NUS National University of Singapore | School of Computing

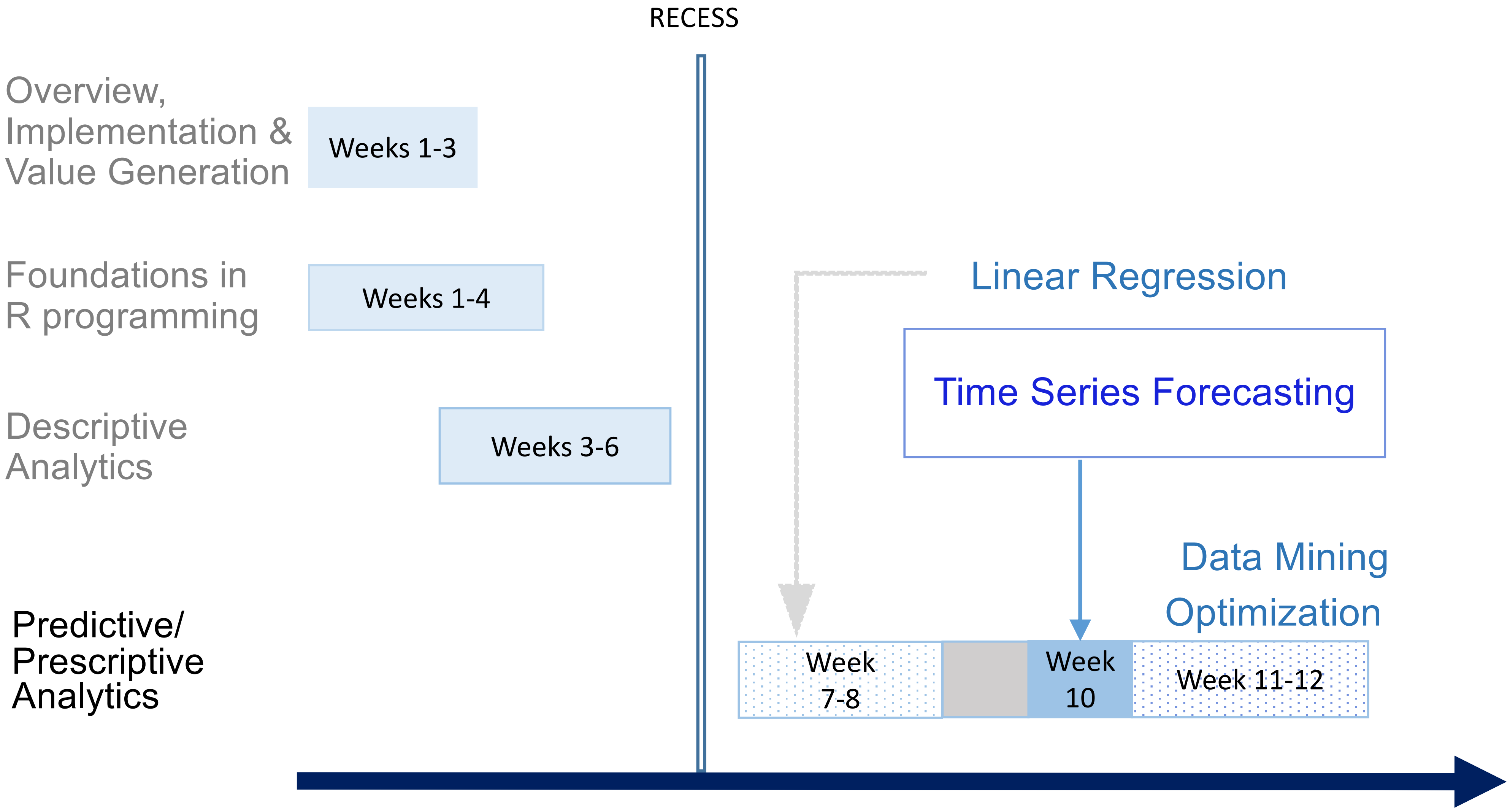
Leading The World With Asia's Best

Introduction to Business Analytics

Time Series Forecasting

Dr. Sharon Tan

Course Map

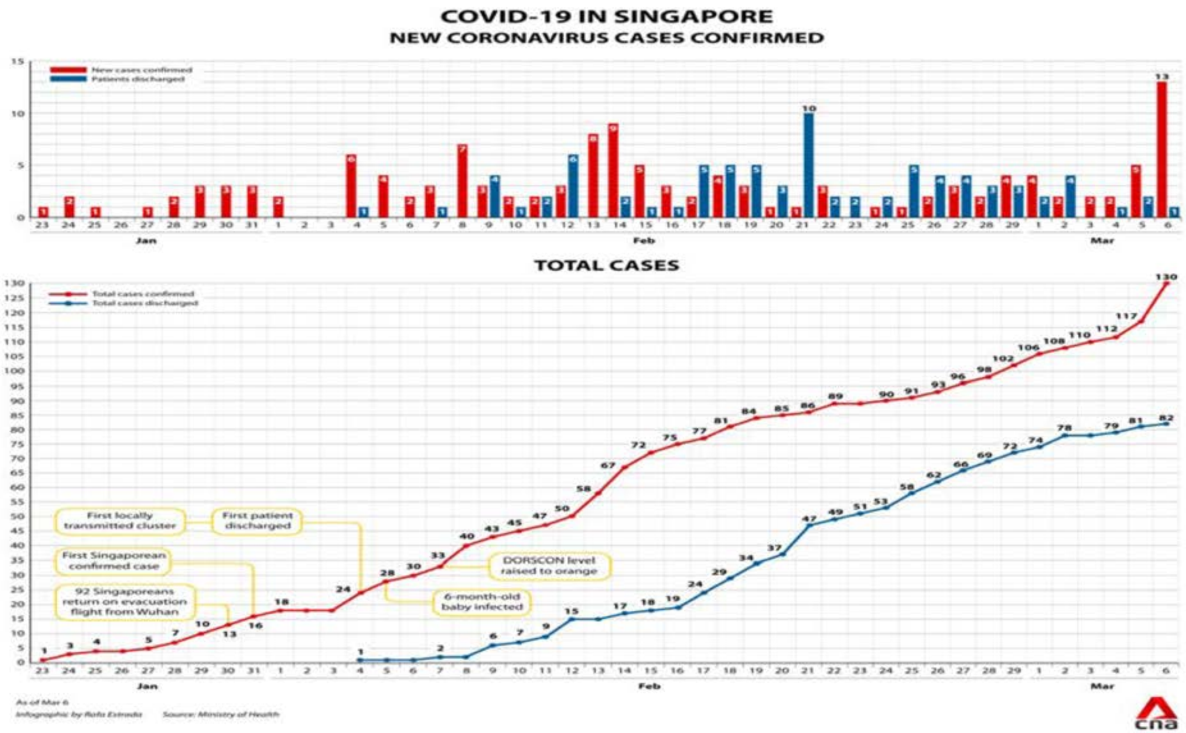


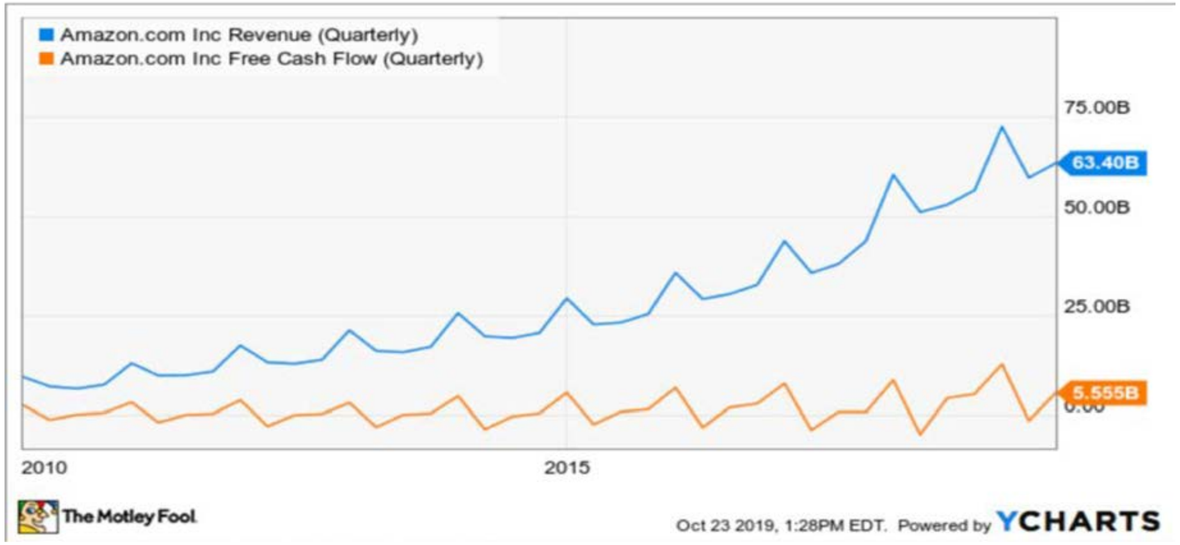
RECESS
Overview,
Implementation &
Value Generation
Weeks 1-3
Foundations in
R programming
Weeks 1-4
Descriptive
Analytics
Weeks 3-6
Linear Regression
Time Series Forecasting
Data Mining
Optimization
Predictive/
Prescriptive
Analytics
Week
7-8
Week
10
Week 11-12

Learning outcomes

- Understand concepts important to time-series modeling, such as trends, seasonality, lagged analysis
- Understand and be able to use simple time-series models such as the Moving Average, Exponential Smoothing, and Holt-Winters models

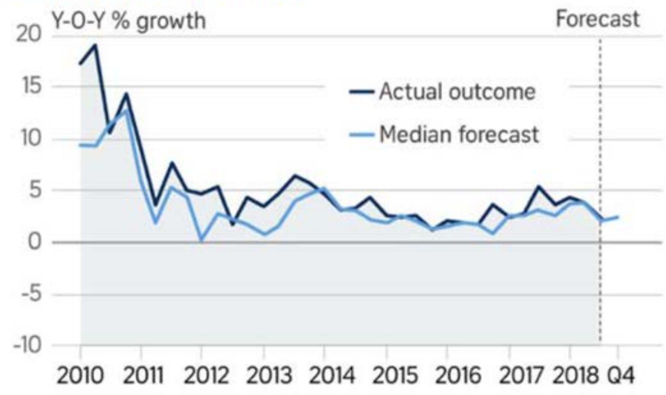
