Time Series Analysis Part 1

Manuela da Cruz Chadreque

25 de Abril de 2019

## Time Series Exploratory Analysis

### ## Time Series Exploratory Analysis

Time Series are data that from the observation of a real phenomenon for a period of time. We have at our disposal, a wide range of tools and techniques to identify models that can help us to:

1. Describe relevant patterns on the time series;
2. Explain how the past can affect the future;
3. Explain how can two series can interact;
4. Forecast the future.

## Time Series Exploratory analysis

Usually the first step when analyzing a time series is to plot our data so we can visualize features such as the patterns, unusual observations, changes over time or relationship between variables. I this step it helps, if we answer the following questions:

1. Is there a **trend** over time? meaning that, on **average**, the measurements tend to increase (or decrease) over time?
2. Is there **seasonality**, meaning that there is a regularly repeating pattern of highs and lows related to calendar time such as seasons, quarters, months, days of the week, and so on?
3. Are their **outliers**? In regression, outliers are far away from your line. With time series data, your outliers are far away from your other data.
4. Is there a **long-run cycle** or period unrelated to seasonality factors?
5. Is there constant **variance** over time, or is the variance non-constant? Are there any **abrupt changes** to either the level of the series or the variance?

For our exploratory analysis we have choose the following data sets:

* OR Tambo South African Airport Montly average Temperatures;
* Eskom Electricity sales to Mozambique from 1996 to 2006;
* Rwanda annual GDP. 1960-2018
* Daily Dam levels for major and minor dams in the Western Cape water supply system. 2011-2019
* And Daily closing Shoprite stock prices.

On the present article we will use the r, the following post covers the same steps in python.

1. Import the data set, use
   1. [read\_csv()](<https://readr.tidyverse.org/reference/read_delim.html>) to import the csv file
   2. [ts()](<https://www.rdocumentation.org/packages/stats/versions/3.6.1/topics/ts>) function or [xts()](<https://www.rdocumentation.org/packages/xts/versions/0.11-2/topics/xts>) function to create a time series object
2. Clean and plot the dataset;
3. Analise.

### Temperatures

Step 1: Create a Time Serie Object

#Read the data set

temp<-read\_csv("https://www.dropbox.com/s/pf9ah91vrirzs8m/ORTamboTemp.csv?dl=1", skip=3)

## Create a time serie(ts) object

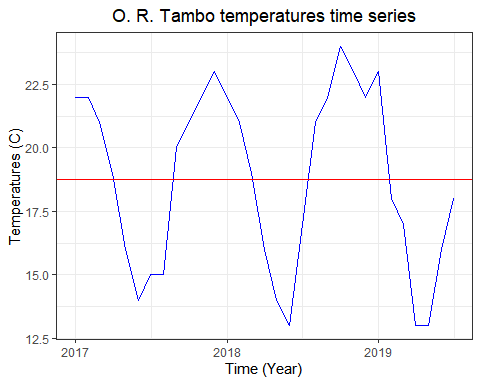
ts.temp<-ts(temp,start=2017, frequency=12)#this makes sure R knows that temp is a time series, starting Jan 2017  
ts.temp

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2017 22 22 21 19 16 14 15 15 20 21 22 23  
## 2018 22 21 19 16 14 13 17 21 22 24 23 22  
## 2019 23 18 17 13 13 16 18

#frequency=12 - means that monthly data

Step 2: Plotting your data to visualize the data set we can use [autoplot function](https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_ts.html) or[plot.ts](https://www.uni-muenster.de/ZIV.BennoSueselbeck/s-html/helpfiles/ts.plot.html).We will use the autoplot function.

