Lecture 9-c
Time Series: Forecasting with ARIMA & Exponential Smoothing

 1

Review: ARIMA Models – Box-Jenkins

• How do we select p, q, and d for an ARIMA model?

• Box-Jenkins Approach

1) Make sure data is stationary –check a time plot. If not, differentiate.

2) Using ACF & PACF, guess small values for p & q.
If order choice not clear, use AIC, AIC Corrected (AICc), BIC, or HQC (Hannan and Quinn (1979)).

3) Estimate order p, q. (ML, MM, OLS for AR; Innovation algorithm for MA, Hannan-Rissanen algorithm for ARMA)

4) Run diagnostic tests on residuals (Check ACF, LB tests).
⇒ Are they white noise? If not, add lags (p or q, or both).

• Value parsimony. When in doubt, keep it simple (KISS).

• Looks simple, but there are a lot of nuances to the process.

Review: ARIMA Models – Box-Jenkins

• With non-stationary series, we talked about trends:
- Deterministic vs Stochastic

⇒ we remove pattern, either by **detrending** (deterministic trends) or **differencing** (stochastic trends).

• Similar situation arises when we have seasonal patterns, which can also be deterministic or stochastic. In general, we remove the seasonal pattern using **seasonal dummies** (deterministic seasonalities). Once removed we follow Box-Jenkins to select an ARIMA model.

• Then, we forecast. We find
- MA(q) forecasts become mean forecasts after q periods.
- AR(p) are very correlated forecasts, using past p forecasts.
- ARMA(p, q) are a combination of both; after q periods, AR dominates.

Non-Stationarity in Variance

• Stationarity in mean does not imply stationarity in variance. However, non-stationarity in mean implies non-stationarity in variance.

• If the mean function is time dependent:
1. The variance, Var(y_t) is time dependent.
2. Var[y_t] is unbounded as $t \to \infty$.
3. Autocovariance functions and ACFs are also time dependent.
4. If t is large with respect to the initial value y_0, then $\rho_k \approx 1$.

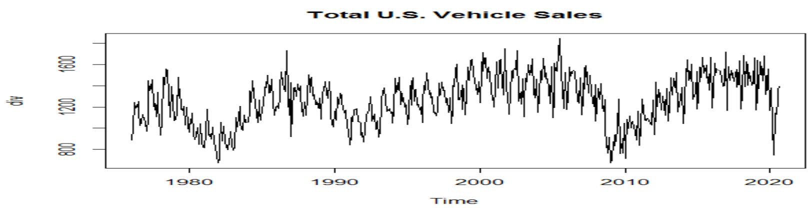
• It is common to use *variance stabilizing* transformations: Find a function $G(.)$ so that the transformed series $G(y_t)$ has a constant variance. Very popular transformation:

$$G(Y_t) = \log(Y_t)$$

Non-Stationarity in Variance – Log Transform

Example: We log transform the monthly variable Total U.S. Vehicle Sales data (1976: Jan – 2020: Sep):

```
Car_da <- read.csv("https://www.bauer.uh.edu/rsusmel/4397/TOTALNSA.csv",
head=TRUE, sep=",")
x_car <- Car_da$TOTALNSA

library(tseries)
ts_car <- ts(x_car,start=c(1976,1),frequency=12)
plot.ts(ts_car,xlab="Time",ylab="div", main="Total U.S. Vehicle Sales")
```

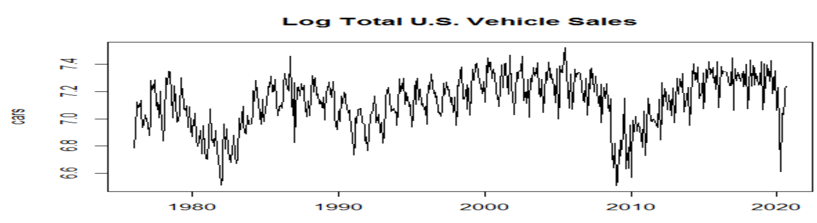



```
> mean(x_car)
[1] 1260.818
> sd(x_car)
[1] 225.5706
```

Non-Stationarity in Variance – Log Transform

Example (continuation):

```
l_car <- log(ts_car)
> plot.ts(l_car,xlab="Time",ylab="div", main="Log Total U.S. Vehicle Sales")
```




```
> mean(l_car)
[1] 7.122416
> sd(l_car)
[1] 0.1889378
```

Note: Big reduction in volatility. Though pattern of series not significantly changed.