Predictive Analytics over Time





The economy is forecast to grow by 3.3% in 2018: MAS survey

QUARTERLY GDP GROWTH



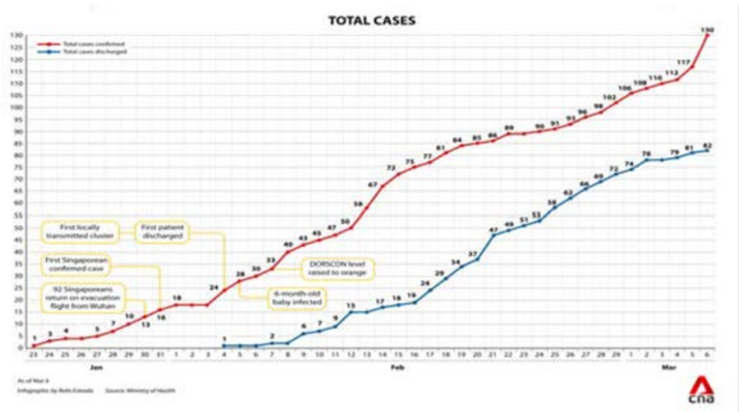
MEDIAN FORECASTS OF MACROECONOMIC INDICATORS

Key macroeconomic indicators year-on-year % change	Sept survey	Current survey
GDP	3.2	**3.3**
Manufacturing	7.6	**7.4**
Finance & insurance	6.7	**6.9**
Construction	-4.2	**-3.5**
Wholesale & retail trade	1.5	**1.3**
Accommodation & food services	2.9	**3.4**
Private consumption	2.8	**3.4**
Non-oil domestic exports	5.0	**6.2**

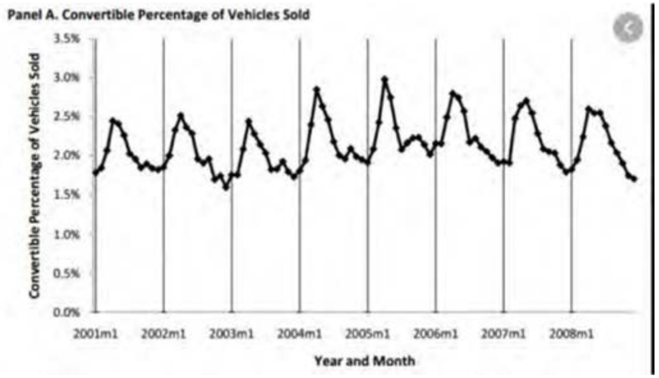
Source: MAS
STRAITS TIMES GRAPHICS

Time series

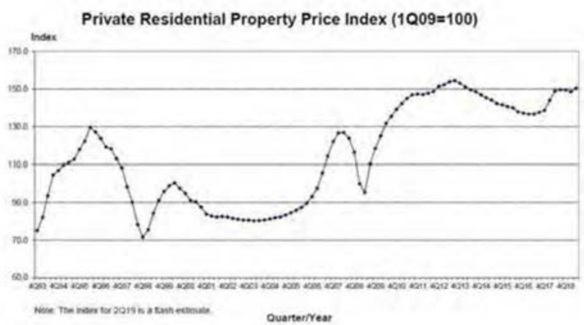
- **Time series**: stream of historical data such as daily attendance, weekly sales, etc..
- Values of a time series over T periods can be characterized as Y_t, for t=1,2,….,T.
- Time series can have one or more of the following:



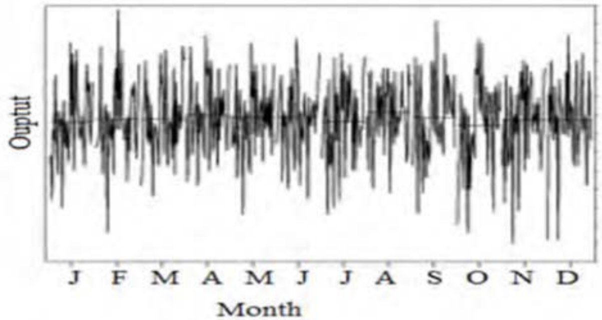
Trends: a gradual upward or downward movement of a time series over time



Seasonal effects: an effect that occurs/repeats at a fixed time interval (e.g. day, week, monthly, year)



Cyclical effects: ups and downs over much longer time frame that do not have a fixed interval/length



Stationarity (stationary time series): when statistical properties of the time series (e.g. mean, variance) do not change over time; time series only exhibit random behavior

Time series models

- A Time series model can include many different types of predictors

$$Y_t = b_0 + b_1 Y_{t-1}$$

Outcome variable at previous time-step(s)

$$+ \ b_2 X_t$$

Predictors/ Regressors at current (or other) time steps

$$+ \ \ldots$$

Variables are indexed by time

In the next few slides, we shall first focus on several forecasting models using only Y's at previous timesteps before we go onto using other regressors.

Forecasting Models for Stationary Time Series

Two simple approaches useful over short time periods when trend, seasonal, or cyclical effects are not significant are

1) simple moving average
2) simple exponential smoothing

Simple Moving Average

A k-period simple moving average model is:

$$\hat{Y}_{t+1} = \frac{1}{k}\left(Y_t + Y_{t-1} + \ldots + Y_{t-(k-1)}\right)$$

- $\hat{Y}_{t+1}$: Forecasted value for period *t*
- Y_t : Observed value in period *t*
- *k* : Average over most recent *k* periods

Smoothing method based on idea of averaging random fluctuations in the time series to identify underlying direction in which the time series is changing.

Assumes future will be similar to recent past

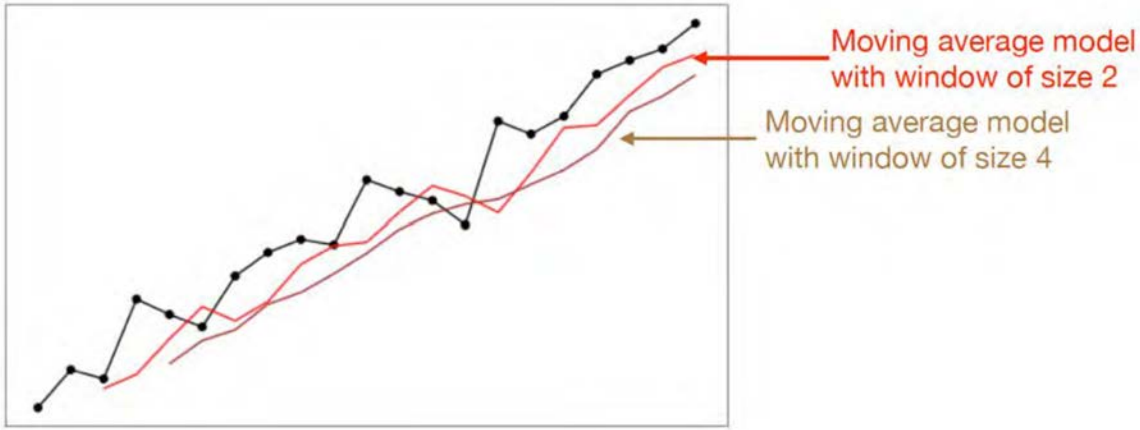
- In general this is called Autocorrelation: when a variable is correlated with itself over time

Simple method but has proven to be quite useful in stable environments such as inventory management; suitable for short-range forecasting

Simple Moving Average

A k-period simple moving average model is:

$$\hat{Y}_{t+1} = \frac{1}{k}\left(Y_t + Y_{t-1} + \ldots + Y_{t-(k-1)}\right)$$



Value of K is somewhat arbitrary but choice affects accuracy

larger K -> current forecast is more dependent on older data and not react as quickly to fluctuations in the time series; smaller K -> quicker the forecast response to changes in time series

larger K -> extreme values has less effect on forecasts

Simple Moving Average

In R, there are several ways to calculate simple moving averages.

Here we use the **TTR** package:

Tablet Computer Sales (c.sales)

	Week	Units Sold	UnitPred
1	1	88	NA
2	2	44	NA
3	3	60	66.0
4	4	56	52.0
5	5	70	58.0
6	6	91	63.0
7	7	54	80.5
8	8	60	72.5
9	9	48	57.0
10	10	35	54.0
11	11	49	41.5
12	12	44	42.0
13	13	61	46.5
14	14	68	52.5
15	15	82	64.5
16	16	71	75.0
17	17	50	76.5

```
install.package("TTR")
library(TTR)
SMA(c.sales$`Units Sold`, n=2)
c.sales$UnitPred<-dplyr::lag(SMA(c.sales$`Units Sold`, n=2),1)
```

SMA (): this function computes the prediction for time t,

$$\hat{Y}_t = \frac{1}{n}\left(Y_t + Y_{t-1} + \ \dots + Y_{t-(n-1)}\right)$$

n: number of windows, which is the k in the previous equation

To shift the prediction one time-step (or downward by one row in the table) to use it as prediction for next time-step, we use dplyr::lag(x,1) (dplyr::lag(x) also works as default lag is 1)

Error Metrics & Forecast Accuracy

Mean Absolute Deviation	MAD = $\frac{1}{n}\sum_t^n \left\lvert Y_t - \widehat{Y}_t \right\rvert$
Mean Square Error	MSE = $\frac{1}{n}\sum_t^n \left(Y_t - \widehat{Y}_t\right)^2$
Root Mean Square Error	RMSE = $\sqrt{\frac{1}{n}\sum_t^n \left(Y_t - \widehat{Y}_t\right)^2}$
Mean Absolute Percentage Error	MAPE = 100 x $\frac{1}{n}\sum_t^n \left\lvert \frac{Y_t - \widehat{Y}_t}{Y_t} \right\rvert$

MAD is less affected by extreme observations; preferred to MSE when extreme observations are rare & have no special meaning

MSE penalized larger errors - most commonly used

RMSE is in same units as data (allows more practical comparison compared to MSE)

MAPE is average of absolute errors divided by actual observation values, allowing for better relative comparison

There is no universal agreement on which is best.