# Wde can now use everything we know about ggplot to make the plot nicer  
autoplot(ts.temp,colour = 'blue') +   
 ggtitle("O. R. Tambo temperatures time series") +   
 geom\_hline(yintercept=mean(ts.temp), color = "red")+ #add a mean horizontal line   
 xlab("Time (Year)") +   
 ylab("Temperatures (C)") +  
 theme\_bw() +  
 theme(plot.title = element\_text(hjust = 0.5))



Analysist

1. Trend: yes(do not have a constant mean-using the abline we can evaluate if there is a trend or not) sazonalyty:yes

* Outliers: No
* cycle: No
* variance: do not have a constant variance

### Anual Eskom Electricity Sales to Mozambique from 1996 to 2006

Eskom is a South African electricity public utility, established in 1923 as the Electricity Supply Commission (ESCOM). The utility is the largest producer of electricity in Africa exporting electricity to Botswana, Mozambique, Namibia, Zimbabwe, Lesotho, Swaziland and Zambia. Our dataset is a subset of the [Eskom Table - Table 2.csv](http://energydata.uct.ac.za/dataset/eskom-holdings-limited-annual-report/resource/18894e94-873e-4617-95f2-b6fa2e3131f4)

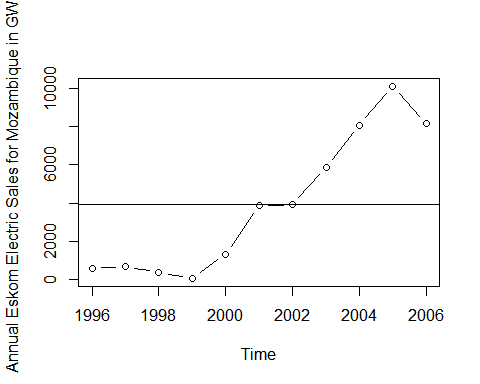
Elecsales<-read\_csv("https://www.dropbox.com/s/950n1na0omcvi7n/eskom-table-table-2\_Mozambique.csv?dl=1", skip=3)

## Parsed with column specification:  
## cols(  
## Sales = col\_double()  
## )

Elecsales<-ts(Elecsales,start=1996, frequency=1)  
Elecsales

## Time Series:  
## Start = 1996   
## End = 2006   
## Frequency = 1   
## Sales  
## [1,] 596  
## [2,] 680  
## [3,] 385  
## [4,] 68  
## [5,] 1331  
## [6,] 3899  
## [7,] 3907  
## [8,] 5875  
## [9,] 8076  
## [10,] 10108  
## [11,] 8167

plot.ts(window(Elecsales, start=1996, end=2006),type="b",xlab="Time",ylab="Annual Eskom Electric Sales for Mozambique in GWh") #time series plot of temp with points market as bubbles  
abline(h=mean(Elecsales)) # the mean value of the time series



Analysist

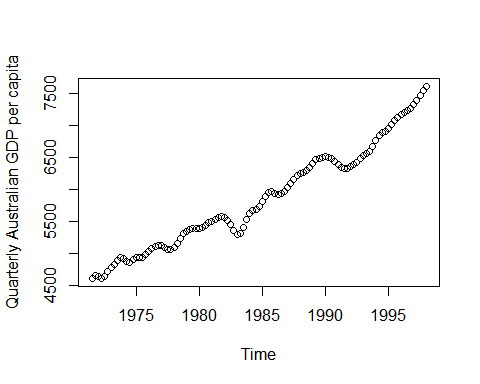
* Trend:
* sazonalyty:
* Outliers:
* cycle:
* variance:

Australia GDP per capital 1971-1998 We generate our dataset from the World Bank DataBank [World Bank national accounts data, and OECD National Accounts data files.](https://databank.worldbank.org/reports.aspx?source=2&country=RWA)

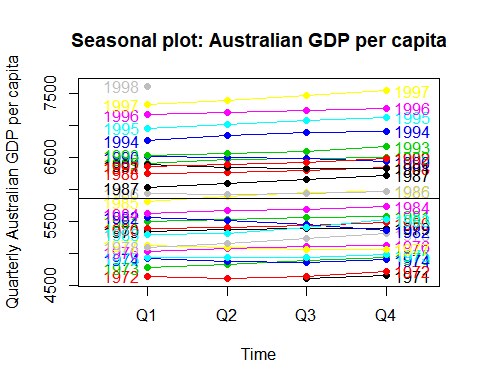
[https://databank.worldbank.org/reports.aspx?source=2&country=RWA#](https://databank.worldbank.org/reports.aspx?source=2&country=RWA)

plot.ts(window(ausgdp, start=1971, end=1998),type="b",xlab="Time",ylab="Quarterly Australian GDP per capita") #time series plot of temp with points market as bubbles

