INSEAD The Business School for the World®

DS(ML)B:
Data Science (& Machine Learning) for Business

Profs. Anton Ovchinnikov, Theos Evgeniou, Spyros Zoumpoulis

Sessions 03-04

- Time Series Models

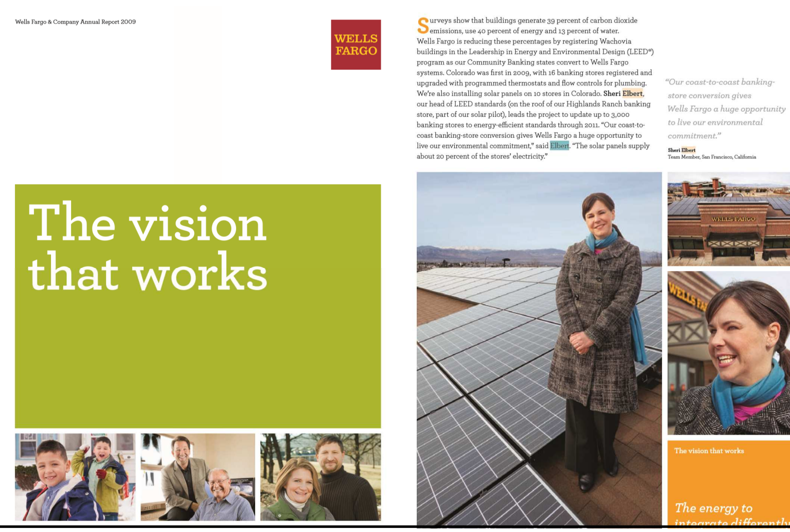
Plan for the day – Learning objectives

INSEAD

1. Conceptual introduction to Time Series modeling

2. Methods for Time Series modeling, and their R implementations

3. Application: predicting future electricity rates to evaluate the NPV of a solar power system
   - Wells Fargo case
   - Project featured in annual report


Wells Fargo & Company Annual Report 2009

WELLS FARGO

The vision that works

Surveys show that buildings generate 39 percent of carbon dioxide emissions, use 40 percent of energy and 13 percent of water. Wells Fargo is reducing these percentages by registering Wachovia buildings in the Leadership in Energy and Environmental Design (LEED®) program as our Community Banking states convert to Wells Fargo systems. Colorado was first in 2009, with 16 banking stores registered and upgraded with programmed thermostats and flow controls for plumbing. We're also installing solar panels on 10 stores in Colorado. **Sheri Elbert**, our head of LEED standards (on the roof of our Highlands Ranch banking store, part of our solar pilot), leads the project to update up to 3,000 banking stores to energy-efficient standards through 2011. "Our coast-to-coast banking-store conversion gives Wells Fargo a huge opportunity to live our environmental commitment," said Elbert. "The solar panels supply about 20 percent of the stores' electricity."

"Our coast-to-coast banking-store conversion gives Wells Fargo a huge opportunity to live our environmental commitment."

Sheri Elbert
Team Member, San Francisco, California

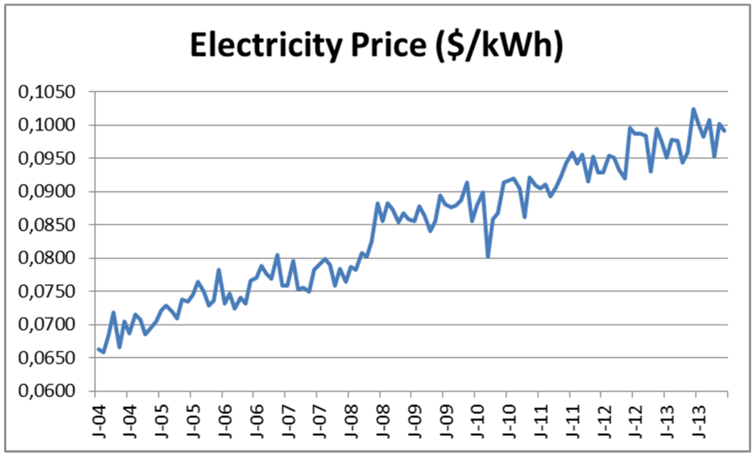
The vision that works

The energy to integrate differently

Sheri Lucas

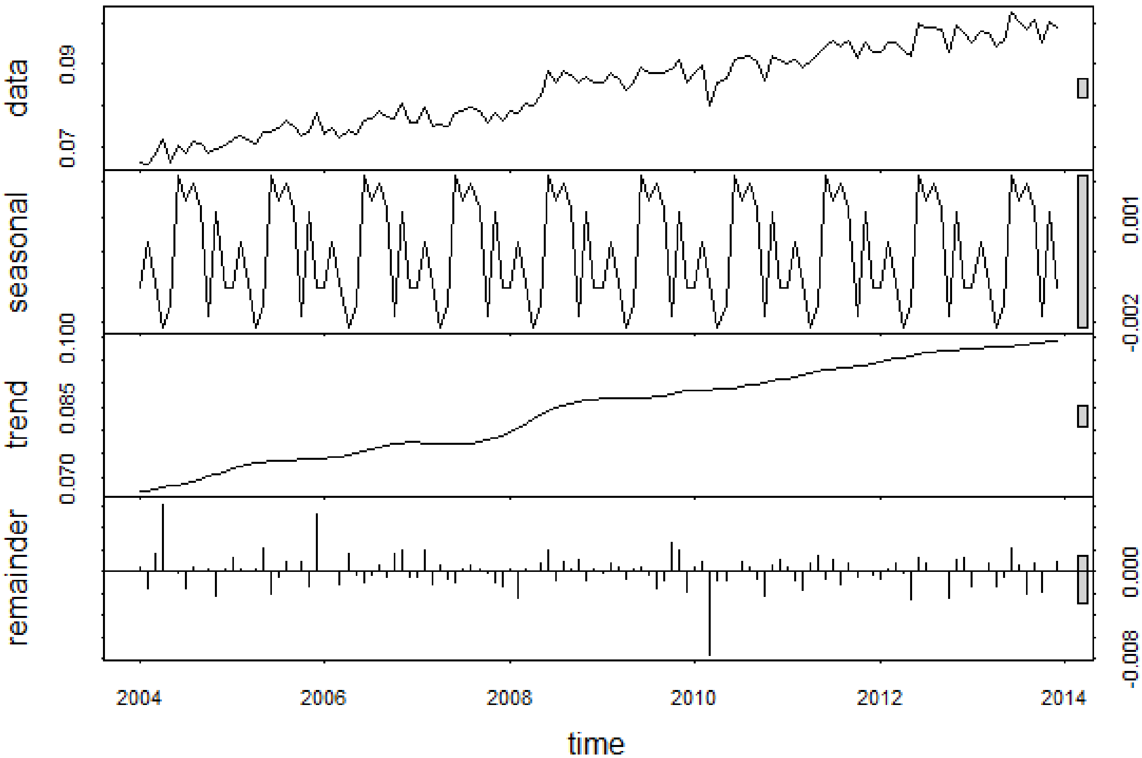
Timeseries

- **Definition**: a time series is a series of data points indexed (listed or graphed) in time order
- Example: monthly electricity prices in the state of California



- What do you "see" from/on the graph?
- BTW, why do we need any special methods for time series? Why will regression not be quite sufficient?

Understanding timeseries: level, noise, trend, seasonality



Level = value of the last datapoint = starting value before trend or seasonal adjustment are added

Seasonality = repetitive "short"–term pattern [seasonal indices vs smooth seasonality]

Trend = long-term movement of the data.
Not to confuse with **cycles**: up- or down-movements with irregular/unpredictable turning points

Noise / **error** = remaining / random variation in the data after accounting for trend and seasonality(ies)

All of the above can be **Additive or Multiplicative:**
e.g., $\$y_t + \$trend$ vs $\$y_t \times (1 + \%trend)$

