21.80898242  12.75387294   4.09333181  17.39321300 -10.57802888  14.60514068   0.55390177
2001  16.83413563  -8.70079099  17.95587987  -2.03321739  15.11196359  22.70949252  -9.06720384  18.75713087  16.88057582
2002   8.49777586  19.96998686   7.06970729   9.13051164  11.96646513   7.07612851   6.35434438 -11.41822269 -10.28244054
2003  11.40941401  11.53560057   2.22913541  17.24297383   7.69284304  19.86477546   7.91930355  20.61773202   9.14860921
2004   2.87899469 -11.33889299   8.40242698   5.64228431  19.63226476  17.26269676  18.74759919   1.24425412  19.46219686
2005  23.18580876   8.97863052   6.95724286  -9.81183990  22.88539909  -7.67146337  -8.81892116  19.70750814   6.30134267
2006  -3.45973089  12.71365264  -3.87053676  -0.53419483  -5.73658260  16.85146501  -6.73384591  17.61782610  -0.08407817
2007 -11.20822625  23.74961209   9.02179844  16.14563034  20.10826041  15.17515633  23.25133428 -10.41070555  20.52243185
2008 -10.48411025   1.11879894  -2.14495429  18.61682931   1.04646165  -1.03879954  15.32887222  18.41404648   4.48517620
2009  22.34224492  15.07335148  17.48239380   3.03997566   9.37818575  16.75764808  19.98271141   1.86689736  -8.74840497
2010  -1.18136600  10.12423766  15.10555409  21.01405679   5.12701820   8.41630078  14.51833246  18.86810728  20.72927416
2011  18.44002043  17.03166590  -7.77607582  13.65671598  -3.53032111  -9.30155696  21.68326130  -6.34195829  11.29406460
2012  -8.89399584  -1.81191554   3.13565644   9.14201247  17.04099947  -4.72817248   4.53928818   4.13315210  14.41491930
2013  22.29520466   7.82220199   6.80408758  -5.85486281   5.25377616  -2.86245133 -10.56789707  10.85281835   7.42419291
2014  -6.05906788  -8.53316002   3.43007129   0.80741637  18.41760744  -2.63523095 -11.16679828  19.04638343   0.04516660
2015  -5.30857146   3.44186128  10.70784955   6.75032465  11.74636814  14.25699353   5.52562344   1.84043816 -11.75399368
```

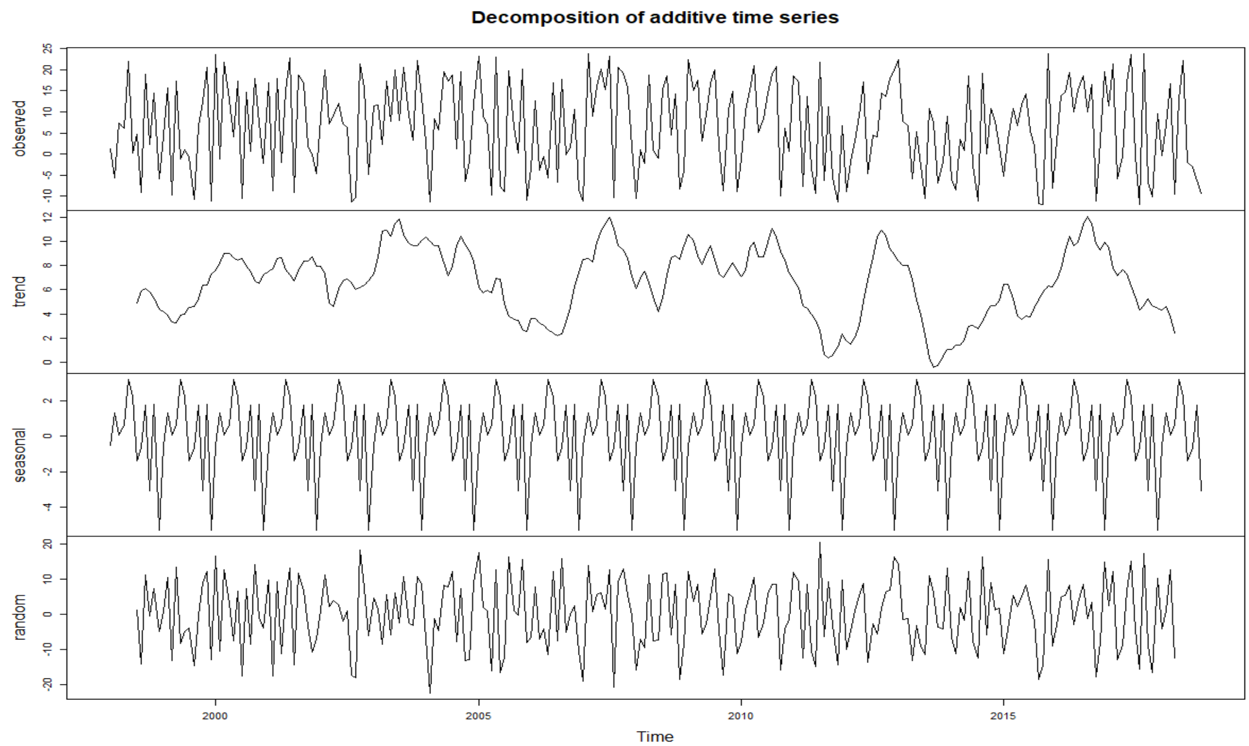
```
$figure
 [1] -0.52789200  1.30988939  0.06015542  0.64465260  3.17290886  2.22344037 -1.37653154 -0.65798286  1.72072162 -3.07048934
[11]  1.80575978 -5.30463230

$type
[1] "additive"

attr(,"class")
[1] "decomposed.ts"
> 
```

plot(decompose(datats, type = "additive"))

using the Dublin weather data set which clearly has stable seasonality and no trend. So that can be perfectly described with an additive model. Generally, if the amplitude of the seasons stay roughly the same that means the distance between highs and lows of the season do not constantly increase over time.