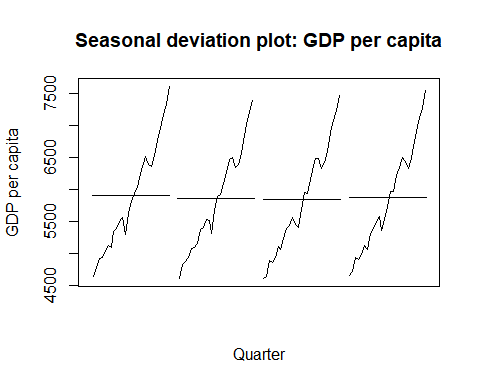
## Warning in window.default(x, ...): 'start' value not changed



seasonplot(ausgdp,ylab="Quarterly Australian GDP per capita", xlab="Time", main="Seasonal plot: Australian GDP per capita", year.labels=TRUE, year.labels.left=TRUE, col=1:20, pch=19)  
abline(h=mean(ausgdp)) # the mean value of the time series



monthplot(ausgdp,ylab="GDP per capita",xlab="Quarter",xaxt="n", main="Seasonal deviation plot: GDP per capita")



#axis(1,at=1:4,labels=quarter.abb,cex=0.8)

Analysist

* Trend:
* sazonalyty:
* Outliers:
* cycle:
* variance:

Dam levels for major and minor dams in the Western Cape water supply system. 2011-2019

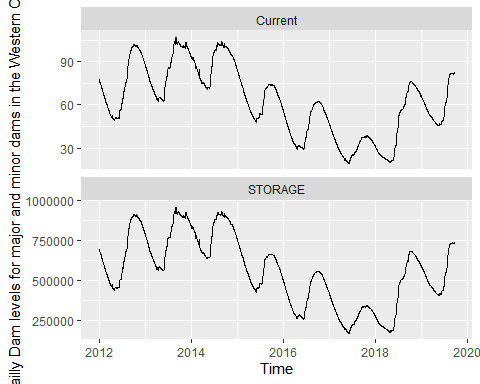
DamLevels<-read\_csv("https://www.dropbox.com/s/0mj1dru4cn53jnj/Dam%20levels%202012%20to%202019.csv?dl=1")

## Parsed with column specification:  
## cols(  
## DATE = col\_character(),  
## STORAGE = col\_double(),  
## Current = col\_double()  
## )

DamLevels<-column\_to\_rownames(DamLevels, var = "DATE")  
DamLevels<-xts(DamLevels, order.by =as.Date(rownames(DamLevels),format="%d-%b-%y") )  
tail(DamLevels)

## STORAGE Current  
## 2019-09-18 736670 82.0  
## 2019-09-19 736574 82.0  
## 2019-09-20 735718 81.9  
## 2019-09-21 736985 82.0  
## 2019-09-22 738124 82.2  
## 2019-09-23 738870 82.3

autoplot(DamLevels["2012/2019-09"],xlab="Time",ylab="Dailly Dam levels for major and minor dams in the Western Cape") #time series plot of temp with points market as bubbles



#Trend - maybe, seasonality - yes, outlier - maybe  
#abline(h=mean(DamLevels))

Analysist

* Trend:
* sazonality:
* Outliers:
* cycle:
* variance:

Daily closing Shoprite stock prices

Single-column extractor functions

Op() - opening price

Hi() - high price

Lo() - low price

Cl() - close price

Vo() - traded volume

Ad() - adjusted close price

tmp\_file<-"https://www.dropbox.com/s/ix5520k1t3cdmho/SRGHY\_Close.csv?dl=1"  
  
  
# Create dat by reading tmp\_file  
Shopriteclose<-read\_csv(tmp\_file, skip=2)

## Parsed with column specification:  
## cols(  
## Date = col\_character(),  
## Close = col\_double()  
## )

Shopriteclose<-column\_to\_rownames(Shopriteclose, var = "Date")  
Shopriteclose<-xts(Shopriteclose, order.by = as.Date(rownames(Shopriteclose), "%m/%d/%Y"))