Conceptual understnaing: noise, level, and trend

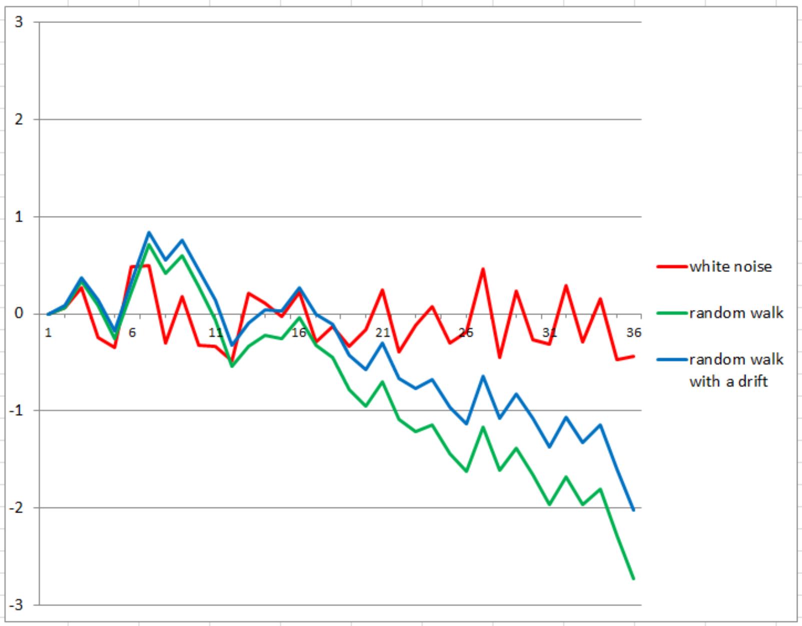
INSEAD

	A	B	C	D
1				0,02
2	month	white noi:	random w	random walk with a drift
3	1	0	0	0
4	2	0,063765	0,063765	0,083765
5	3	0,268283	0,332048	0,372048
6	4	-0,24494	0,087109	0,147109
7	5	-0,34402	-0,25692	-0,17692
8	6	0,480934	0,224019	0,324019
9	7	0,49198	0,715999	0,835999
10	8	-0,29916	0,416842	0,556842
11	9	0,179109	0,595951	0,755951
12	10	-0,3208	0,275148	0,455148
13	11	-0,33511	-0,05997	0,140035
14	12	-0,48094	-0,5409	-0,3209
15	13	0,208396	-0,33251	-0,09251
16	14	0,112326	-0,22018	0,03982
17	15	-0,03069	-0,25087	0,029125
18	16	0,216677	-0,0342	0,265802
19	17	-0,2879	-0,3221	-0,0021
20	18	-0,12668	-0,44877	-0,10877
21	19	-0,33757	-0,78634	-0,42634
22	20	-0,16417	-0,95051	-0,57051
23	21	0,247773	-0,70273	-0,30273
24	22	-0,38823	-1,09097	-0,67097
25	23	-0,12052	-1,21148	-0,77148
26	24	0,069519	-1,14197	-0,68197
27	25	-0,30347	-1,44543	-0,96543
28	26	-0,18409	-1,62952	-1,12952
29	27	0,466844	-1,16268	-0,64268
30	28	-0,45246	-1,61514	-1,07514
31	29	0,231228	-1,38391	-0,82391
32	30	-0,27339	-1,6573	-1,0773
33	31	-0,31254	-1,96983	-1,36983
34	32	0,285999	-1,68383	-1,06383
35	33	-0,28586	-1,9697	-1,3297
36	34	0,159024	-1,81067	-1,15067
37	35	[illegible]0,47562	[illegible]2,28629	[illegible]1,60629
38	36	-0,43901	-2,7253	-2,0253
39				

=RAND()-0,5

=C3+B4

=D3+B4+$D$1



What is your forecast for months 37? 38? 39?

File “0304 Time Series Examples….xls” on portal (press F9 to refresh)

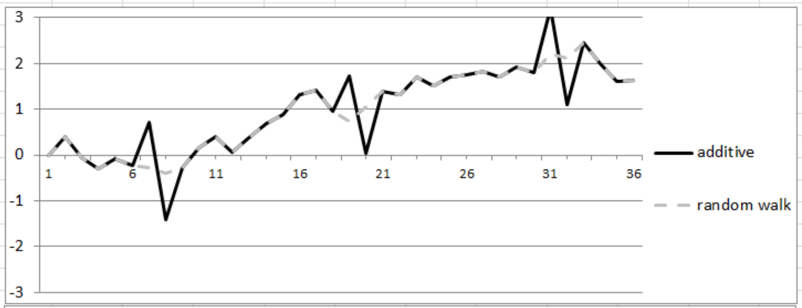
Conceptual understnaing: seasonality

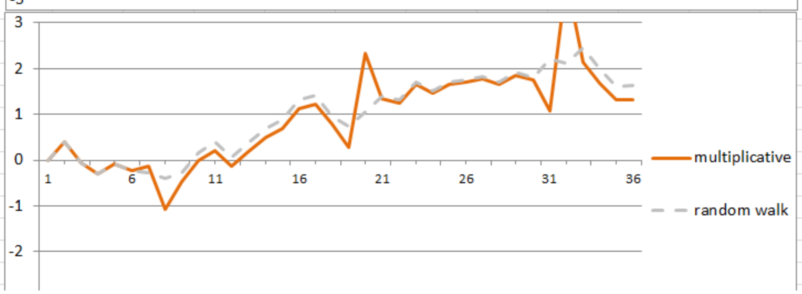
month	month inc	random w	additive (seasonality)	multiplicative
1	Jan	0	0	0
2	Feb	0,399443	0,399443	0,399443
3	March	-0,05734	-0,05734	-0,05734
4	April	-0,31371	-0,31371	-0,31371
5	May	-0,08643	-0,08643	-0,08643
6	June	-0,22046	-0,22046	-0,22046
7	July	-0,27772	0,722277	-0,13886
8	Aug	-0,40519	-1,40519	-1,06533
9	Sept	-0,28733	-0,28733	-0,47373
10	Oct	0,167582	0,167582	-0,01882
11	Nov	0,400684	0,400684	0,214284
12	Dec	0,056892	0,056892	-0,12951
13	Jan	0,37146	0,37146	0,185059
14	Feb	0,692372	0,692372	0,505972
15	March	0,875074	0,875074	0,688674
16	April	1,304348	1,304348	1,117947
17	May	1,414004	1,414004	1,227604
18	June	0,962493	0,962493	0,776093
19	July	0,727442	1,727442	0,270521
20	Aug	1,040654	0,040654	2,334931
21	Sept	1,395114	1,395114	1,344696
22	Oct	1,305113	1,305113	1,254695
23	Nov	1,705179	1,705179	1,654761
24	Dec	1,502029	1,502029	1,451611
25	Jan	1,694055	1,694055	1,643637
26	Feb	1,752033	1,752033	1,701615
27	March	1,819066	1,819066	1,768648
28	April	1,70069	1,70069	1,650272
29	May	1,909615	1,909615	1,859197
30	June	1,790179	1,790179	1,739761
31	July	2,19774	3,19774	1,073661
32	Aug	2,110298	1,110298	3,944878
33	Sept	2,445713	2,445713	2,140146
34	Oct	1,994083	1,994083	1,688517
35	Nov	1,610876	1,610876	1,30531
36	Dec	1,618388	1,618388	1,312822

=(F3+C4)*VLOOKUP(B4;$L$2:$N$13;3;FALSE)

=E3+C4+VLOOKUP(B4;$H$4:$I$15;2;FALSE)

month inc	additive	multiplicative
Jan	0	1
Feb	0	1
March	0	1
April	0	1
May	0	1
June	0	1
July	1	0,5
Aug	-2	4
Sept	1	0,5
Oct	0	1
Nov	0	1
Dec	0	1





Models for TimeSeries

[from the reading]

- Moving average
- Exponential smoothing
  - New Forecast (“level”) = α * Actual + (1 – α) * Old Forecast (“level”)
  - Holt’s model: Smoothing with [additive] trend: New Forecast = New Level + New Trend
    - New Level = α * Actual + (1 – α) * Old Forecast
    - New Trend = β * (New Level – Old Level) + (1 – β) * Old Trend
  - Winter’s model: Smoothing with [additive] trend and seasonality
- Multiplicative smoothing methods
- Decompositions: TBATS (trigonometric Fourier transforms)
- Auto-regressive methods
  - ARMA, ARIMA, etc. (ARCH, GARCH, etc. for variance)
- [Time-permitting/optional] any of the above with regressors (covariates / features), “dynamic regressions” + some cool use-cases

Timeseries modeling in R: your first example

INSEAD

Context: Wells Fargo (a very large bank) decides whether to install solar panels on the roof of its branchesand needs to obtain a 30year forecast of electricity prices:

- CSV datafile "0304 CSV data -- electric rates.csv" contains the data for monthly averages of prices over the last 10 years
- R script "0304 R code -- Timeseries I - decompositions ets and tbats.R" contains the code

- **Goal**: predict monthly prices for the next 30 years (360 values). Analytical complications: which model(s) to use?
  - We will first look at the exponential smoothing ("`ets`") and trigonometric decompositions ("`tbats`") models
  - Coding complications:
    - Neither `ets` nor `tbats` are part of the standard R installation; will need to install a package and call a library

File with code: "0304 R code -- Timeseries I - decompositions, ets and tbats.R"

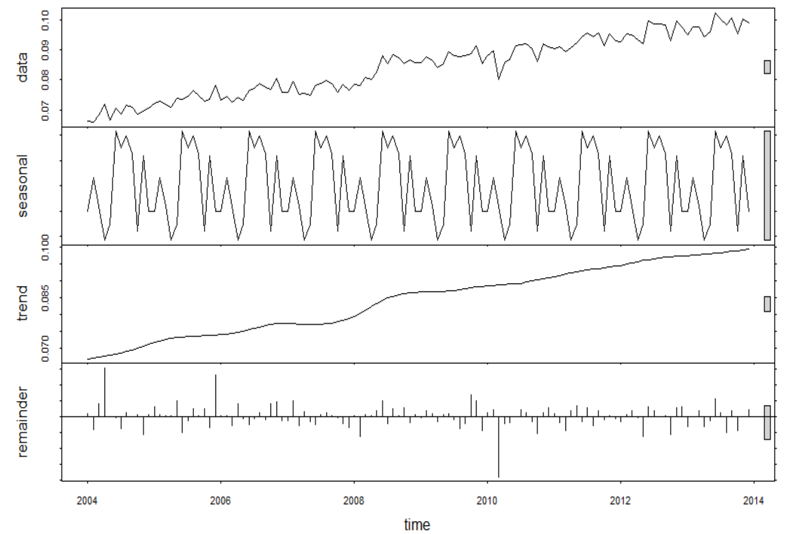
Load the data, define a timeseries

```
#install.packages("forecast") #-- do this only once

#Check the book: https://www.otexts.org/fpp2 and the blog:
http://robjhyndman.com/hyndsight

library("forecast")

ElectricPriceData<-read.csv(file.choose(), header=TRUE, sep=",")

ElectricPrice_ts <- ts(ElectricPriceData$ElectricRate,start=2004,
frequency=12) # ts function defines the dataset as timeseries starting Jan
2004 and having seasonality of frequency 12 (monthly)
```

Decomposition(s)