Some of the R functions, such as HoltWinters() already does its own optimization to choose its parameters.

Comparing Error Metrics

Going back to Tablet Computer Sales (c.sales)

	Week	Units Sold	UnitPred2	AbsDev2	SqErr2	AbsErr2	UnitPred3	AbsDev3	SqErr3	AbsErr3	UnitPred4	AbsDev4	SqErr4	AbsErr4
1	1	88	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	2	44	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	3	60	66.0	6.0	36.00	10.000000	NA	NA	NA	NA	NA	NA	NA	NA
4	4	56	52.0	4.0	16.00	7.142857	64.00000	8.0000000	64.0000000	14.2857143	NA	NA	NA	NA
5	5	70	58.0	12.0	144.00	17.142857	53.33333	16.6666667	277.7777778	23.8095238	62.00	8.00	64.0000	11.4285714
6	6	91	63.0	28.0	784.00	30.769231	62.00000	29.0000000	841.0000000	31.8681319	57.50	33.50	1122.2500	36.8131868
7	7	54	80.5	26.5	702.25	49.074074	72.33333	18.3333333	336.1111111	33.9506173	69.25	15.25	232.5625	28.2407407
8	8	60	72.5	12.5	156.25	20.833333	71.66667	11.6666667	136.1111111	19.4444444	67.75	7.75	60.0625	12.9166667
9	9	48	57.0	9.0	81.00	18.750000	68.33333	20.3333333	413.4444444	42.3611111	68.75	20.75	430.5625	43.2291667
10	10	35	54.0	19.0	361.00	54.285714	54.00000	19.0000000	361.0000000	54.2857143	63.25	28.25	798.0625	80.7142857
11	11	49	41.5	7.5	56.25	15.306122	47.66667	1.3333333	1.7777778	2.7210884	49.25	0.25	0.0625	0.5102041
12	12	44	42.0	2.0	4.00	4.545455	44.00000	0.0000000	0.0000000	0.0000000	48.00	4.00	16.0000	9.0909091
13	13	61	46.5	14.5	210.25	23.770492	42.66667	18.3333333	336.1111111	30.0546448	44.00	17.00	289.0000	27.8688525
14	14	68	52.5	15.5	240.25	22.794118	51.33333	16.6666667	277.7777778	24.5098039	47.25	20.75	430.5625	30.5147059
15	15	82	64.5	17.5	306.25	21.341463	57.66667	24.3333333	592.1111111	29.6747967	55.50	26.50	702.2500	32.3170732
16	16	71	75.0	4.0	16.00	5.633803	70.33333	0.6666667	0.4444444	0.9389671	63.75	7.25	52.5625	10.2112676
17	17	50	76.5	26.5	702.25	53.000000	73.66667	23.6666667	560.1111111	47.3333333	70.50	20.50	420.2500	41.0000000
				MAD= 13.63	MSE= 254.38	MAPE= 23.63		MAD= 14.86	MSE= 299.84	MAPE= 25.37		MAD= 16.13	MSE= 355.25	MSE= 28.07

k=2 *k=3* *k=4*

Based on the error metrics, which is the best forecasting model?

Simple Exponential Smoothing

An Exponential Smoothing Model is:

$$\hat{Y}_{t+1} = (1-\alpha)\hat{Y}_t + \alpha Y_t$$
$$= \hat{Y}_t + \alpha(Y_t - \hat{Y}_t)$$

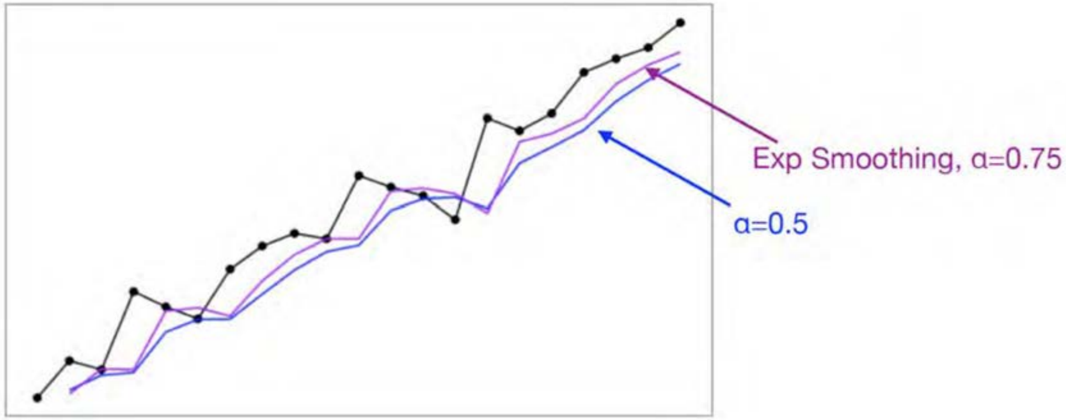
α: smoothing constant $0 \leq \alpha \leq 1$

$\hat{Y}_t$: Model prediction at previous time-step

Model can be predicted with just previous forecast ($\hat{Y}_t$) and actual value (Y_t) , and choosing the smoothing constant (α).

The closer alpha (α) is to 1, the quicker the model responds to changes in the time series as it puts more weight on the actual current observation.

To begin, set $\hat{Y}_1$ and $\hat{Y}_2$ to be Y_1. Then choose the α that produces the lowest error metrics.



Difference from SMA: ESM gives

- more weight to more recent values
- takes into account all past data

Forecasting Models for Time Series with a Linear Trend

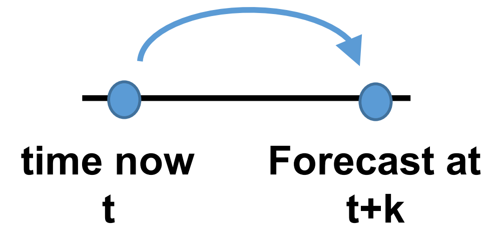
Unlike the single exponential smoothing model that cannot handle any trends or seasonality, the double exponential smoothing model can account for trends, using the following linear trend equation:

$$\hat{Y}_{t+k} = \overset{\text{“intercept”}}{a_t} + \overset{\text{“slope”}}{b_t} k$$

$$a_t = \alpha \hat{Y}_t + (1-\alpha)(\underbrace{a_{t-1} + b_{t-1}}_{\text{looks like } \hat{Y}_t})$$

$$b_t = \beta(\underbrace{a_t - a_{t-1}}_{\text{updated slope estimate}}) + (1\text{-}\beta)\, \underbrace{b_{t-1}}_{\text{previous slope estimate}}$$

FYI ONLY: NO NEED TO MEMORIZE OR UNDERSTAND DERIVATION OF THE EQUATIONS



Interpretation:

“Intercept” (a_t) is known as level component; is like the simple exponential model

“slope” (b_t) is known as trend component

At each time-step, we are smoothing a_t and b_t using parameters α (smoothing constant) and β (trend smoothing constant).