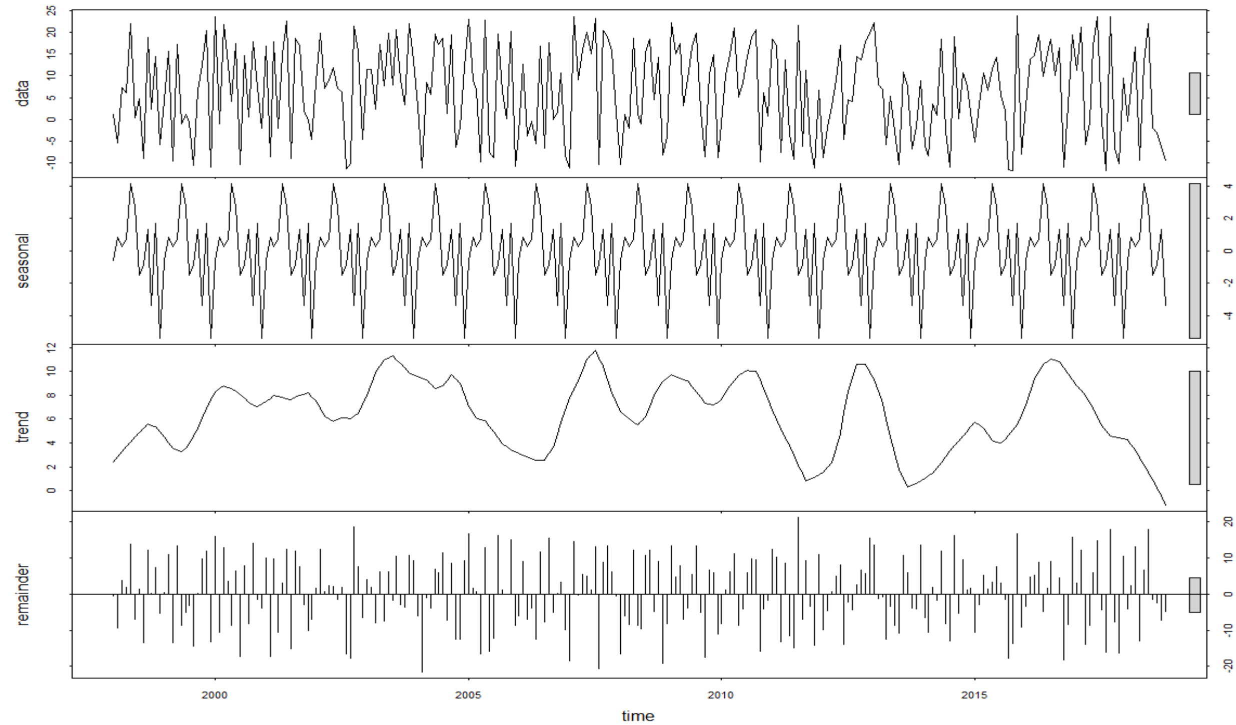
I have to evaluate for trend seasonality and white noise for each time point except some points at the beginning and the end which are needs to compute the whole model. I can also see the trend component or here and the seasonality. Now the trend line goes up and down over the 20 years there are some peaks at around 2003, 2007, 2016. But overall the main stays pretty constant and there is no clear direction of the trend curve. Hence there is no trend at all in the dataset. The seasonal part on the other side is quite clear to recognize and it stays constant over the whole time series and then just in nature of this type of decomposition. The seasonal component stays totally constant over the whole series. So again this sort of graph is very useful whenever you're working with time series data that is seasonal

library(forecast)

library(ggplot2)

 # library(ggplot2 and forecast)

autoplot(decompose(datats, type = "additive"))



Now as already mentioned an alternative to the rather primitive decompose is the function STL which again if I wrap it in a plot command i would get a handy visualization of your data set for this function to work however, need the argument stopped window which is the seasonal window to be used in order to calculate the seasonal part. could set it to periodic or just provide an odd number of 7 or higher.