INSEAD

```
#plot various decompositions into error/noise,
trend and seasonality

fit <- decompose(ElectricPrice_ts,
type="multiplicative") #decompose using
"classical" method, multiplicative form

plot(fit)


fit <- decompose(ElectricPrice_ts,
type="additive") #decompose using "classical"
method, additive form

plot(fit)


fit <- stl(ElectricPrice_ts, t.window=12,
s.window="periodic", robust=TRUE) #decompose
using STL (Season and trend using Loess)

plot(fit)
```



File with code: "0304 R code -- Timeseries I - decompositions, ets and tbats.R"

Exponential smoothing: `ets` models

`ets` = "error-trend-seasonality"

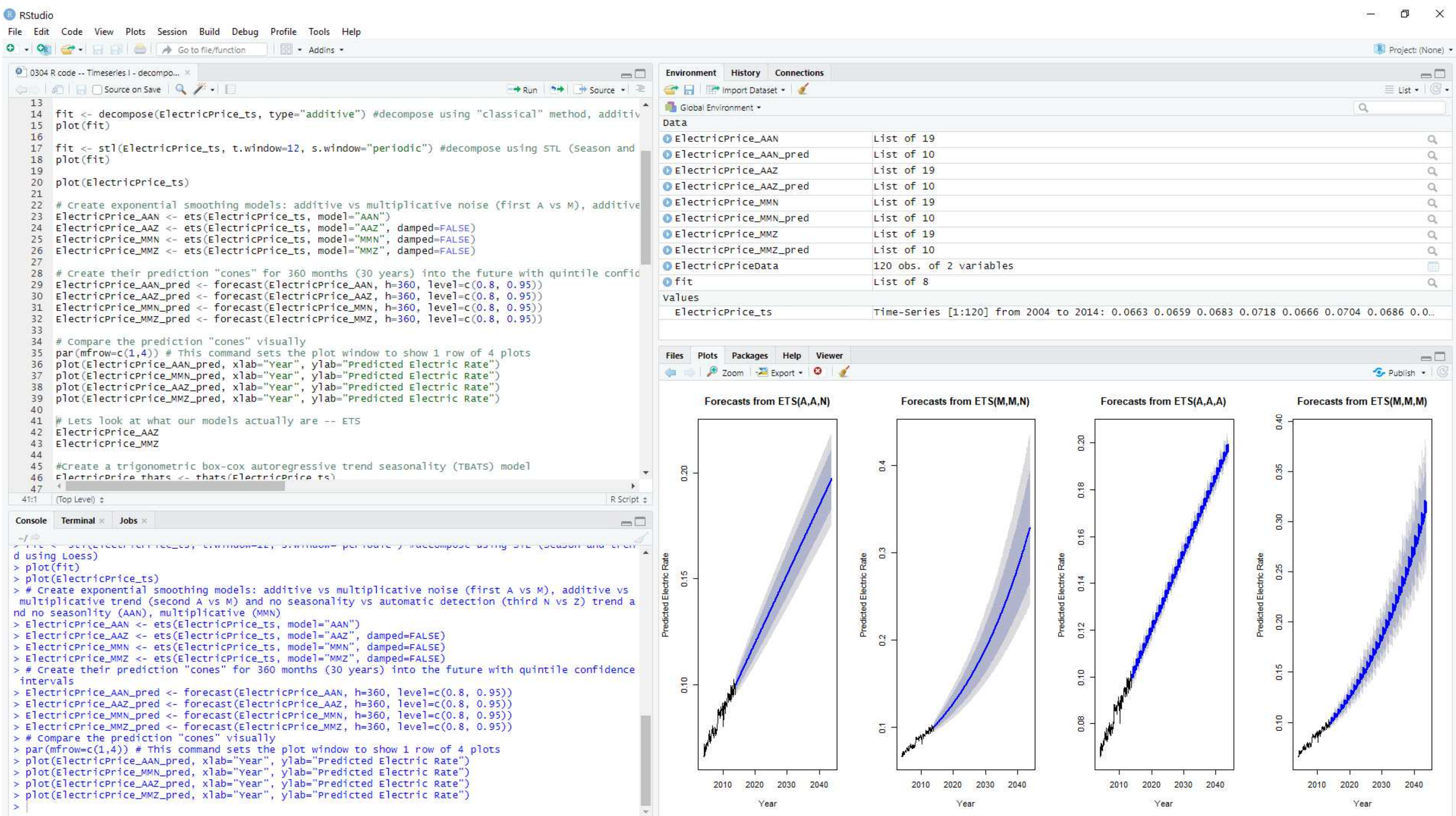
error ={A,M}

trend ={N,A,M} `Dampening=True` {Ad,Md}

seasonality ={N,A,M}

		Seasonal Component	
Trend	N	A	M
Component	(None)	(Additive)	(Multiplicative)
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)
M (Multiplicative)	(M,N)	(M,A)	(M,M)
M_d (Multiplicative damped)	(M_d,N)	(M_d,A)	(M_d,M)

Super-powerful `ets` models in R are VERY easy



File with code: "0304 R code -- Timeseries I - decompositions, ets and tbats.R"

`ets` models: taxonomy

Trend	Seasonal N	Seasonal A	Seasonal M
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t / \ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t / (\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t / (\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$
M	$\hat{y}_{t+h\|t} = \ell_t b_t^h$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t / (\ell_{t-1}b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{M_d}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h}$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}^{\phi}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t / (\ell_{t-1}b_{t-1}^{\phi})) + (1-\gamma)s_{t-m}$

ets models: reading output

```
> ElectricPrice_AAZ
ETS(A,A,A)
Call: ets(y = ElectricPrice_ts, model = "AAZ", damped = FALSE)
Smoothing parameters:
    alpha = 0.0583
    beta  = 5e-04
    gamma = 0.0311
Initial states:
    l = 0.0678
    b = 3e-04
    s=-0.0013 0.0015 -0.0011 0.0012 0.0025 9e-04 0.0022 -0.0019 -0.0016 -
0.0016 3e-04 -0.001
```

What does AAA mean? **A**dditive noise, **A**dditive trend, **A**dditive seasonality

What does a small alpha mean? Stable series, not much reaction to noise

beta~0, means no trend? No, beta=0 is a stable trend (little change)

What does small gamma mean? Stable seasonal indices

(N,N)	=	simple exponential smoothing
(A,N)	=	Holts linear method
(M,N)	=	Exponential trend method
(A_d,N)	=	additive damped trend method
(M_d,N)	=	multiplicative damped trend method
(A,A)	=	additive Holt-Winters method
(A,M)	=	multiplicative Holt-Winters method
(A_d,M)	=	Holt-Winters damped method

Method	Initial values
(N,N)	$\ell_0 = y_1$
(A,N) (A_d,N)	$\ell_0 = y_1, b_0 = y_2 - y_1$
(M,N) (M_d,N)	$\ell_0 = y_1, b_0 = y_2/y_1$
(A,A) (A_d,A)	$\ell_0 = \frac{1}{m}(y_1 + \cdots + y_m)$ $b_0 = \frac{1}{m}\left[\frac{y_{m+1}-y_1}{m} + \cdots + \frac{y_{m+m}-y_m}{m}\right]$ $s_0 = y_m - \ell_0,\ s_{-1} = y_{m-1} - \ell_0,\ \ldots,\ s_{-m+1} = y_1 - \ell_0$
(A,M) (A_d,M)	$\ell_0 = \frac{1}{m}(y_1 + \cdots + y_m)$ $b_0 = \frac{1}{m}\left[\frac{y_{m+1}-y_1}{m} + \cdots + \frac{y_{m+m}-y_m}{m}\right]$ $s_0 = y_m/\ell_0,\ s_{-1} = y_{m-1}/\ell_0,\ \ldots,\ s_{-m+1} = y_1/\ell_0$

File with code: "0304 R code -- Timeseries I - decompositions, ets and tbats.R"

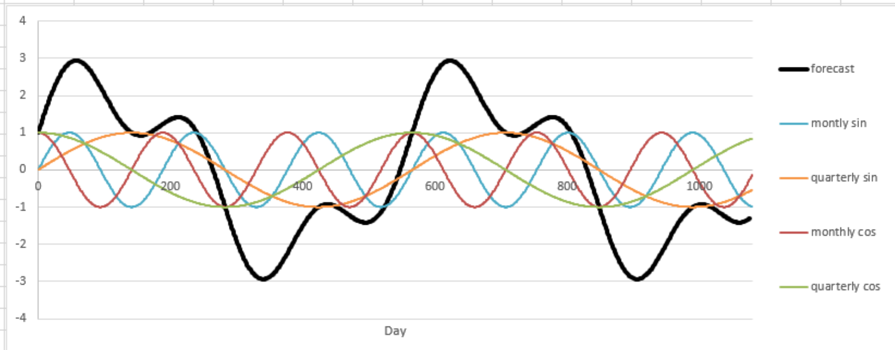
Trigonometric seasonality: sin() and cos() "waves"

=A4/30

=SIN(B4

=SUMPRODUCT($D$2:$G$2;D4:G4)

