Time Series Analysis
DSC534

Dr Yiolanda Englezou

Fall semester 2025-2026

Notes based on notes of Prof. Konstantinos Fokianos

Lecture 5

Visualization

Most important method for visualizing time series is a time series plot, data are plotted against time.

Generic plot(.) function (gaps might occur in the plot when we have missing values)

Aspect ratio might become an issue (there are no any rules).

For long series you can split plots in different frames

Example (use maine.dat)

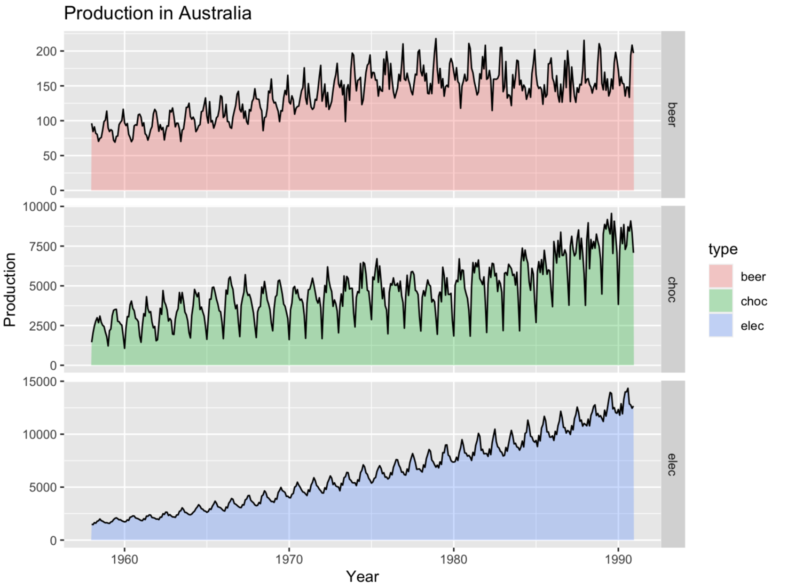
```
####read data
Maine.unemp <- read.csv("Maine.dat", sep="", header = T)
unemp <- Maine.unemp$unemp

####simple plot
ts.plot(unemp, ylab="(%)", main="Unemployment in Maine")

####another view
plot(unemp, type="o", pch=20, ylab="(%)", main="Unemployment in Maine")

###Horizontal line plot
plot(unemp, type="h", ylab="(%)", main="Unemployment in Maine")
```

For monthly economic data the series shows a **non-linear trend** and a **seasonal pattern** that increases with the level of series.
Maybe transform the data?
Worth studying further because unemployment is a basic economic indicator which is used for policy development.

Example (multiple time series plot) - cbe.dat

- Monthly supply of electricity (millions of kWh)
- Monthly supply of beer (millions of litres)
- Monthly chocolate production (tones)

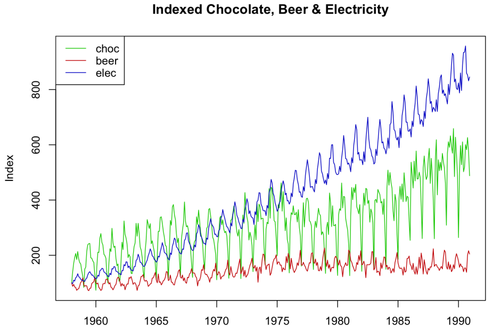
```
library(ggplot2)

####input data
dat <- read.table("cbe.dat",sep="", header=T)
cbe <- ts(dat, start=1958, freq=12)
#####simple plot of all series
plot(cbe, main="Chocolate, Beer & Electricity")

#####nicer plots
cbedf <- data.frame(t=rep(as.numeric(time(cbe)), times=3),
                    values=c(cbe[,1], cbe[,2], cbe[,3]),
                    type=rep(c("choc", "beer", "elec"), each=nrow(cbe)))
ggplot(cbedf, aes(x=t, values, fill=type)) +
  geom_area(alpha=0.3) + geom_line() +
  facet_grid(type~., scales="free") +
  ggtitle("Production in Australia") +
  xlab("Year") + ylab("Production")
```



All series show a distinct **seasonal pattern** along with a **trend**.
It is useful to know that the population of Australia has increased by 1.8 for this time period.