Non-Stationarity in Variance – Box-Cox

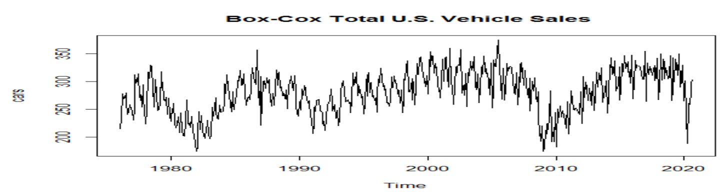
• Another popular transformation is the the Box-Cox transformation:

$$G(Y_t) = \frac{Y_t^{\lambda} - 1}{\lambda}$$

where λ>0, usually between 0 and 2 (it can be estimated too). When λ=1, we have a linear y_t; when $\lambda \to 0$, a log transformation for y_t.

Example: We do a Box-Cox transformation of the monthly variable Total U.S. Vehicle Sales data (1976: Jan – 2020: Sep), setting λ = 0.75:

```
lambda <- 0.75
b_cox_car <- (ts_car^lambda - 1)/lambda
> plot.ts(b_cox_car, xlab="Time",ylab="cars", main=" Box-Cox Total U.S. Vehicle Sales")
```



Variance Stabilizing Transformation - Remarks

• Variance stabilizing transformation is only done for positive series, usually for nominal series (say, in USD total retail sales or units, like Total U.S. vehicle sales). If a series has negative values, then we need to add each value with a positive number so that all the values in the series are positive.

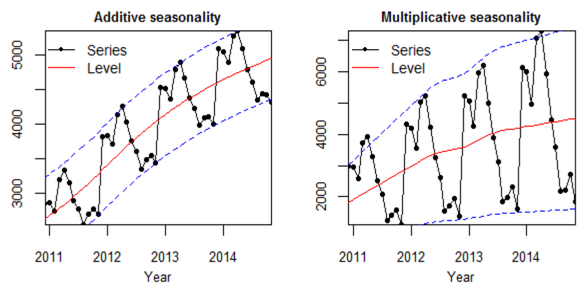
• Then, we can search for any need for transformation.

• It should be performed before any other analysis, such as finding an ARMA(p, q) model or differencing.

• Not only stabilize the variance, but we tend to find that it also improves the approximation of the distribution by Normal distribution.

Seasonal Time Series

• In time series, seasonal patterns ("*seasonalities*") can show up in two forms: additive and multiplicative.
- Additive: The seasonal variation is independent of the level.
- Multiplicative: The seasonal variation is a function of the level.

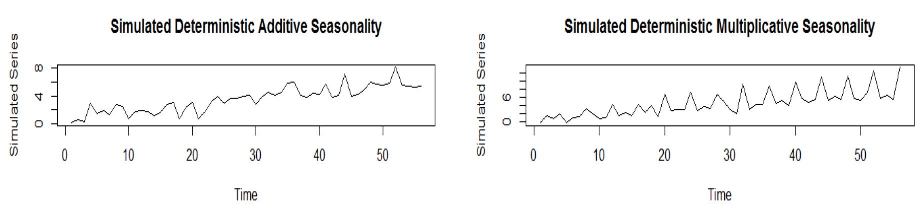


Note: In the multiplicative case, the amplitude of the seasonal pattern is changing over time, while in the additive the amplitude is constant.