	A	B	C	D	E	F	G	H
1					weights			
2		Indicies		1	0	2	1	
3	Day	Monthly	Quarterly	montly sin	monthly cos	quarterly sin	quarterly cos	forecast
4	0	0	0	0	1	0	1	1
5	1	0,033333	0,011111	0,033327161	0,999444496	0,011110882	0,999938272	1,055487
6	2	0,066667	0,022222	0,066617295	0,997778601	0,022220393	0,999753097	1,110811
7	3	0,1	0,033333	0,099833417	0,995004165	0,033327161	0,999444496	1,165932
8	4	0,133333	0,044444	0,132938623	0,991124272	0,044429814	0,999012508	1,220811
9	5	0,166667	0,055556	0,165896133	0,986143232	0,055526982	0,998457187	1,275407
10	6	0,2	0,066667	0,198669331	0,980066578	0,066617295	0,997778601	1,329683
11	7	0,233333	0,077778	0,231221806	0,972901062	0,077699384	0,996976833	1,383597
12	8	0,266667	0,088889	0,263517391	0,964654645	0,08877188	0,996051983	1,437113
13	9	0,3	0,1	0,295520207	0,955336489	0,099833417	0,995004165	1,490191
14	10	0,333333	0,111111	0,327194697	0,944956946	0,110882629	0,993833509	1,542793
15	11	0,366667	0,122222	0,358505671	0,933527549	0,121918151	0,992540158	1,594882
16	12	0,4	0,133333	0,389418342	0,921060994	0,132938623	0,991124272	1,64642
17	13	0,433333	0,144444	0,419898367	0,907571133	0,143942682	0,989586027	1,69737
18	14	0,466667	0,155556	0,449911881	0,893072953	0,154928971	0,987925612	1,747695
19	15	0,5	0,166667	0,479425539	0,877582562	0,165896133	0,986143232	1,797361



By adding more "waves" of different periodicities (monthly, quarterly, semi-annually, etc.), changing their weights, and adding "shifted waves" (month$_{t-1}$ wave, etc.) you can create <u>very elaborate </u>seasonal patterns – "TBATS" model

File "0304 Time Series Examples….xls" on portal

TBATS model

(trigonometric box-cox autoregressive trend seasonality)

```
> ElectricPrice_tbats
TBATS(1, {0,0}, 1, {<12,5>})

Call: tbats(y = ElectricPrice_ts)

Parameters
  Alpha: 0.3119223
  Beta: 0.001059521
  Damping Parameter: 1
  Gamma-1 Values: -2.603098e-05
  Gamma-2 Values: 6.753875e-05

Seed States:
             [,1]
 [1,]  6.745943e-02
 [2,]  1.987776e-04
 [3,] -1.073112e-03
 [4,]  1.006799e-03
 [5,] -3.903458e-04
 [6,] -7.505053e-04
 [7,]  2.052910e-04
 [8,] -7.061269e-04
 [9,]  4.476049e-05
[10,]  1.308425e-04
[11,]  4.571218e-05
[12,]  9.594969e-04

Sigma: 0.001754992
AIC: -916.3708
>
>
>
>
>
>
>
>
>
>
```

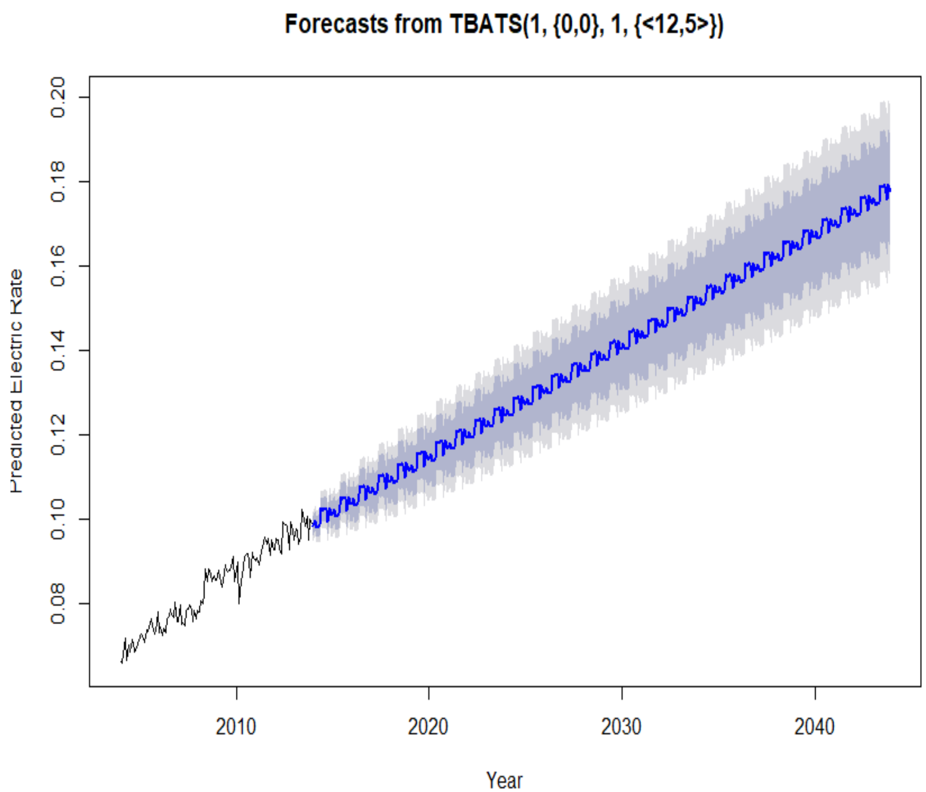
Understanding TBATS output

"1":
Box-cox transformation (removing outliners)
1=nothing removed

"{0,0}":
Autoregressive moving average:
{p,q} = numbers of values in ARMA

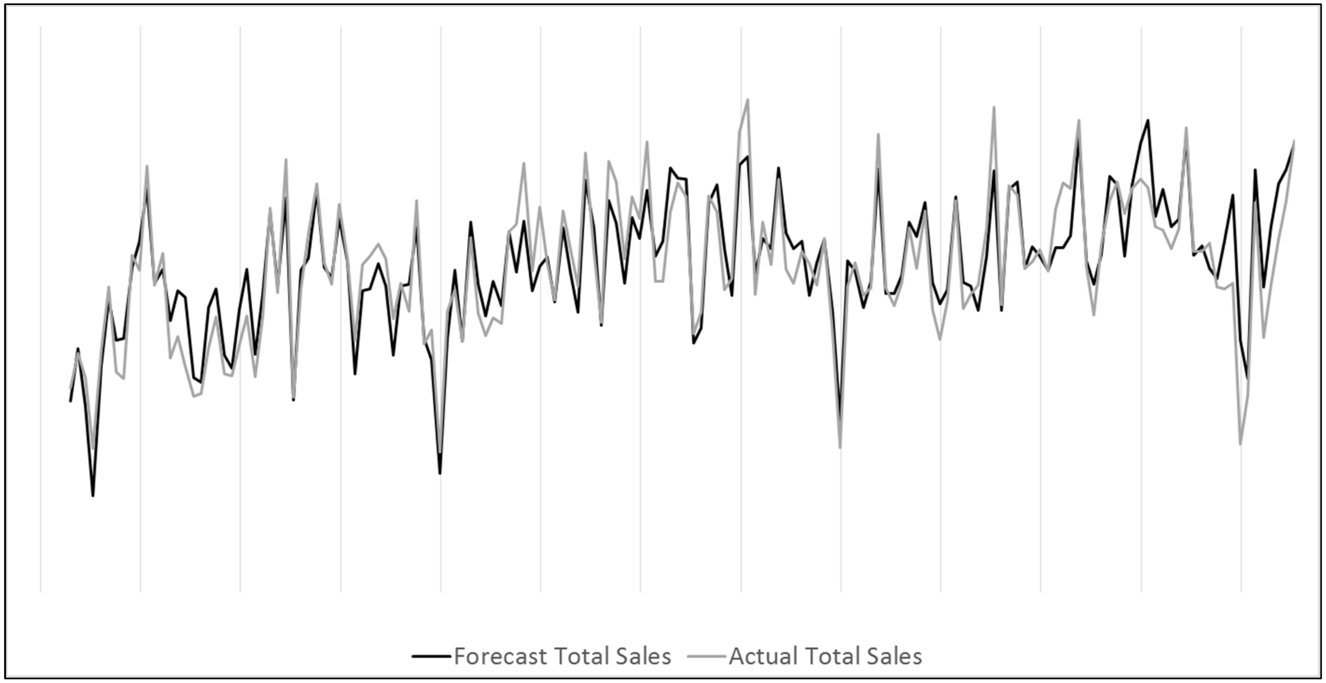
"1":
Dampening parameter (1=not dampened)

"(12,5)":
Number of seasonal periods, Number of trigonometric ("Fourier") terms for each seasonality