Analysist

* Trend:
* sazonalyty:
* Outliers:
* cycle:
* variance:

## simple Forecasting Methods

### Average Method

The forecast of all future values are equal to the mean

### Naive method

Forecasts equal to last observed value

###Seasonal naive method Forecasts equal to last value from same season.

where: m is the seasonal period

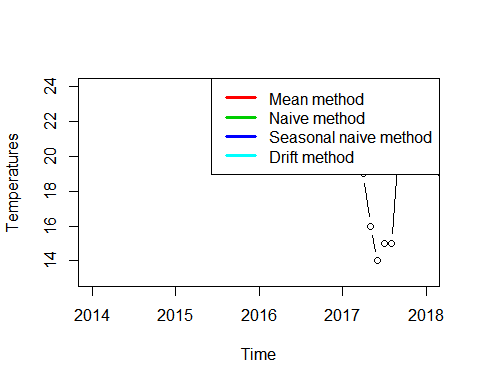
Drift method Forecasts equal to last value plus average change. Equivalent extrapolating a line drawn between first and last observations.

##Decomposition methods Decomposition methods are based in the decomposition of three movements or strengths: **trend**, **seasonality** and **noise**:

##ARIMA models (Autoregressive Integrated Moving Average) Models that relate the present value of a series to past values and past prediction errors

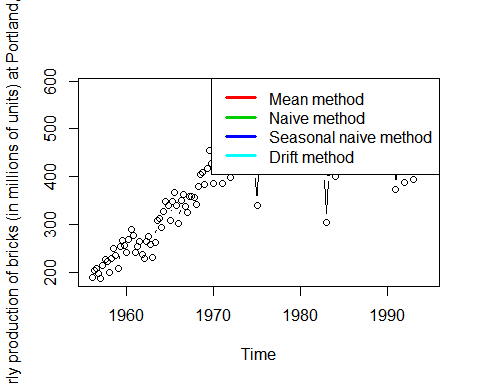
### Temperatures

plot.ts(window(x=ts.temp),type="b",xlab="Time",ylab="Temperatures", xlim=c(2014,2018))  
lines(meanf(window(ts.temp),h=24)$mean,col=2,lwd=3)  
lines(naive(window(ts.temp),h=24)$mean,col=3,lwd=3)  
lines(snaive(window(ts.temp),h=24)$mean,col=4,lwd=3)  
lines(rwf(window(ts.temp),drift=T,h=24)$mean,col=5,lwd=3)  
text<-c("Mean method","Naive method", "Seasonal naive method", "Drift method")  
legend("topright",text,col=2:5,lty=1,ncol=1,cex=1,lwd=3)



analysis

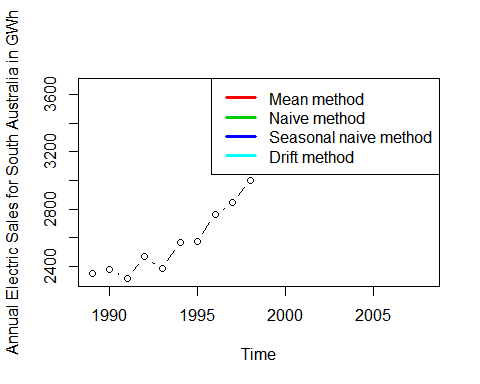
plot.ts(window(bricksq),type="b",xlab="Time",ylab="Quarterly production of bricks (in millions of units) at Portland, Australia")   
lines(meanf(window(bricksq),h=24)$mean,col=2,lwd=3)  
lines(naive(window(bricksq),h=24)$mean,col=3,lwd=3)  
lines(snaive(window(bricksq),h=24)$mean,col=4,lwd=3)  
lines(rwf(window(bricksq),drift=T,h=24)$mean,col=5,lwd=3)  
text<-c("Mean method","Naive method", "Seasonal naive method", "Drift method")  
legend("topright",text,col=2:5,lty=1,ncol=1,cex=1,lwd=3)