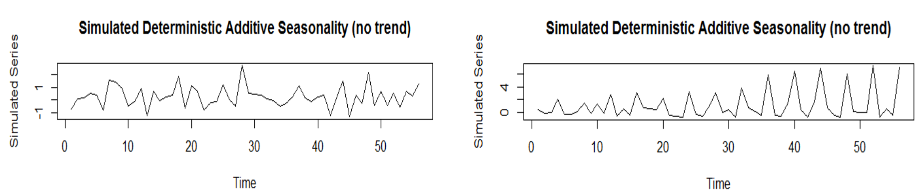
Seasonal Time Series

Examples: We simulate the two seasonal patterns, additive and multiplicative, with trend and no trend.

A. With trend



B. With no trend



Seasonal Time Series: Types

• Two types of seasonal behavior:

- **Deterministic** – Usual treatment: Build a deterministic function,

$$f(t) = f(t + k \times s), \qquad k = 0, \pm 1, \pm 2, \cdots$$

We can include seasonal (means) dummies, for example, monthly or quarterly dummies. (This is the approach in Brooks' Chapter 10).

Instead of dummies, trigonometric functions (sum of cosine curves) can be used. A linear time trend is often included in both cases.

-**Stochastic** – Usual treatment: SARIMA model. For example:

$$y_t = \theta_0 + \Phi_1\, y_{t-s} + \varepsilon_t + \Theta_1 \varepsilon_{t-s}$$

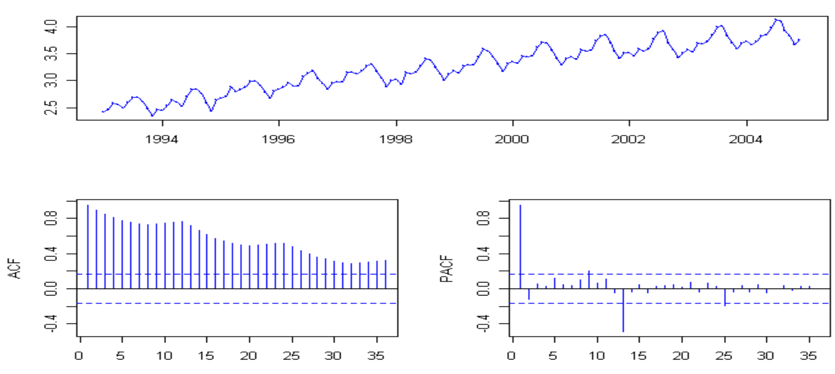
or

$$(1 - \Phi_1 L^s)\, y_t = (1 - \Theta_1 L^s)\, \varepsilon_t$$

where *s* the seasonal periodicity –associated with the frequency– of y_t. For quarterly data, *s* = 4; monthly, *s*= 12; daily, *s* = 7, etc.

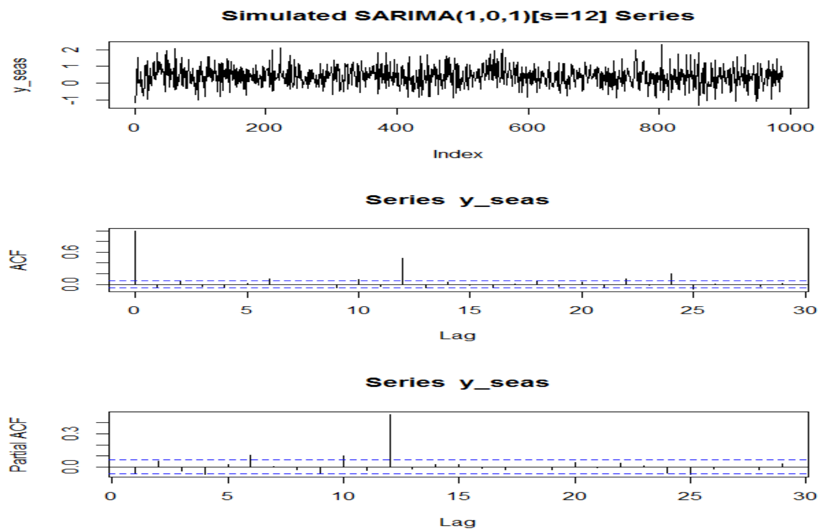
Seasonal Time Series – Visual Patterns

• The raw series along with the ACF and PACF can be used to discover seasonal patterns.