Implementation:

Initialize a_1 as Y_1 and b_1 as Y_2-Y_1

Vary α and β and find best estimates of a_t and b_t for entire time series which can then be used for forecasting

Forecasting Time Series with Seasonality

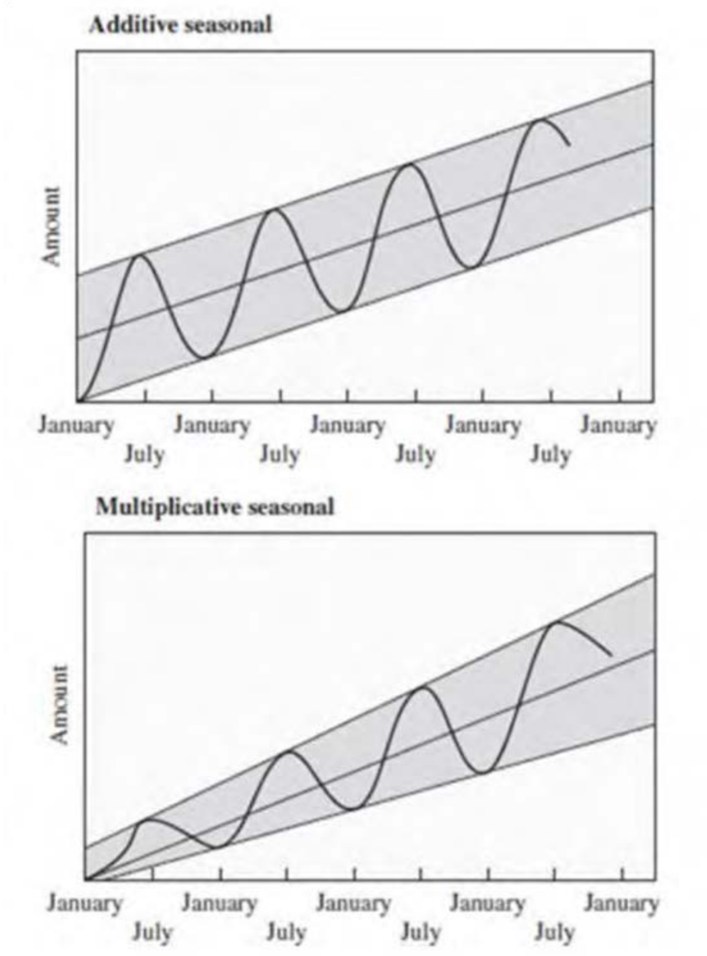
Holt-Winters models are similar to exponential smoothing models but it adds a 3rd smoothing parameter (gamma) that can smooth out seasonality.

In R, the Holt-Winters () function can do all 3 types of exponential smoothing:

```
HoltWinters(x, alpha, beta, gamma, ...)
```

Method	Trend	Seasonality	Parameters	R Function
Simple Exponential Smoothing	No	No	α	HoltWinters(x, beta=FALSE, gamma=FALSE)
Double Exponential Smoothing	Yes	No	α, β (trend)	HoltWinters(x, gamma=FALSE)
Holt-Winters (Triple Exponential Smoothing)	Yes	Yes	α, β (trend), γ (seasonality)	HoltWinters(x)

Additive vs Multiplicative Seasonality



Additive seasonality:

seasonality is relatively stable (amplitude does not change over time)

Multiplicative seasonality:

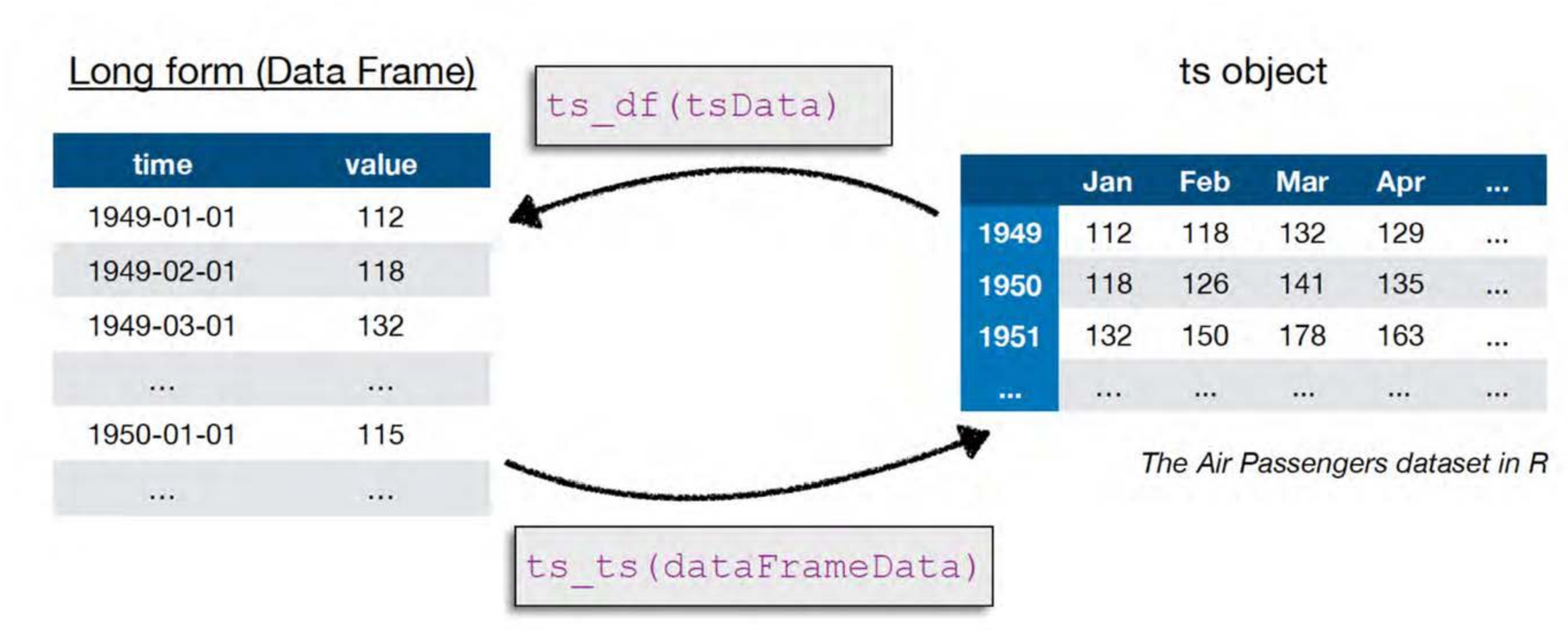
amplitude of time series increases or decreases over time

R’s default seasonality is additive. You can use parameter, seasonal = “multiplicative” if you want to specify as multiplicative

Running HoltWinters: Preparation

R's HoltWinters() expects a ts (time series) object, not a data frame. To convert your dataframe to a ts objective (and vice versa), you can use packages like tsbox

```
install.packages("tsbox")
library(tsbox)
```



time	value
1949-01-01	112
1949-02-01	118
1949-03-01	132
...	...
1950-01-01	115
...	...