We can plot in a single frame after standardisation.

```
####First standardizing plot
## Indexing the series by standardizing with the first observation
tsd <- cbe
tsd[,1] <- tsd[,1]/tsd[1,1]*100
tsd[,2] <- tsd[,2]/tsd[1,2]*100
tsd[,3] <- tsd[,3]/tsd[1,3]*100
## Plotting in one single frame
clr <- c("green3", "red3", "blue3")
plot(tsd, plot.type="single", ylab="Index", col=clr)
title("Indexed Chocolate, Beer & Electricity")
## Legend
ltxt <- names(dat)
legend("topleft", lty=1, col=clr, legend=ltxt)
```



Suppose that $X_t^{(1)}$ is the electricity time series

$X_t^{(2)}$ beer time series

$X_t^{(3)}$ chocolate time series

We plot $X_t^{(1)}/X_1^{(1)}$, $X_t^{(2)}/X_1^{(2)}$, $X_t^{(3)}/X_1^{(3)}$ to obtain

visually differences with respect to the first observation of each series.

Note that electricity production increased around 8 times

chocolate " " " 4 times

beer " " " 2 times

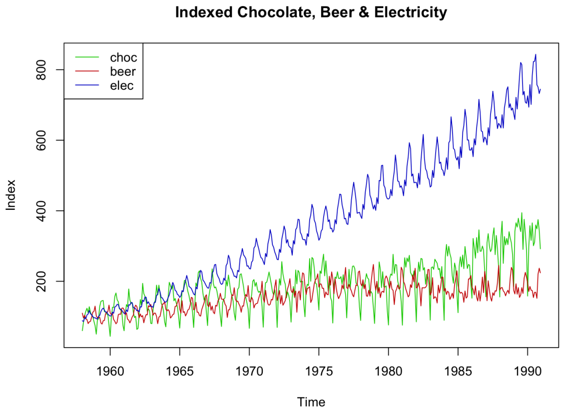
Seasonality is more visible for chocolate followed by electricity and beer.

We use a different strategy because of seasonality.

We consider for standardization the first 12 monts as a reference period.

for instance, $X_t^{(1)} \Big/ \left(\sum_{t=1}^{12} X_t^{(1)}/12 \right)$ and similarly for other series.

```
## Indexing the series vs. the first period
tsd <- cbe
tsd[,1] <- tsd[,1]/mean(tsd[1:12,1])*100
tsd[,2] <- tsd[,2]/mean(tsd[1:12,2])*100
tsd[,3] <- tsd[,3]/mean(tsd[1:12,3])*100

## Plotting in one single frame
plot(tsd, plot.type="single", ylab="Index", col=clr)
title("Indexed Chocolate, Beer & Electricity")
## Legend
ltxt <- names(dat)
legend("topleft", lty=1, col=clr, legend=ltxt)
```



We observe that the evolution of beer and chocolate production changes.

Graphical displays are important!!

1.) Linear Transformations

$$Y_t = a + b X_t$$

Such transformations do not affect autocorrelation, model forecasts so we can easily use it.