Signs: Periodic repetitive wave pattern in ACF, repetition of significant ACFs, PACFs after *s* periods.

Seasonal Time Series – Visual Patterns



Sign: Significant spikes in ACF/PACF at frequency *s*, in this case *s*= 12.

Seasonal Time Series – Deterministic

• We use **seasonal dummy variables**, say monthly, in a linear model to capture the seasonalities. Depending on the seasonality pattern, we have different specifications to remove the pattern.

• Suppose y_t has monthly frequency and we suspect that in every December y_t increases.
– For the additive model, we can regress y_t against a constant and a December dummy, $\boldsymbol{D}_t$:

$$y_t = \mu + \boldsymbol{D}_t \ \boldsymbol{\mu}_s + \varepsilon_t$$

For the multiplicative model, we can regress y_t against a constant and a December dummy, $\boldsymbol{D}_t$, interacting with a trend, t:

$$y_t = \mu + \boldsymbol{D}_t \ \boldsymbol{\mu}_s * t + \varepsilon_t$$

The residuals of these regressions, e_t, –i.e., e_t = *filtered* y_t, free of "monthly seasonal effects"– are used for further ARMA modeling.

Seasonal Time Series – Deterministic

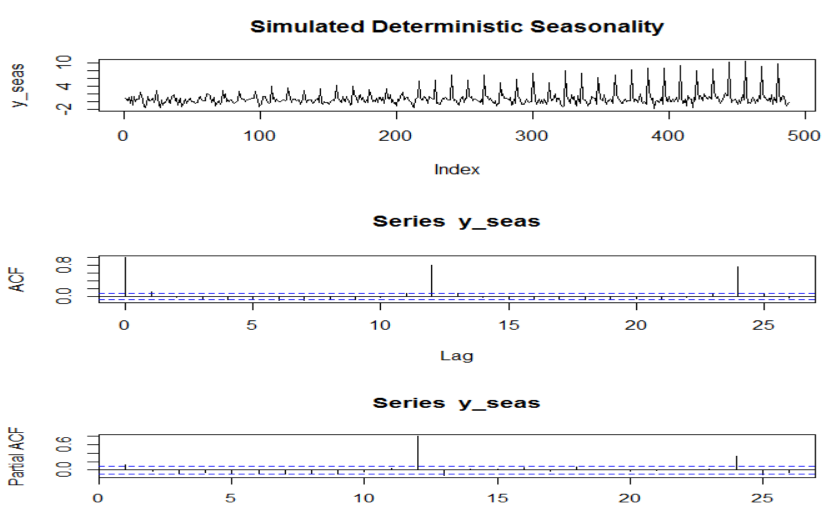
Example: We simulate an AR(1) series, with a multiplicative December seasonal behavior.

$$y_t = \mu + \phi_1 y_{t-1} + D_t \,\boldsymbol{\mu_s} * t + \varepsilon_t$$

```
T_sim <- 500                          # Size of simulation
y_sim <- matrix(0,T_sim,1)            # vector to accumulate simulated data
Seas_12 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1), (length(y_sim)/12+1))   # Create Oct dummy
u <- rnorm(T_sim, sd=0.75)            # Draw T_sim normally distributed errors
phi1 <- 0.2                           # Change to create different correlation patterns
k <- 12                               # Seasonal Periodicity
a <- k+1                              # Time index for observations
mu <- 0.2
mu_s <- .02
while (a <= T_sim) {
        y_sim[a] = mu + phi1 * y_sim[a-1] + Seas_12[a] * mu_s * a + u[a]
        # y_sim simulated autocorrelated values
a <- a + 1
}
y_seas <- y_sim[(k+1):T_sim]
plot(y_seas, type="l", main="Simulated Deterministic Seasonality")
```

Seasonal Time Series – Deterministic

Example (continuation): We plot simulated series, ACF, & PACF.