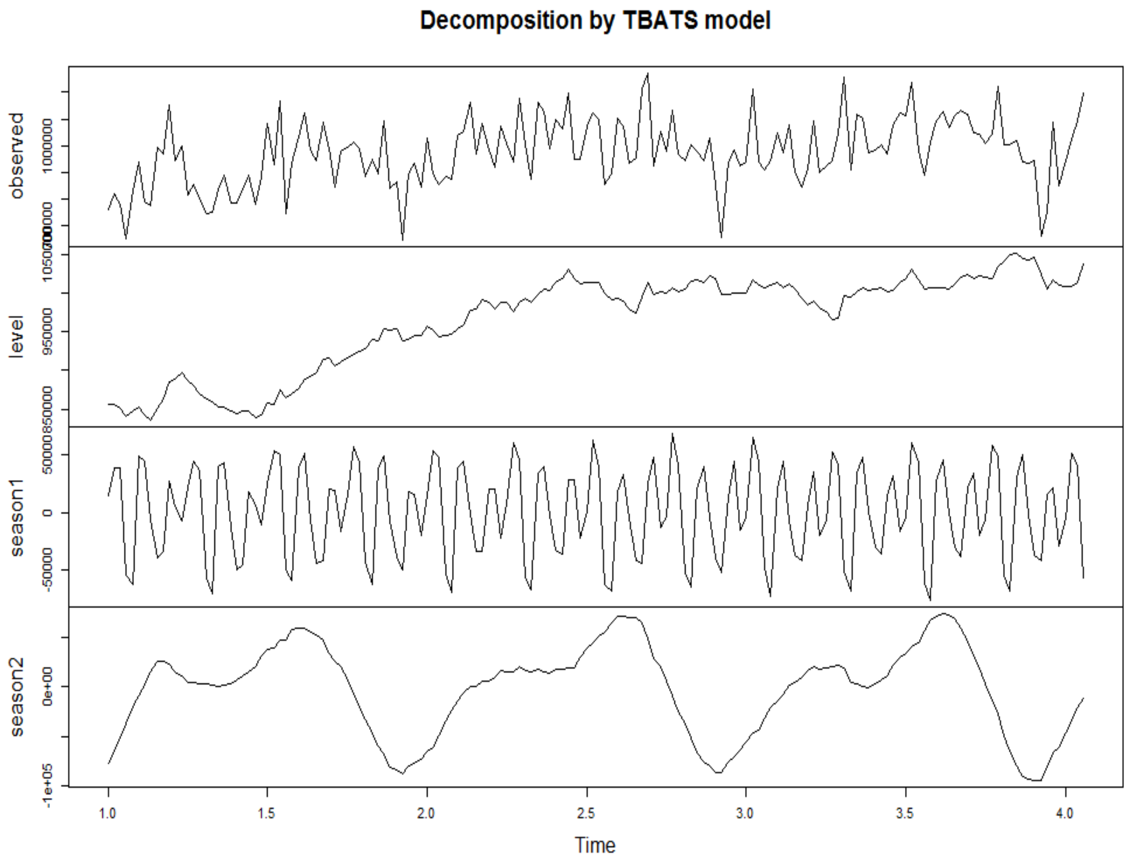
File with code: "0304 R code -- Timeseries I - decompositions, ets and tbats.R"

Multiple seasonalities:
Motivation: Forecasting Weekly Beer Sales



Weekly beverage sales forecasting over ~4yrs: TBATS w dummies/regressors, MAPE ~1%

Multiple seasonalities (`msts`) beyond this course ☹



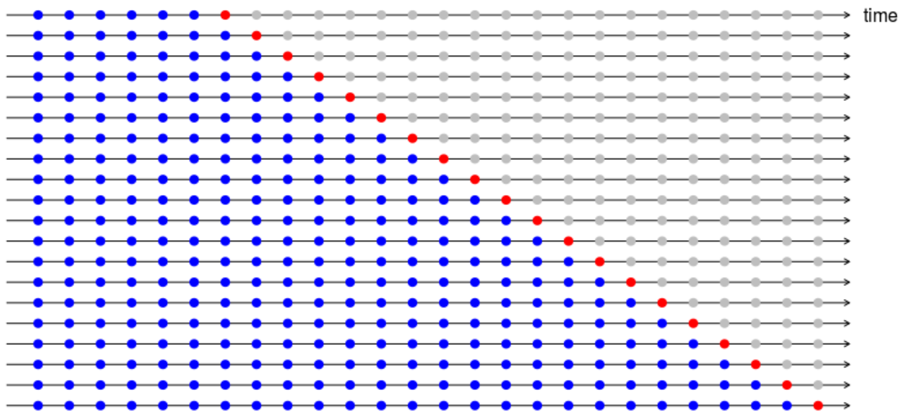
quarterly
(frequency=13)

yearly
(frequency=52)

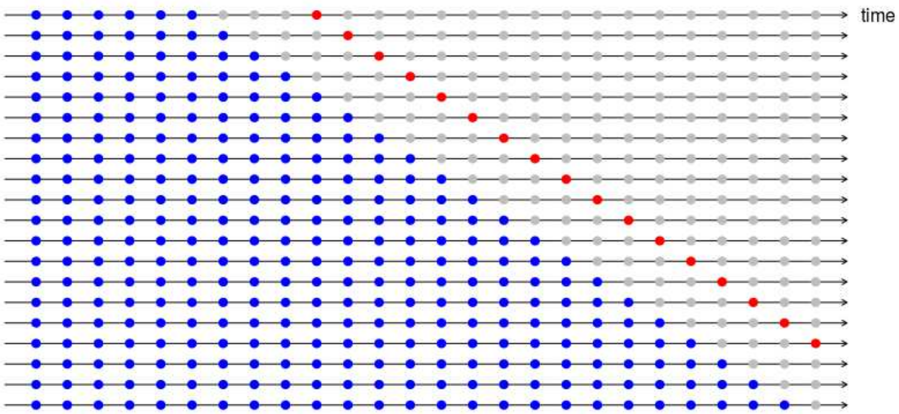
Weekly beverage sales forecasting over ~4yrs: TBATS w dummies/regressors, MAPE ~1%

Which model is better?
Timeseries “rolling horizon” cross-validation: `tsCV`

`tsCV(data, model function, h=1)`



one step ahead (h=1)

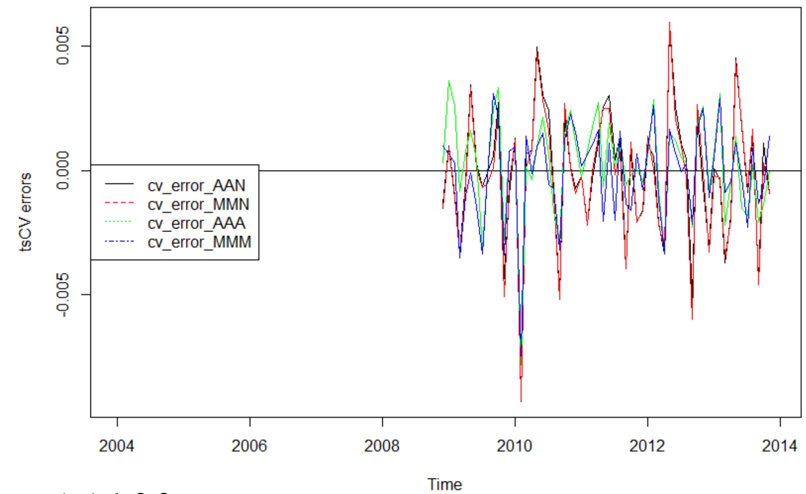


k-steps ahead (h=k)

https://robjhyndman.com/hyndsight/tscv/

Timeseries cross-validation: `tsCV` function

```
f_AAN  <- function(y, h) forecast(ets(y, model="AAN"), h = h)
error_AAN <- tsCV(ElectricPrice_ts, f_AAN, h=1, window=60)
…
plot(error_AAN, ylab='tsCV errors')
abline(0,0)
lines(error_MMN, col="red")
…
legend("left",legend=c("CV_errors"…
…

mean(abs(error_AAN/ElectricPrice_ts), na.rm=TRUE)*100
```




```
> mean(abs(e_AAN/ElectricPrice_ts), na.rm=TRUE)*100
[1] 2.178963
> mean(abs(e_MMN/ElectricPrice_ts), na.rm=TRUE)*100
[1] 2.124025
> mean(abs(e_AAA/ElectricPrice_ts), na.rm=TRUE)*100
[1] 1.629696
> mean(abs(e_MMM/ElectricPrice_ts), na.rm=TRUE)*100
[1] 1.600519
```

MMM model is most accurate

File with code: "0304 R code -- Timeseries I - decompositions, ets and tbats.R"

ARIMA: auto-regression, moving average and "differencing"

INSEAD

Main idea: feature-engineer "X" variables out of the Y variable. What kinds of X variables?

- Auto-regressive Xs:
  - $y_t = a + b^*y_{t-1} + error_t$

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t,$$

- Moving average Xs:
  - $y_t = a + b^*error_{t-1} + error_t$

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q},$$

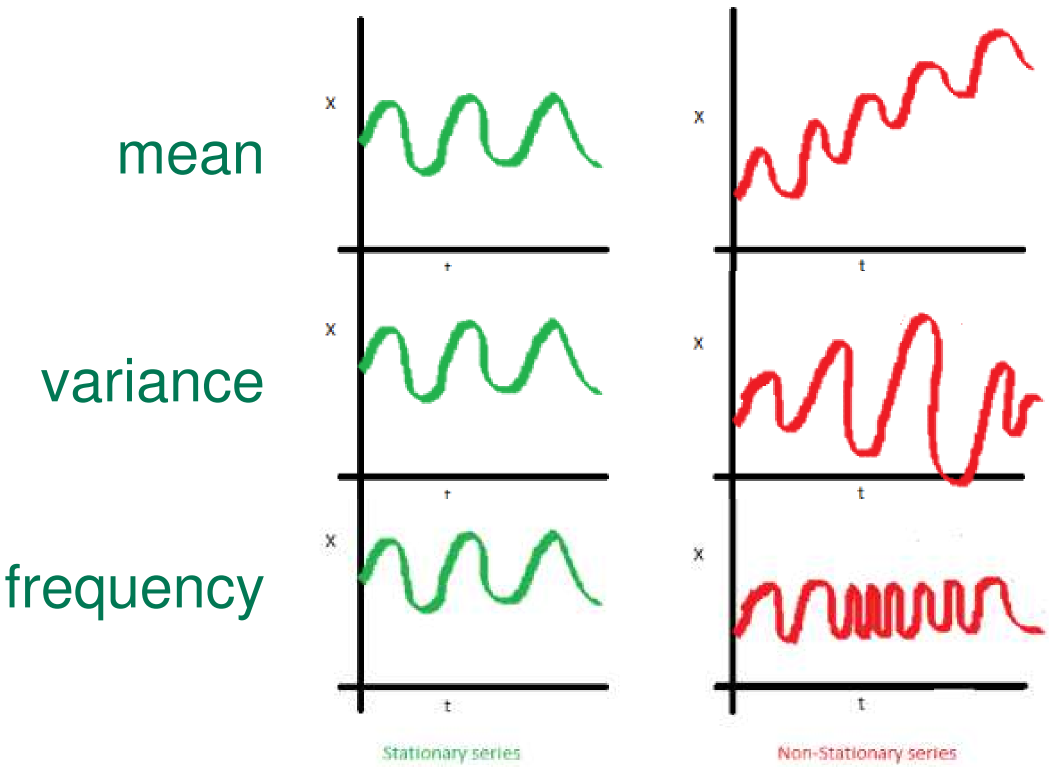
- Difference Xs and Y: create new variables by taking differences between consecutive observations ($z_t \equiv y'_t = y_t - y_{t-1}$)
- ARIMA: "**A**uto-**R**egressive **I**ntegrated **M**oving **A**verage"
  - ARIMA(p,d,q):
    - p = num of auto-regressive terms
    - d = "order" of first difference
    - q = num of moving average terms