analysis

### Annual Electric Sales for South Australia in GWh

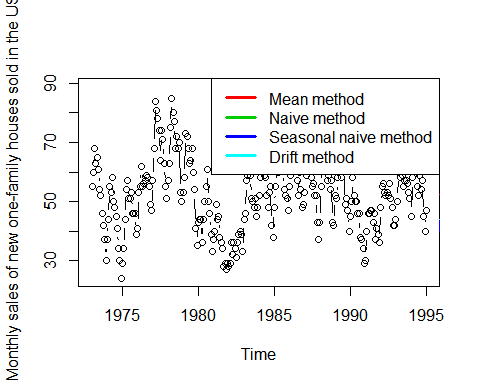
plot.ts(window(elecsales),type="b",xlab="Time",ylab="Annual Electric Sales for South Australia in GWh")  
lines(meanf(window(elecsales),h=24)$mean,col=2,lwd=3)  
lines(naive(window(elecsales),h=24)$mean,col=3,lwd=3)  
lines(snaive(window(elecsales),h=24)$mean,col=4,lwd=3)  
lines(rwf(window(elecsales),drift=T,h=24)$mean,col=5,lwd=3)  
text<-c("Mean method","Naive method", "Seasonal naive method", "Drift method")  
legend("topright",text,col=2:5,lty=1,ncol=1,cex=1,lwd=3)



analysis

### Monthly sales of new one-family houses sold in the USA

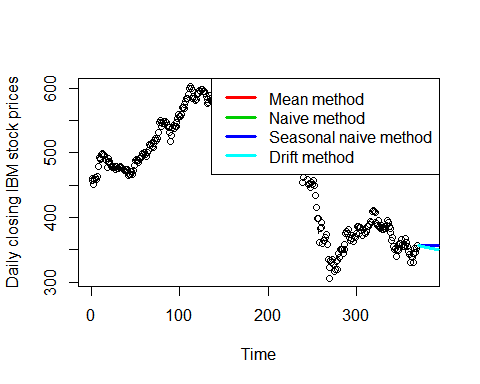
plot.ts(window(hsales, start=1973, end=1995),type="b",xlab="Time",ylab="Monthly sales of new one-family houses sold in the USA") #time series plot of temp with points market as bubbles  
  
lines(meanf(window(hsales),h=24)$mean,col=2,lwd=3)  
lines(naive(window(hsales),h=24)$mean,col=3,lwd=3)  
lines(snaive(window(hsales),h=24)$mean,col=4,lwd=3)  
lines(rwf(window(hsales),drift=T,h=24)$mean,col=5,lwd=3)  
text<-c("Mean method","Naive method", "Seasonal naive method", "Drift method")  
legend("topright",text,col=2:5,lty=1,ncol=1,cex=1,lwd=3)



analysis

Daily closing IBM stock prices

plot.ts(window(ibmclose),type="b",xlab="Time",ylab="Daily closing IBM stock prices", xlim=c(1, 379)) #time series plot of temp with points market as bubbles  
  
lines(meanf(window(ibmclose),h=24)$mean,col=2,lwd=3)  
lines(naive(window(ibmclose),h=24)$mean,col=3,lwd=3)  
lines(snaive(window(ibmclose),h=24)$mean,col=4,lwd=3)  
lines(rwf(window(ibmclose),drift=T,h=24)$mean,col=5,lwd=3)  
text<-c("Mean method","Naive method", "Seasonal naive method", "Drift method")  
legend("topright",text,col=2:5,lty=1,ncol=1,cex=1,lwd=3)



analysis

@Manual{R-base, title = {Using R for Time Series Analysis}, author = {{Avril Coghlan}}, url = {<https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html#time-series-analysis>}, }

@Manual{R-base, title = {Forecasting: Principles and Practice}, author = {{Rob J Hyndman, George Athanasopoulos}}, organization = {Monash University}, address = {Austria}, year = {2016}, url = {<https://otexts.com/fpp2/>}, }