	Jan	Feb	Mar	Apr	...
1949	112	118	132	129	...
1950	118	126	141	135	...
1951	132	150	178	163	...
...	...	...	...	...	...

The Air Passengers dataset in R

Reference: https://www.tsbox.help/

Running HoltWinters: Preparation

R's HoltWinters() expects a ts (time series) object, not a data frame. To convert your vector/dataframe to a ts objective, you can also use the ts() function.

```
ts(x, start=, end=, frequency= )
```

Tablet Computer Sales

Week	Units Sold
1	88
2	44
3	60
4	56
5	70
6	91
7	54
8	60
9	48
10	35
11	49
12	44
13	61
14	68
15	82
16	71
17	50

1 to 17 of 17 entries, 2 total columns

start: time of first observation
end: time of last observation
frequency: number of observations per unit time (eg if a year is one unit of time, then 1=annual, 4=quarterly, 12=monthly, 56=weekly, etc.)

Example:

```
tabletdf<-Tablet_Computer_Sales
csale.ts<-ts(tabletdf,start=1,end=17,freq=1)
```

frequency set to 1 to treat a week as one unit of time

Running HoltWinters: Example 1

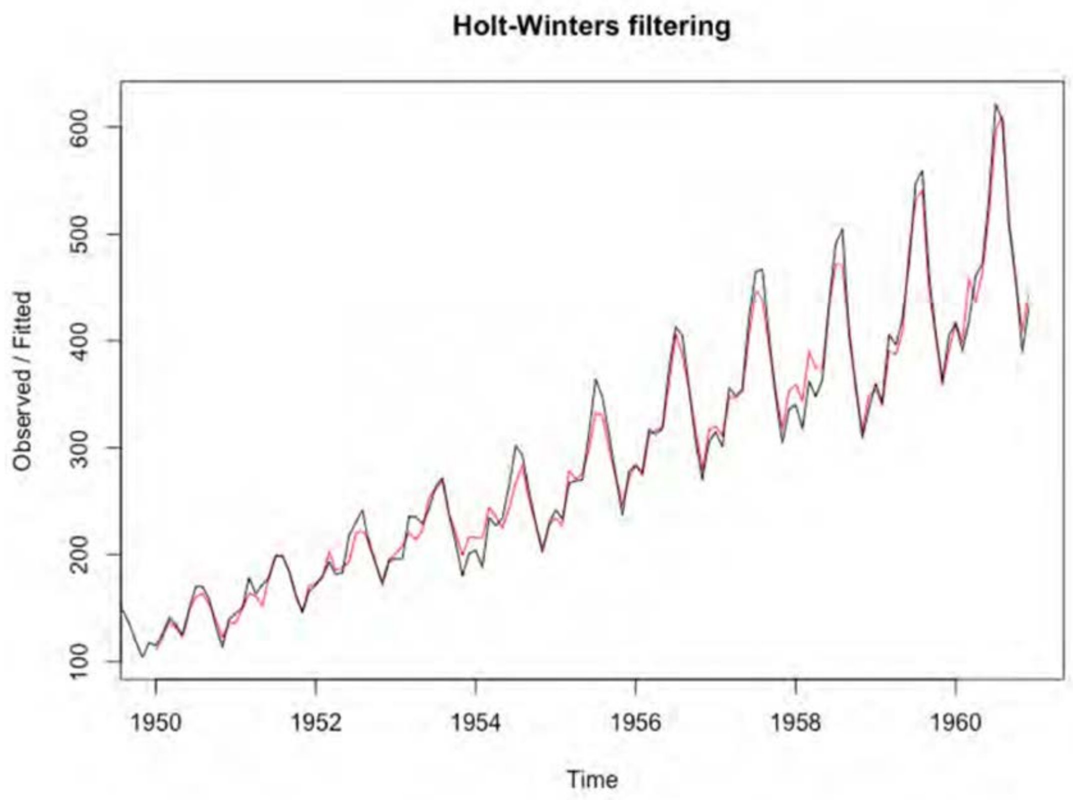
```
# Using AirPassengers dataset, which is already a ts object
> dataset (AirPassengers)
> hw1<-HoltWinters(AirPassengers)
> hw1 # chart can be produced by calling plot(hw1)
```

```
Holt-Winters exponential smoothing with trend and additive seasonal
component.

Call:
HoltWinters(x = AirPassengers)

Smoothing parameters:
 alpha: 0.2479595
 beta : 0.03453373
 gamma: 1

Coefficients:
           [,1]
a   477.827781
b     3.127627
s1  -27.457685
s2  -54.692464
s3  -20.174608
s4   12.919120
s5   18.873607
s6   75.294426
s7  152.888368
s8  134.613464
s9   33.778349
s10 -18.379060
s11 -87.772408
s12 -45.827781
```