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t,$$

- Stationarity
- Auto-Correlation Function (ACF), ACF plot
- Partial auto-correlation, PACF plot

White noise	ARIMA(0,0,0)
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregression	ARIMA(p,0,0)
Moving average	ARIMA(0,0,q)

Differencing and Stationarity



Tests to check stationarity:

- ADF (Augmented Dicky Fuller test)
- KPSS (Kwiatkowski–Phillips–Schmidt–Shin test)
- PP (Phillips-Perron test)

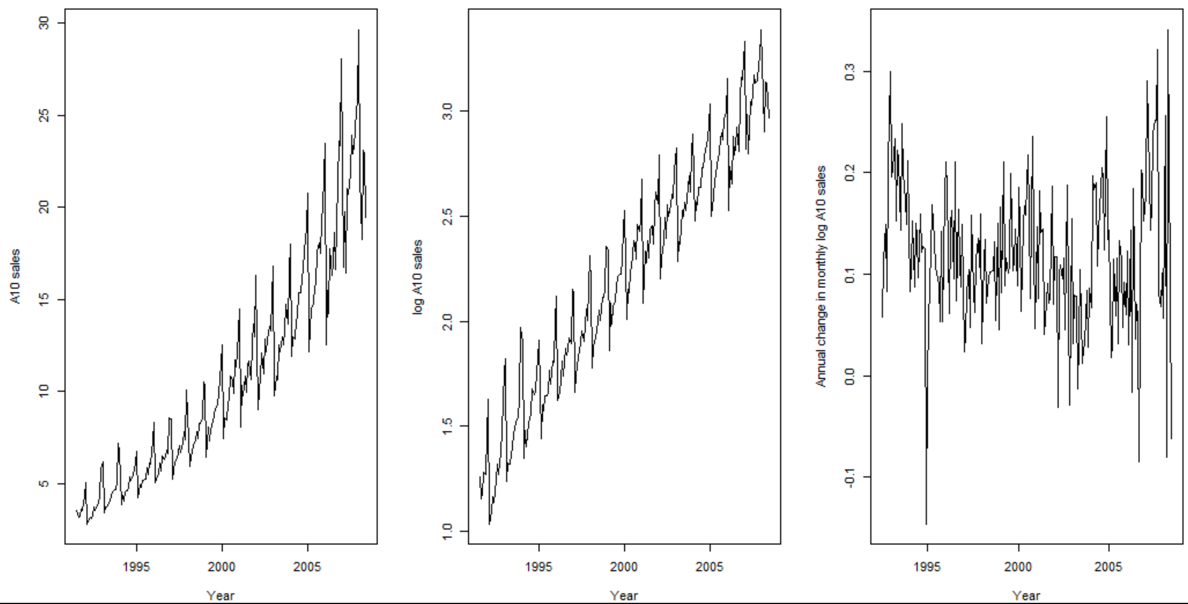
Related concept: unit root

`auto.arima` in `forecast` package tests and corrects automatically (via appropriate differencing)

Stationarity, Log-transform and Differencing: a10 dataset from FPP book

R script “0304 R code -- Timeseries II – ARIMAs and dynamic regressions.R” contains the code

```
#a10 dataset from fpp - sales of antidiabetic drug in Australia
par(mfrow=c(1,3))
plot(a10, xlab="Year", ylab="A10 sales")
plot(log(a10), xlab="Year", ylab="log A10 sales")
plot(diff(log(a10),12), xlab="Year", ylab="Annual change in monthly log A10 sales")
```

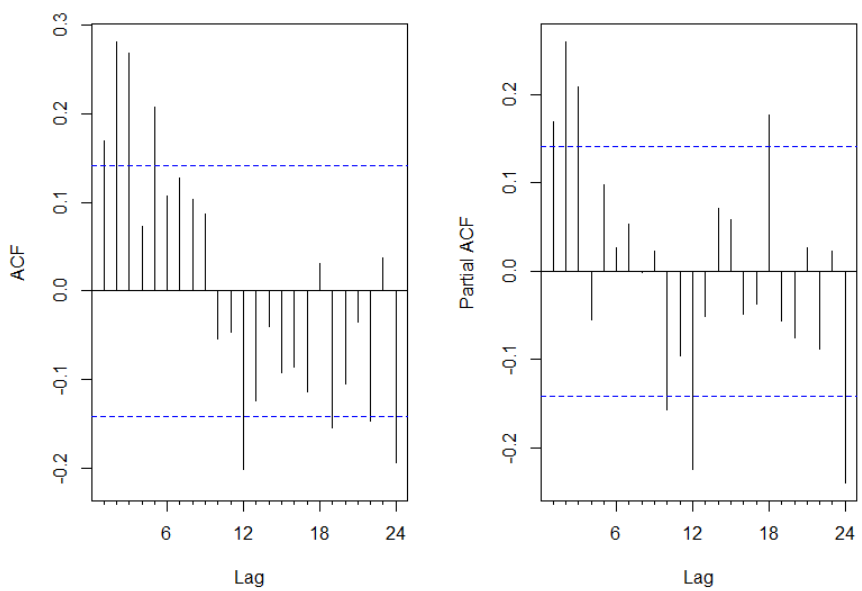


File with code: “0304 R code Timeseries II - ARIMAs and dynamic regressions.R”

Auto-correlation (ACF) and Partial Auto-corr. (PACF): a10 dataset from FPP book

R script "0304 R code -- Timeseries II – ARIMAs and dynamic regressions.R" contains the code

```
#a10 dataset from fpp - sales of antidiabetic drug in Australia
par(mfrow=c(1,2))
Acf(diff(log(a10),12),main="") # auto-correlation function
Pacf(diff(log(a10),12),main="") # partial auto-correlation function
```



File with code: "0304 R code Timeseries II - ARIMAs and dynamic regressions.R"

ARIMA [without seasonality]
a10 dataset from FPP book

INSEAD

```
> # non-seasonal first
> fit <- auto.arima(a10,seasonal=FALSE)
> fit
Series: a10
ARIMA(1,1,1) with drift

Coefficients:
         ar1      ma1   drift
       0.314  -0.9164  0.0965
s.e.   0.075   0.0262  0.0171

sigma^2 estimated as 3.626:  log likelihood=-417.92
AIC=843.84   AICc=844.04   BIC=857.09
```

(p,**1**,q) means that we are forecasting $z_t = y_t - y_{t-1}$

So the regression equation is:

- $z_{t+1} = const + 0.314\, z_t - 0.9164\, error_t + \; error_{t+1}$

Substituting:

- $y_{t+1} - y_t = const + 0.314\,(y_t - y_{t-1}) - 0.9164\, error_t + \; error_{t+1}$
- $y_{t+1} = const + (1 + 0.314)y_t - 0.314y_{t-1} \; - \; 0.9164\, error_t + \; error_{t+1}$, where $const$ "(intercept") = (1 - 0.314)*0.965

But future errors (residuals) are unknown, but on average $= \; N(0, \sigma^2) = N(0, 3.626)$, so:

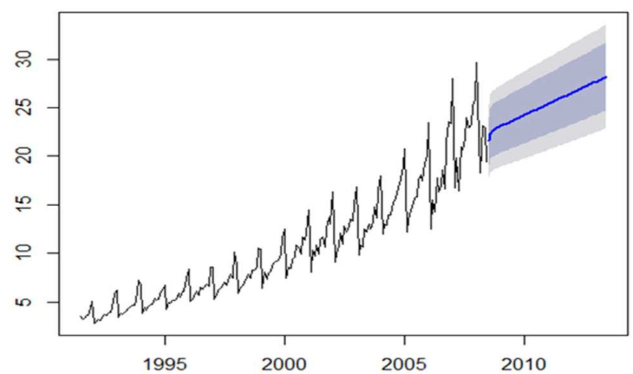
- $\boldsymbol{y_{t+1} = const + (1 + 0.314)y_t - 0.314y_{t-1} - 0.9164}\; error_t$
- $\boldsymbol{y_{t+2} = const + (1 + 0.314)y_{t+1} - 0.314y_t}$