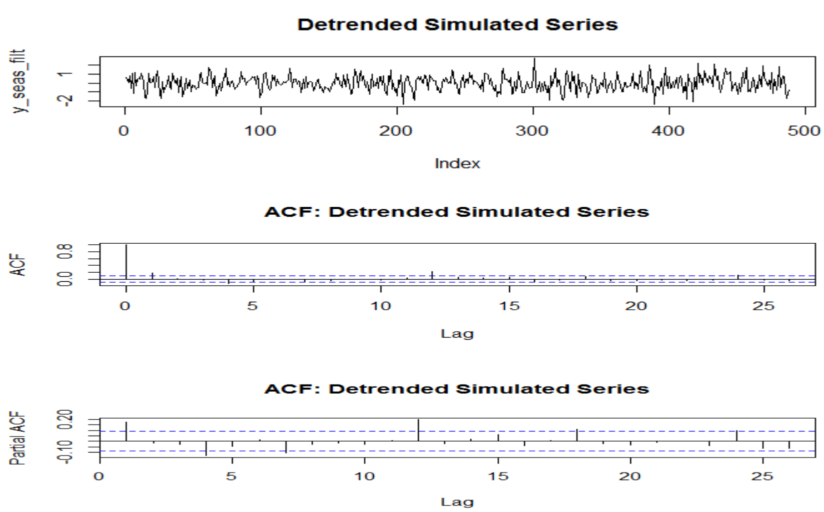
Seasonal Time Series – Deterministic

Example (continuation): We detrend ("filter" the simulated series).

```
trend <- c(1:T_sim)
trend_sim <- trend[(k+1):T_sim]
seas_d <- Seas_12[(k+1):T_sim]
sea_trend <- seas_d*trend_sim
fit_seas <- lm(y_seas ~ seas_d + trend_sim + sea_trend)
> summary(fit_seas)

Coefficients:
             Estimate   Std. Error t value Pr(>|t|)
(Intercept)  0.1356538  0.0804474   1.686  0.09239 .
seas_d       0.6929134  0.2859528   2.423  0.01575 *
trend_sim    0.0008504  0.0002749   3.093  0.00209 **
sea_trend    0.0174034  0.0009766  17.821  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8209 on 484 degrees of freedom
Multiple R-squared:  0.7929,    Adjusted R-squared:  0.7917
F-statistic: 617.8 on 3 and 484 DF,  p-value: < 2.2e-16
```

Seasonal Time Series – Deterministic

Example (continuation): We plot the detrended simulated series, along with the ACF and PACF.