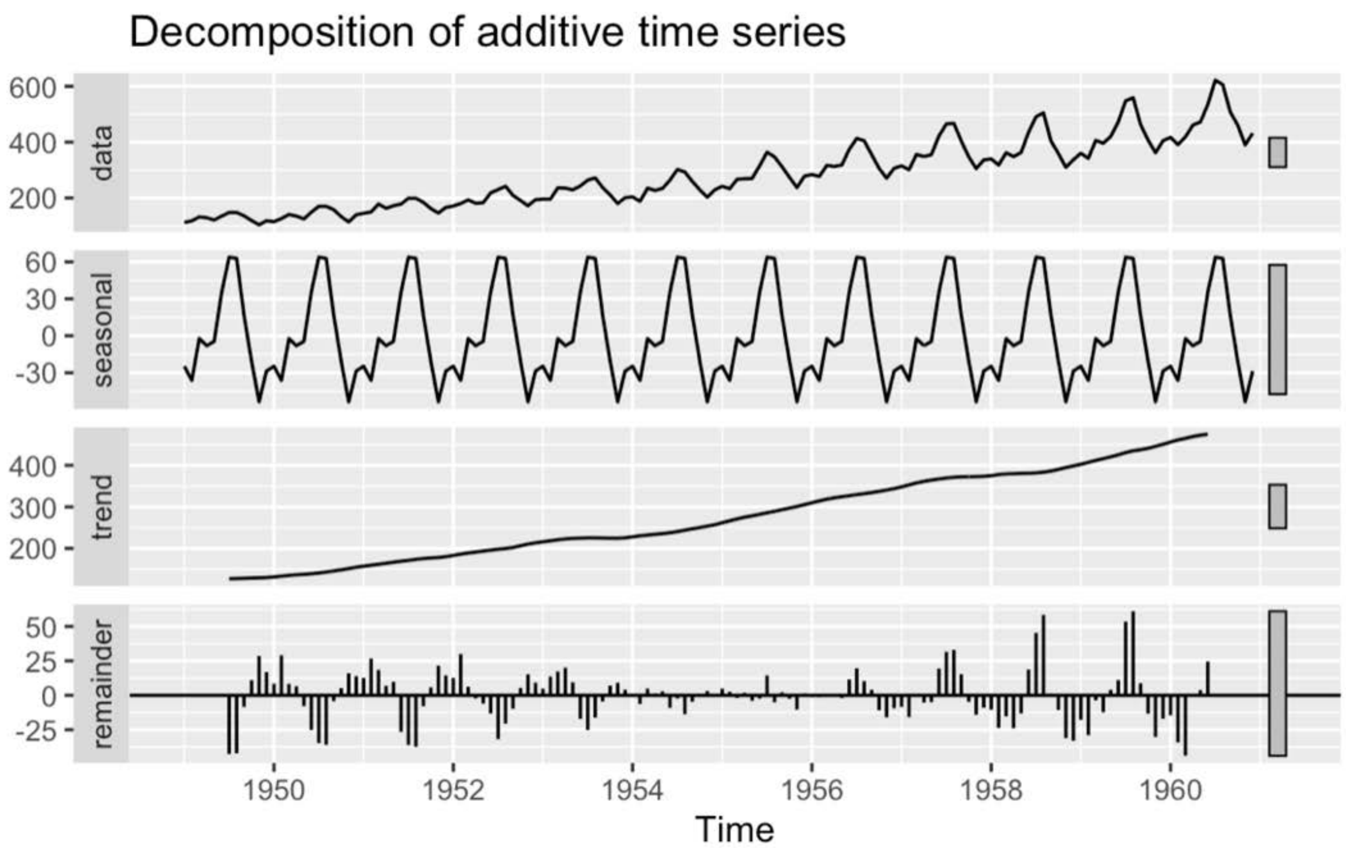


red = fitted

Running HoltWinters: Example 1

In R, you can also use decompose() function to decompose a time series into seasonal, trend and irregular components using moving averages.

```
autoplot(decompose(AirPassengers))
```

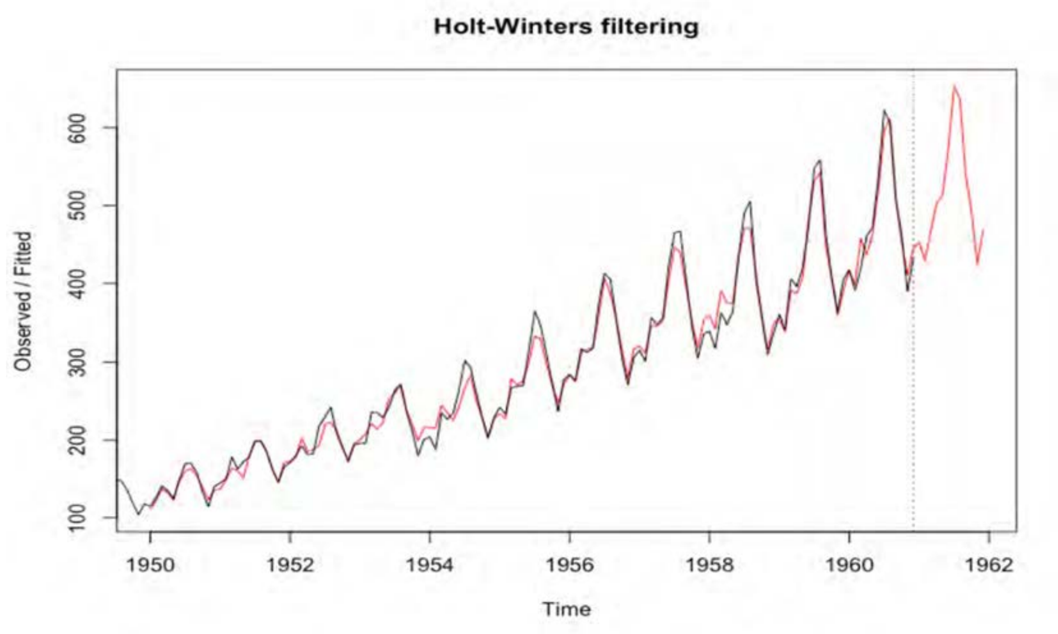


Running HoltWinters: Example 1

To use the model to make a forecast, say for next 12 periods, we use the predict () function

```
hw_pred<-predict(hw1, n.ahead=12) #
# hw_pred<-forecast(hw1,h=12) # uses forecast package
plot(hw1, hw_pred)
```

```
> hw_pred
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep
1961 453.4977 429.3906 467.0361 503.2574 512.3395 571.8880 652.6095 637.4623 539.7548
          Oct      Nov      Dec
1961 490.7250 424.4593 469.5315
```



Running HoltWinters: Example 2

dfHW2

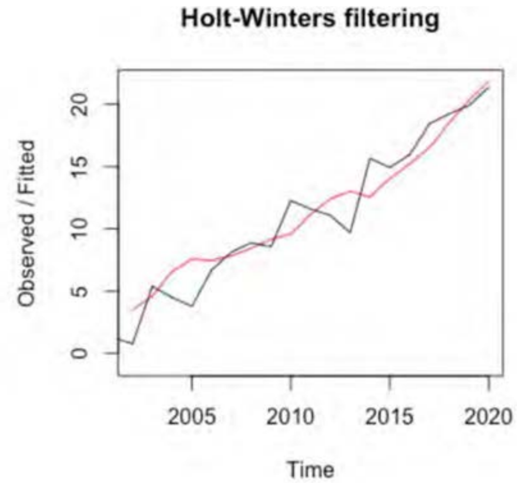
	time	value
1	2000	-0.4420807
2	2001	0.9913492
3	2002	5.6069801
4	2003	4.1453902
5	2004	2.8014861
6	2005	3.2785145
7	2006	5.5658076
8	2007	6.5511773
9	2008	7.3827338
10	2009	9.3783352
11	2010	8.6621183
12	2011	11.6535249
13	2012	10.1436924
14	2013	12.6635982
15	2014	14.5660935
16	2015	15.2000045
17	2016	17.2062843
18	2017	16.9143398
19	2018	18.7554120
20	2019	20.6286540
21	2020	18.9635692