Suppose that (X_t) is stationary, $E(X_t)=0$, $\gamma_X(h) = Cov(X_t, X_{t+h})$

$$\rho_X(h) = \gamma_X(h) / \gamma_X(0)$$

$$E(Y_t) = a + b\, E(X_t) = a$$

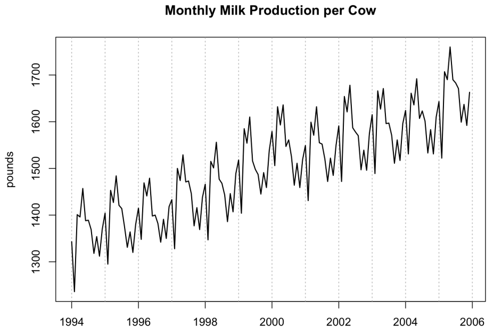
$$\gamma_Y(h) = Cov(Y_t, Y_{t+h}) = Cov(a + bX_t, a + bX_{t+h}) = b^2 \gamma_X(h)$$

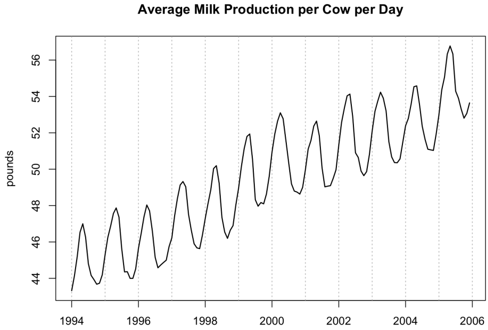
$$\rho_Y(h) = \frac{\gamma_Y(h)}{\gamma_Y(0)} = \frac{\cancel{b^2}\gamma_X(h)}{\cancel{b^2}\gamma_X(0)} = \rho_X(h)$$

2.) Monthly Sums and Averages

Instead of considering monthly data
we can conider daily averages / month.

```
###required packages
library(TSA)
library(forecast)

###data belongs to TSA package
data("milk")
## Monthly totals
plot(milk, xlab="Year", ylab="pounds", main="Monthly Milk Production per Cow")
abline(v=1994:2006, col="grey", lty=3)
lines(milk, lwd=1.5)

##The maonthdays() function
## Monthly average per day
milk.adj <- milk/monthdays(milk)
plot(milk.adj,xlab="Year",ylab="pounds",
main="Average Milk Production per Cow per Day")
abline(v=1994:2006, col="grey", lty=3)
lines(milk.adj, lwd=1.5)
```


Monthly Milk Production per Cow
pounds
1300
1400
1500
1600
1700
1994
1996
1998
2000
2002
2004
2006


Average Milk Production per Cow per Day
pounds
44
46
48
50
52
54
56
1994
1996
1998
2000
2002
2004
2006

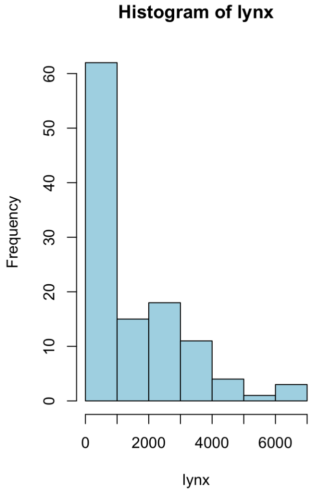
3) Box-Cox transformation

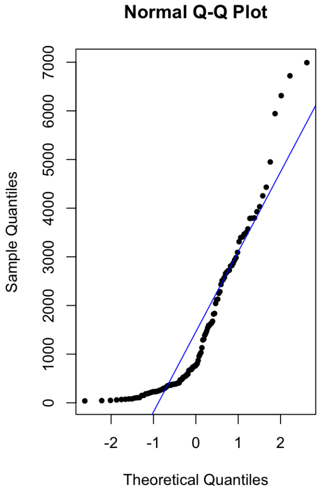
$$g_\lambda(x) = \begin{cases} \dfrac{x^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log x, & \lambda = 0 \end{cases}$$

We use these transformations to stabilize variance, reduce skewness

Inference is made on the transformed scale so we need to be cautious.

```
###Log Transformation
par(mfrow=c(1,2))
hist(lynx, col="lightblue")
qqnorm(lynx, pch=20); qqline(lynx, col="blue")
```





```
ts.plot(lynx, main="Lynx Trappings")
ts.plot(log(lynx), main="Logged Lynx Trappings")
```