ARIMA [without seasonality]
a10 dataset from FPP book

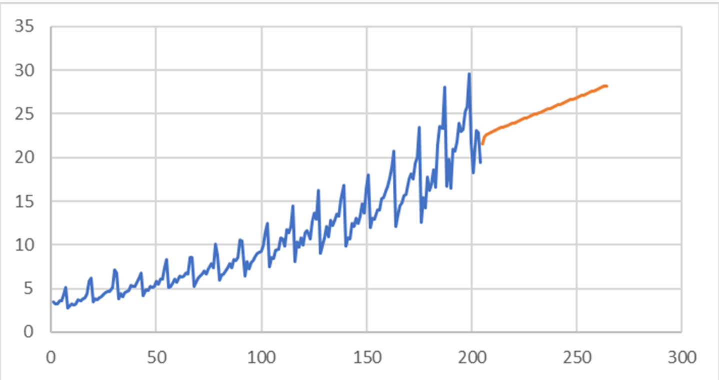
```
> # non-seasonal first
> fit <- auto.arima(a10,seasonal=FALSE)
> fit
Series: a10
ARIMA(1,1,1) with drift

Coefficients:
         ar1      ma1   drift
       0.314  -0.9164  0.0965
s.e.   0.075   0.0262  0.0171

sigma^2 estimated as 3.626:  log likelihood=-417.92
AIC=843.84   AICc=844.04   BIC=857.09
```

(p,**1**,q) means that we are forecasting $z_t = y_t - y_{t-1}$

Forecasts from ARIMA(1,1,1) with drift



$$y_{t+1} = (1 - 0.314) * 0.965 + (1 + 0.314)y_t - 0.314y_{t-1}$$

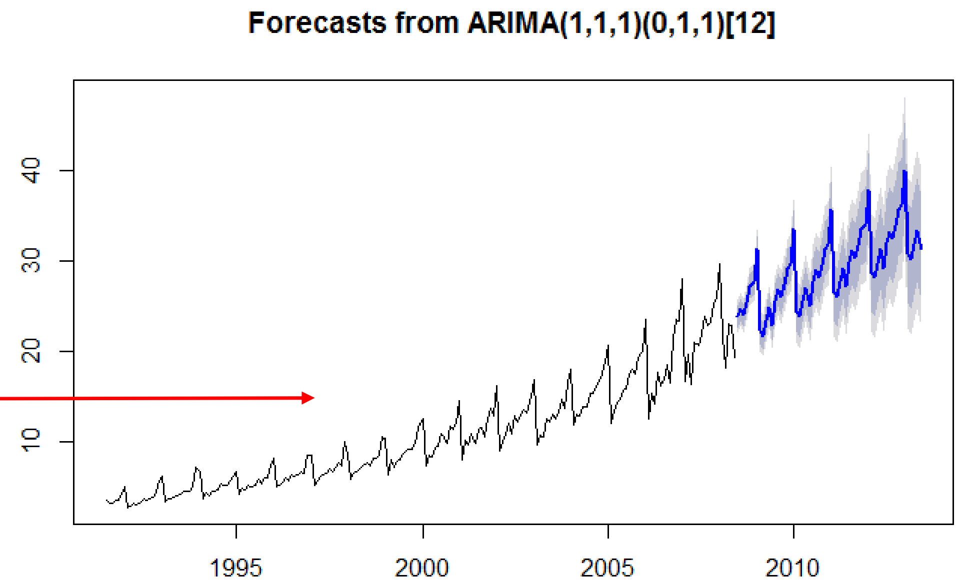


ARIMA with seasonality: a10 dataset from FPP book

- ARIMA(p,d,q) $\underbrace{(P,D,Q)}_{\text{seasonal part}}{}_{m}$ ← "m" is the number of periods in a season

```
> fit <- auto.arima(a10,seasonal=TRUE)
> fit
Series: a10
ARIMA(1,1,1)(0,1,1)[12]

Coefficients:
          ar1      ma1     sma1
      -0.2504  -0.6674  -0.4725
s.e.   0.1007   0.0870   0.0641

sigma^2 estimated as 0.8756:  log likelihood=-258.82
AIC=525.63   AICc=525.85   BIC=538.64
```

The model has just 3 "variables", but they are very elaborately constructed to capture a rich trend & seasonality pattern in the data



$$y_{t+1} = y_t + (y_{t-11} - y_{t-12}) - 0.2504(y_t - y_{t-1}) + 0.2504(y_{t-12} - y_{t-13}) - 0.6674\, error_t - 0.4725\, error_{t-11} + 0.6674 * 0.4725\, error_{t-12} + error_{t+1}$$

where $error = N(0, 0.8756)$

[Optional/Time Permitting]: Dynamic Regression: ts+regression

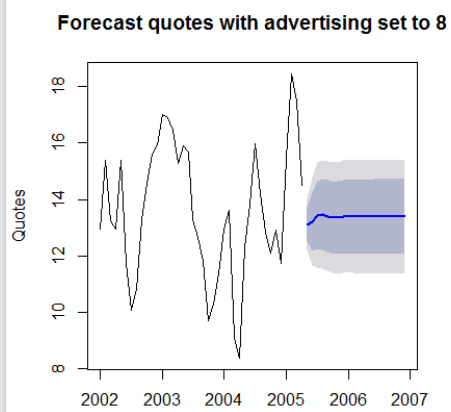
- In classes 0102 (Sarah's case) we considered predicting Y variable (price) by knowing some other X variables (weight, color, etc.)
  - Those X variables are sometimes called covariates, or features
- Today (so far) we considered predicting Y variable (eclectic price, consumption, etc.) by knowing how Y itself evolved in the past – timeseries analyses
- Can the two approaches be combined?
- Of course! This has two names:
  - "dynamic regression" or
  - "time series with regressors/covariates/features"

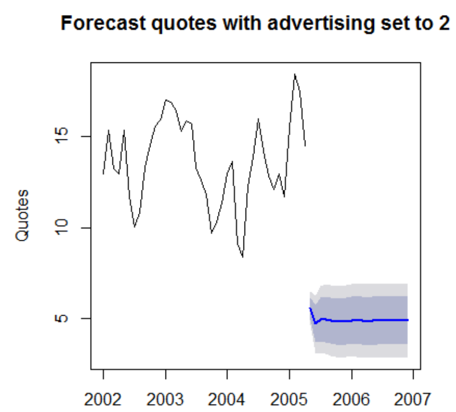
[Optional/Time Permitting]: Dynamic Regression example

- **Context**: we have data on the number of new insurance quotes and advertising in the previous period. Naturally expect two kinds of dependencies:
  - "direct effect": more advertising today = more new quotes today
  - "carryover effects":
    - more advertising yesterday = more new quotes today
    - more new quotes yesterday = more new quotes today

```
>
> Advert <- cbind(insurance[,2],
+                 c(NA,insurance[1:39,2]),
+                 c(NA,NA,insurance[1:38,2]),
+                 c(NA,NA,NA,insurance[1:37,2]))
> colnames(Advert) <- paste("AdLag",0:3,sep="")
>
>
> fit <- auto.arima(insurance[,1], xreg=Advert[,1:2], d=0)
> fit
Series: insurance[, 1]
Regression with ARIMA(3,0,0) errors

Coefficients:
         ar1      ar2     ar3  intercept  AdLag0  AdLag1
      1.4117  -0.9317  0.3591     2.0393  1.2564  0.1625
s.e.  0.1698   0.2545  0.1592     0.9931  0.0667  0.0591

sigma^2 estimated as 0.2165:  log likelihood=-23.89
AIC=61.78   AICc=65.28   BIC=73.6
>
>
> par(mfrow=c(1,2))
> fc8 <- forecast(fit, xreg=cbind(rep(8,20),c(Advert[40,1],rep(8,19))), h=20)
> plot(fc8, main="Forecast quotes with advertising set to 8", ylab="Quotes")
> fc2 <- forecast(fit, xreg=cbind(rep(2,20),c(Advert[40,1],rep(2,19))), h=20)
> plot(fc2, main="Forecast quotes with advertising set to 2", ylab="Quotes")
```