ts_ts(dfHW2)

```
> tsHW2
Time Series:
Start = 2000
End = 2020
Frequency = 1
 [1] -0.4420807  0.9913492  5.6069801  4.1453902  2.8014861  3.2785145
 [7]  5.5658076  6.5511773  7.3827338  9.3783352  8.6621183 11.6535249
[13] 10.1436924 12.6635982 14.5660935 15.2000045 17.2062843 16.9143398
[19] 18.7554120 20.6286540 18.9635692
```

```
# example of a nonseasonal simulated data frame.
set.seed(1) # for reproducibility
dfHW2 = data.frame(time = seq(0,20) + 2000, # "Year"
value = seq(0,20) + rnorm(21,0,1.5))
tsHW2 = ts_ts(dfHW2) #convert to ts
HoltWinters(tsHW2, gamma=FALSE)
```

```
Holt-Winters exponential smoothing with
trend and without seasonal component.
Call:
HoltWinters(x = tsHW2, gamma = FALSE)
Smoothing parameters:
alpha: 0.2005294
beta : 1
gamma: FALSE
```



Note here that if we try to fit a seasonal component and run HoltWinters(tsHW2), R will return an error msg as shown below.

```
> HoltWinters(tsHW2)
Error in decompose(ts(x[1L:wind], start = start(x), frequency = f), seasonal) :
time series has no or less than 2 periods
```

Regression-based forecasting

To model time-series data with seasonality $Y_t = b_0 + b_1 X_t + \ldots + b_{k-1} X_{k-1}$

X_1 to X_{k-1} are dummy variables representing a seasonal categorical variable with k levels. Eg: For month variable with 12 levels, 11 dummy variables could be Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec (Jan is reference month).

To model time-series data with linear trend $Y_t = b_0 + b_1 \text{Time}$

Time is an interval variable, from 1 to n, for n period.

After fitting the regression model, insignificant variables will be dropped from the model. Final model is used for forecasting.

Example 1:

$\text{Flu} = b_0 + b_1 \text{Year}$; fitted model is $\text{Flu} = 345 + 10\text{Year}$

A forecast for year 20 would be $\text{Flu} = 345 + 10*20 = 545$

Example 2:

$\text{Flu} = b_0 + b_1 \text{Spring} + b_2 \text{Summer} + b_3 \text{Fall} + b_4 \text{Year}$

fitted model is $\text{Flu} = 500 + 0.6*\text{Spring} + 0.5*\text{Summer} + 0.8*\text{Fall} + 10*\text{Year}$

Forecast for year 10, winter would be:

$\text{Flu} = 500 - 6*\text{Spring} - 10*\text{Summer} - 8*\text{Fall} + 10*\text{Year}$

$= 500 + 10*10 = 600$

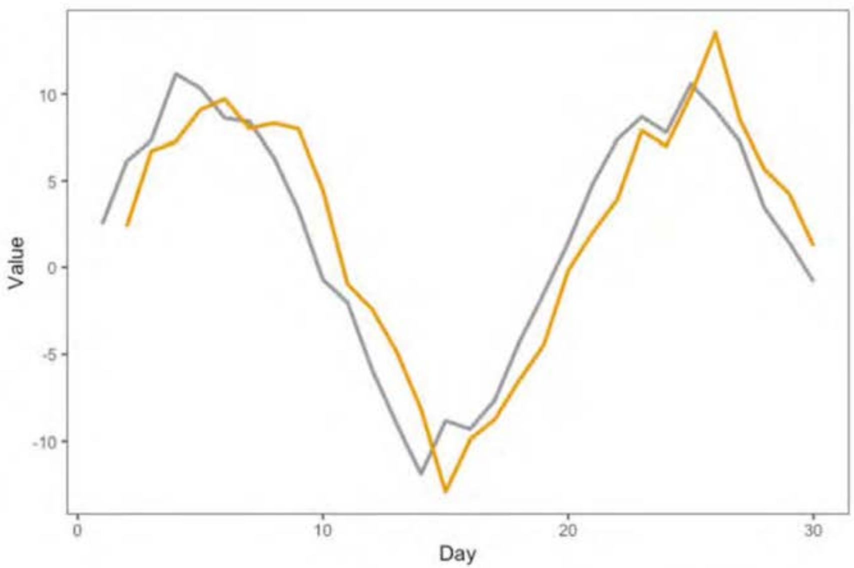
Regression-based forecasting with Causal Variables

$$Y_t = b_0 + b_1 X_t$$

Sometimes Xt and Yt may be correlated, but also offset in time.

- If changes in Xt precedes changes in Yt, then the X variable is called a leading variable. It can also be considered a causal variable.
- Conversely, a variable that follows movements of other variables is called a lagging variable.

(If both tend to co-vary at the same time, these variables are coincident)



Which variable (gold or grey) is the leading variable, and which is the lagging variable?

Regression-based forecasting

Examples:

Leading variable	Dependent variable
Covid-19 infection rate on day (t-lag)	Number of people vaccinated on day t
Advertising expenditure in period (t-lag)	Sales in period t
Time spent studying on day (t-lag)	Grade on exam day t

If X is a leading variable of Y
(i.e.,, we expect X's at previous time points to impact Y at the current time),
we can use lagged values of X (i.e., $X_{t\text{-lag}}$) to predict the current Y_t.

$$Y_t = b_0 + b_1 X_{t-1}$$

You can also regress a dependent variable against a past (lagged) version of itself:

$$\textbf{Sales}_t = b_0 + b_1 \textbf{Sales}_{t-1}$$

This is called an Autoregressive model (of order 1, or AR(1)).

Summary

- Statistical time-series models are commonly used for forecasting problems. Moving-average models, exponential smoothing models, and regression-based forecasting are amongst the most popular but simple time-series models are mostly used for short- and medium-range forecasts while regression analyses is more popular for long-range forecasting.
- No one model is the best; Error metrics help to identify models with less prediction errors.
- Times series data can have trends, seasonality, and autocorrelation. Plotting the data can be useful in visualizing and identifying these.
- As statistical methods are unable to capture factors such as sudden shocks in the economy, a pandemic, large one-time orders, or intangible factors, managers tend to supplement these quantitative forecasting techniques with their qualitative judgement. Comparing quantitative forecasts to judgmental forecast is an important way to see if the quantitative forecasting approach is adding value in terms of improving forecasting.