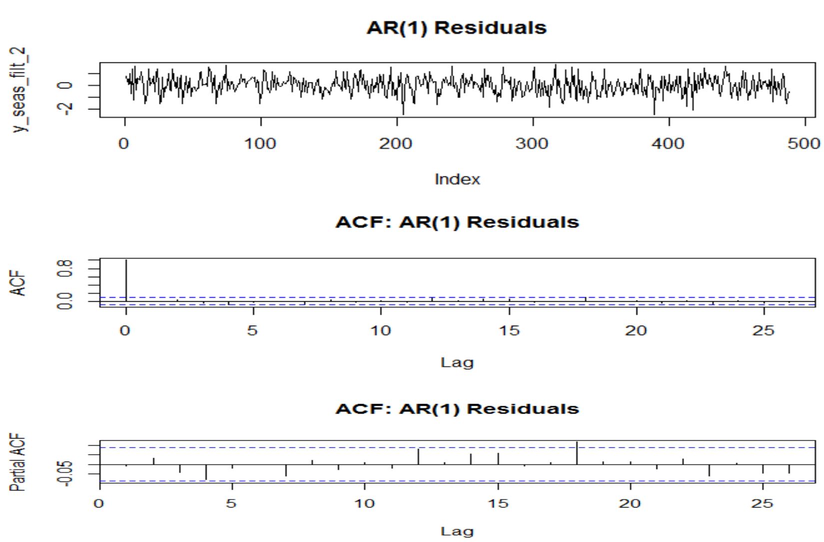
Seasonal Time Series – Deterministic

Example (continuation): The December seasonal pattern is gone from the detrended series. We run an ARIMA(1,0,0):

```
> fit_y_seas_ar1 <- arima(y_seas_filt, order=c(1,0,0))

Call:
arima(x = y_seas_filt, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.1785    -0.0001                    ⇒ Very close to phi1 = 0.20.
s.e.  0.0446     0.0443

sigma^2 estimated as 0.6471:  log likelihood = -586.26,  aic = 1178.51

y_seas_filt_2 <- fit_y_seas_det_ar1$residuals                # Extract Residuals

plot(y_seas_filt_2,type="l", main="AR(1) Residuals")
acf(y_seas_filt_2, main="ACF: AR(1) Residuals")
pacf(y_seas_filt_2, main="“PACF: AR(1) Residuals")
```

Seasonal Time Series – Deterministic

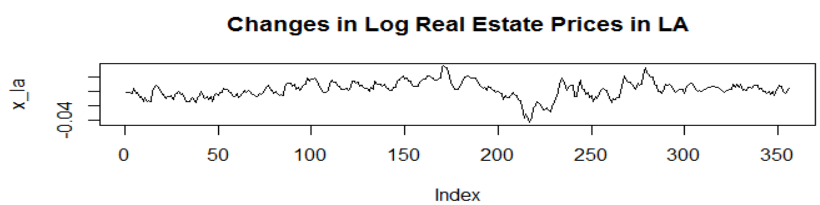
Example (continuation): We check the residuals of AR(1) regression



Seasonal Time Series – Deterministic

Example: We model log changes in real estate prices in the LA market, y_t. First, we run a regression to remove (filter) the monthly effects from y_t. Then, we model y_t as an ARMA(*p, q)* process.

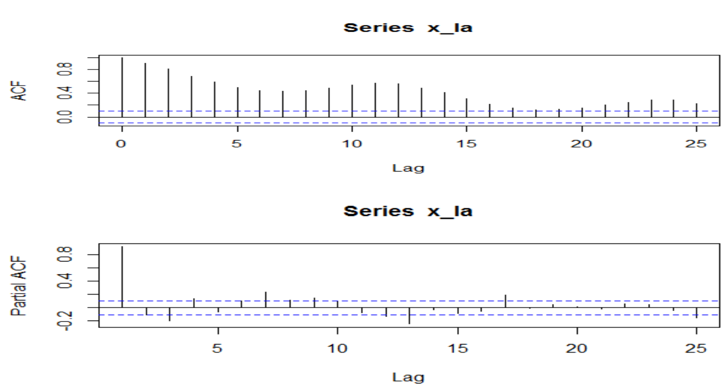
```
RE_da <- read.csv(" http://www.bauer.uh.edu/rsusmel/4397/Real_Estate_2019.csv",
head=TRUE, sep=",")
x_la <- RE_da$LA_c
zz <- x_la
T <- length(zz)
plot(x_la, type="l", main="Changes in Log Real Estate Prices in LA")
```



Seasonal Time Series – Deterministic

Example (continuation): We look at the ACF & PACF for LA

```
> acf(x_la)
> pacf(x_la)
```



Note: ACF shows highly autocorrelated data, with some seasonal pattern (there is a periodic decreasing wave).