 # alternatively the function stl could be used

plot(stl(datats, s.window="periodic"))

stl(datats, s.window="periodic")

 # seasonal adjustment

mydatats = decompose(datats, "additive")

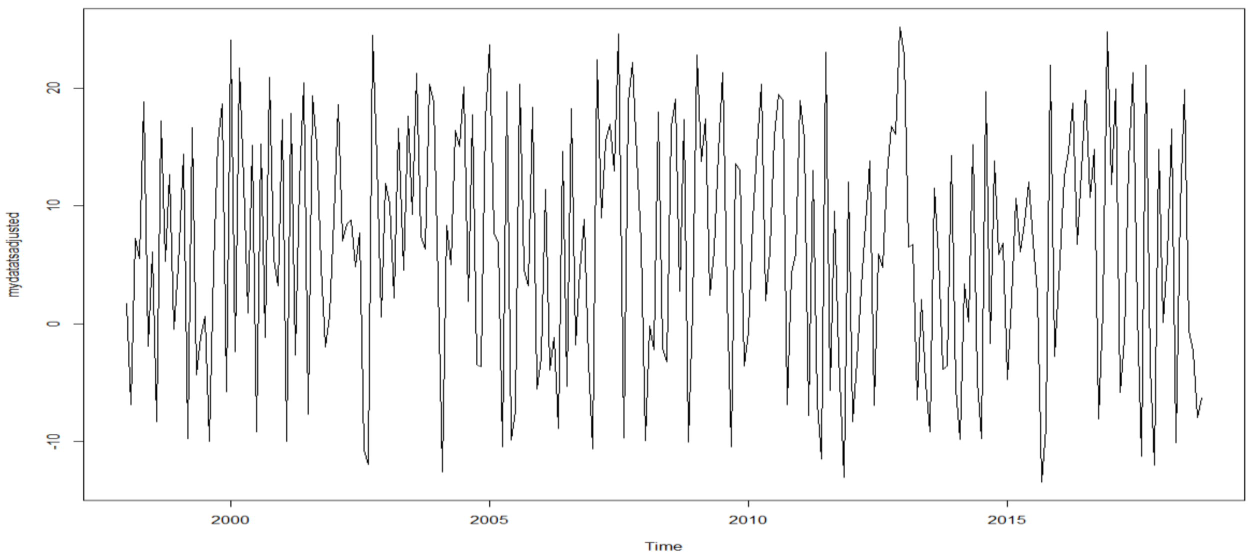
class(mydatats)

 # we are subtracting the seasonal element

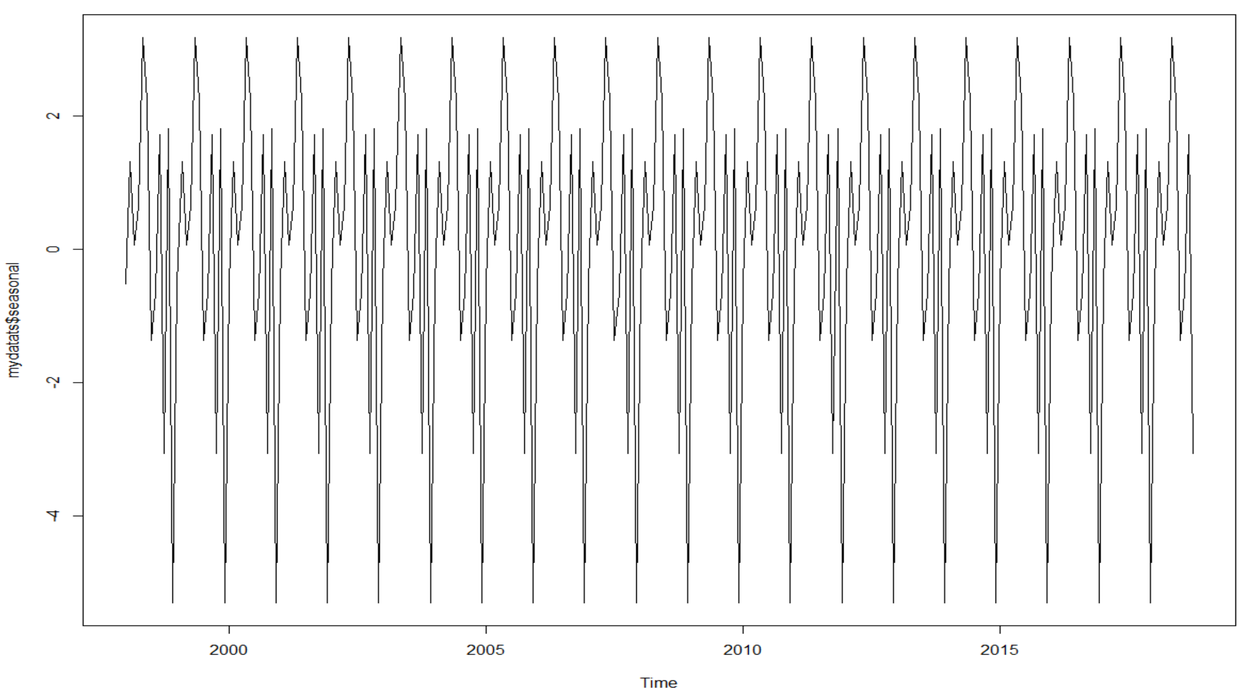
mydatatsadjusted = datats-mydatats$seasonal

 # getting a plot

plot(mydatatsadjusted)



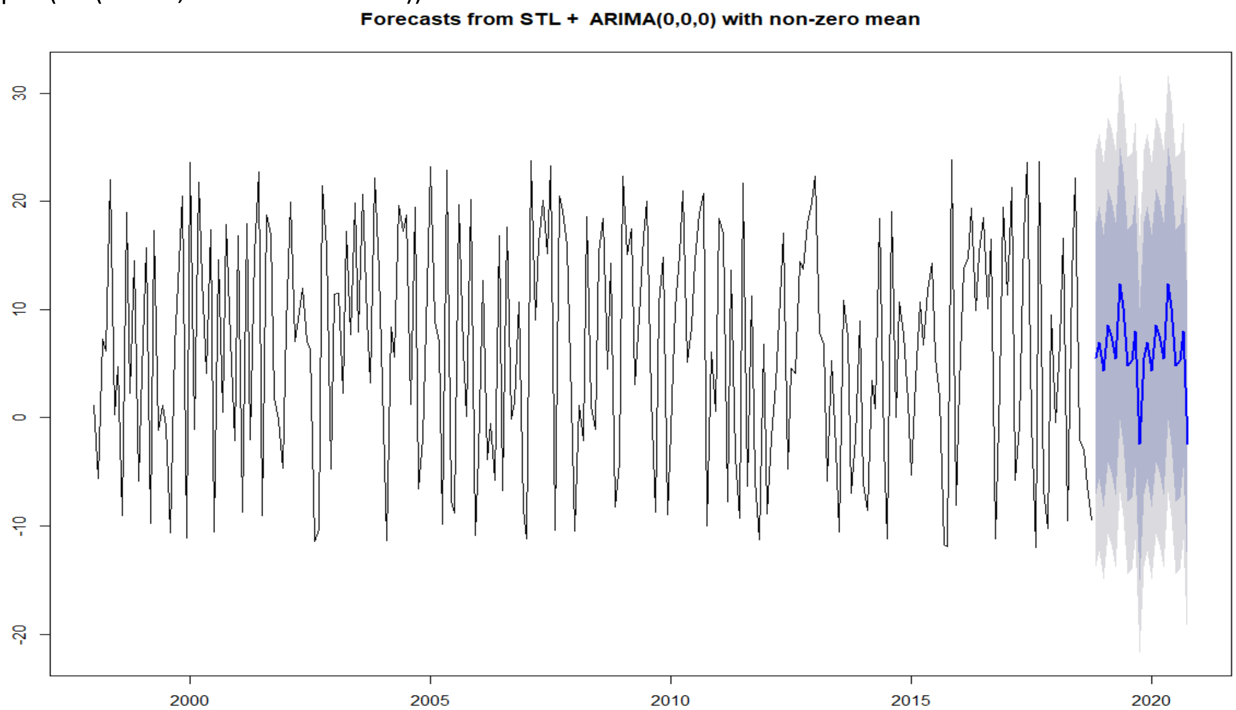
plot(mydatats$seasonal)



a stl forecast from the package forecast

library(forecast)

plot(stlf(datats, method = "arima"))



The forecast is visible in the blue section like always with these forecast plots the length of the forecast the interval was auto generated by are based on the length of the already existing time series. I would routinely use an age of doubled the frequency which is two times 12. The settings for the Arima model were automatically adjusted and optimized by our. So as i can see seasonal decomposition offers some really exciting opportunities for seasonal datasets.

GitHub Link :- https://github.com/prateekparasher/web_scrap