custom (written "by you") R function to create x variables with lags 2,3,4



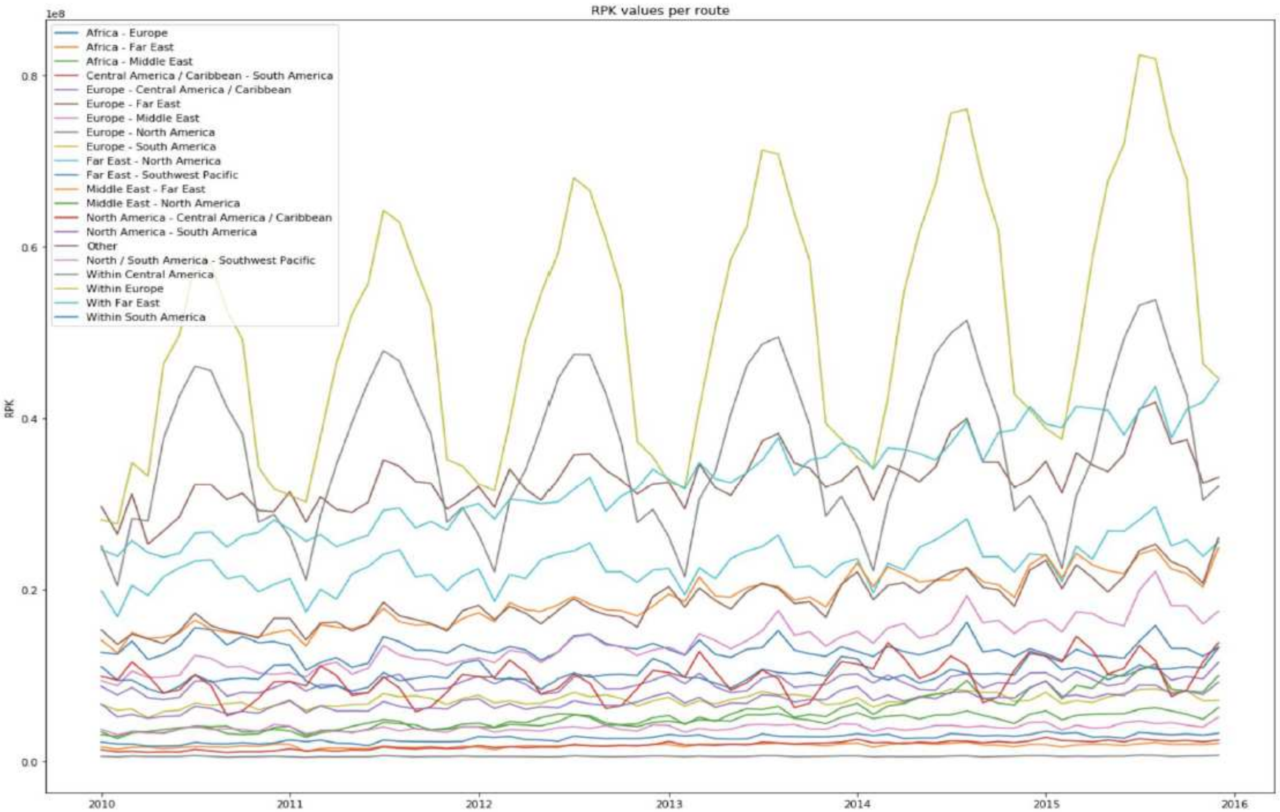


Regression equation: $q_{t+1} = 2.0393 + 1.41 * q_t - 0.93 q_{t-1} + 0.35 * q_{t-2} + 1.25 * ads_t + 0.16 * ads_{t-1} + error$

File with code: "0304 R Code -- Timeseries II - ARIMAs and dynamic regressions.R"

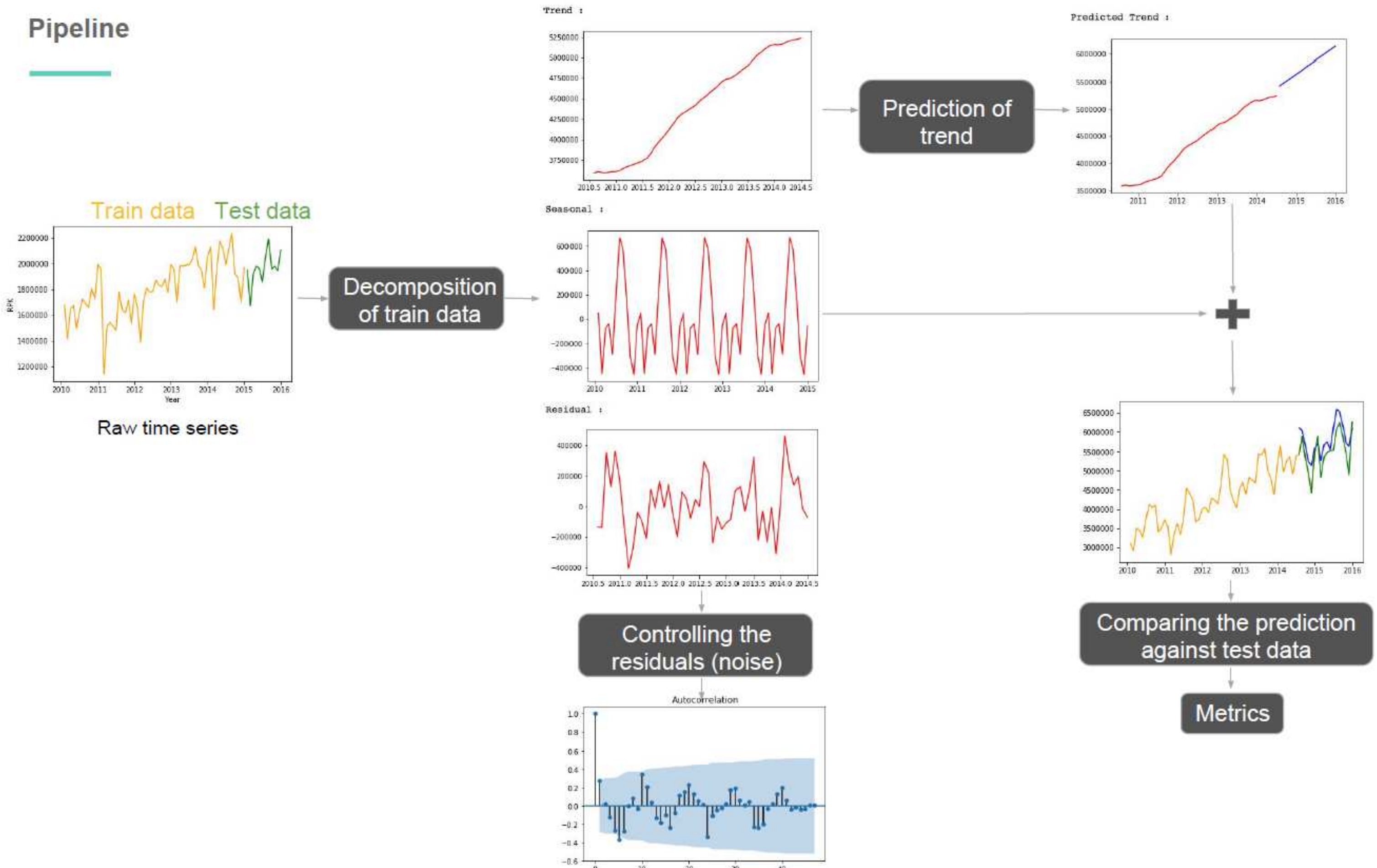
[Optional/Time Permitting]: IATA Case Study

Dataset



[Optional/Time Permitting]: IATA Case Study

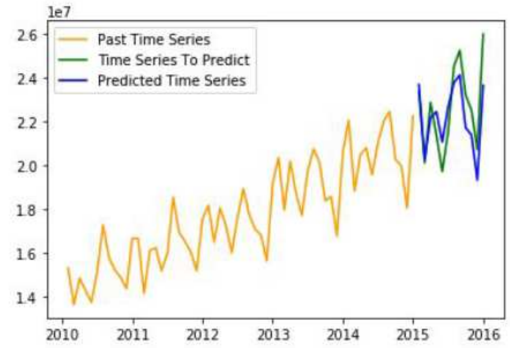
Pipeline



[Optional/Time Permitting]: IATA Case Study

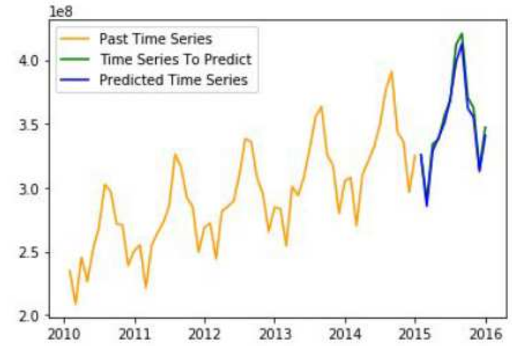
Results :

```
Prediction for route OTH:
```




```
MAE for route OTH : 1083617.04
MAPE for route OTH : 4.77%
```

```
Prediction for route IND:
```




```
MAE for route IND : 5316232.94
MAPE for route IND : 1.45%
```

Averaged metrics on all routes :

```
Global MAE : 685781.43
Global MAPE : 3.19%
```

[Optional/Time Permitting]: M4 Competition

INSEAD

- World's leading time series forecasting competition (100,000 time series dataset) https://www.m4.unic.ac.cy/
- Roots in "M1…" competitions; see "A brief history of time series forecasting competitions": https://robjhyndman.com/hyndsight/forecasting-competitions/
- Rob Hyndman (who is this?) and his team often come on top
- This year's winner, however: Uber Engineering with combined ETS * RNN ("recurrent neural network") model https://eng.uber.com/m4-forecasting-competition/

$$\widehat{y_{t+1}} = \hat{y}_t + \alpha(y_t - \widehat{y_t})$$

$$l_t = \alpha(y_t/s_t) + (1-\alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$
$$s_{t+m} = \gamma\frac{y_t}{(l_t+b_t)} + (1-\gamma)s_t$$

$$\widehat{y_{t+1}} = (l_t + b_t)s_t$$

Ets(*AM) ("Holt-Winters")

$$l_t = \alpha(y_t/s_t) + (1-\alpha)l_{t-1}$$
$$s_{t+m} = \gamma(y_t/l_t) + (1-\gamma)s_t$$

$$\widehat{y_{t+1..t+h}} = \underbrace{RNN(X_t)}_{\text{Feature-based part "transfer learning"}} * \underbrace{l_t * s_{t+1..t+h}}_{\text{Timeseries-based part}}$$

New Model

Summary of Sessions 3-4

- On many occasions data are indexed by time – timeseries data
- Such data requires special analytical tools, which explicitly account for the fact that prediction errors increase over time
- We discussed concepts and implementations of four families of models:
  - Exponential smoothing (`ets`)
  - Trigonometric decompositions (`tbats`)
  - Auto-regressive moving averages (ARIMA)
  - [opt/time-permitting] Dynamic regressions (we saw an example based on ARIMA, but the concept applies to any method)
- As with regression: through coding and R, very powerful time series analytics can be implemented in minutes
- Many resources online: e.g., FPP book https://www.otexts.org/fpp2

Next…

- Group Assignment 1: Yahoo's acquisition of Tumblr
  - Due by Sessions 5-6, upload to INSEAD portal

- Module II of the course: predicting events / "classification"
  - Sessions 5-6: metrics for classification and two main methods: logistic regression and CART
    - Reading on Logistic regression
    - STC case
  - Assignment 2 (predicting credit defaults)
  - Session 7-8: discussion of the assignment + new methods

INSEAD

The Business School for the World®

Europe | Asia | Middle East