Seasonal Time Series – Deterministic

Example (continuation): We define monthly dummies

```
Feb1 <- rep(c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create January dummy
Mar1 <- rep(c(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create March dummy
Apr1 <- rep(c(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create April dummy
May1 <- rep(c(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create May dummy
Jun1 <- rep(c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create June dummy
Jul1 <- rep(c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create Jul dummy
Aug1 <- rep(c(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create Aug dummy
Sep1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0), (length(zz)/12+1))     # Create Sep dummy
Oct1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0), (length(zz)/12+1))     # Create Oct dummy
Nov1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0), (length(zz)/12+1))     # Create Oct dummy
Dec1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0), (length(zz)/12+1))     # Create Oct dummy
seas1 <- cbind(Feb1, Mar1, Apr1, May1, Jun1, Jul1, Aug1, Sep1, Oct1, Nov1, Dec1)
seas <- seas1[1:T,]

x_la_fit_sea <- lm(x_la ~ seas)          # Regress x_la against constant + seasonal dummies
> summary(x_la_fit_sea)
```

Seasonal Time Series – Deterministic

Example (continuation): We define monthly dummies

```
> summary(x_la_fit_sea)
Coefficients:
              Estimate   Std. Error t value Pr(>|t|)
(Intercept)  -0.0014063  0.0020125  -0.699 0.485157
seasFeb1      0.0006752  0.0028223   0.239 0.811079
seasMar1      0.0049095  0.0028223   1.740 0.082838 .
seasApr1      0.0090903  0.0028223   3.221 0.001400 **
seasMay1      0.0104159  0.0028223   3.691 0.000260 ***
seasJun1      0.0103464  0.0028223   3.666 0.000285 ***
seasJul1      0.0080593  0.0028223   2.856 0.004557 **
seasAug1      0.0062247  0.0028223   2.206 0.028080 *
seasSep1      0.0032244  0.0028223   1.142 0.254055
seasOct1      0.0011967  0.0028461   0.420 0.674421
seasNov1     -0.0006218  0.0028461  -0.218 0.827181
seasDec1     -0.0009031  0.0028461  -0.317 0.751195
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note: Returns –i.e., home prices– are higher from April to August.

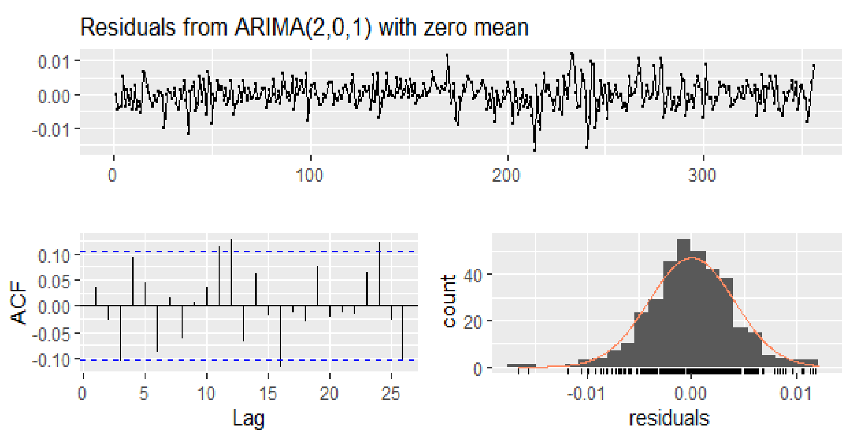
Seasonal Time Series – Deterministic

Example (continuation): Now, we model e_t, the filtered LA series

```
x_la_filt <- x_la_fit_sea$residuals          # residuals, e_t = filtered x_la series
fit_ar_la_filt <- auto.arima(x_la_filt)      # use auto.arima to look for a good model
> fit_ar_la_filt
Series: x_la_filt
ARIMA(2,0,1) with zero mean

Coefficients:
        ar1     ar2     ma1
     0.0987  0.7737  0.7245
s.e. 0.0963  0.0866  0.1136

sigma^2 estimated as 1.668e-05:  log likelihood=1453.66
AIC=-2899.33   AICc=-2899.21   BIC=-2883.83

> checkresiduals(fit_ar_la_filt)

        Ljung-Box test

data:  Residuals from ARIMA(2,0,1) with zero mean
Q* = 13.5, df = 7, p-value = 0.06083      ⇒ Reject H0 at 5% lever. But, judgement call is OK.

Model df: 3.   Total lags used: 10
```

Seasonal Time Series – Deterministic

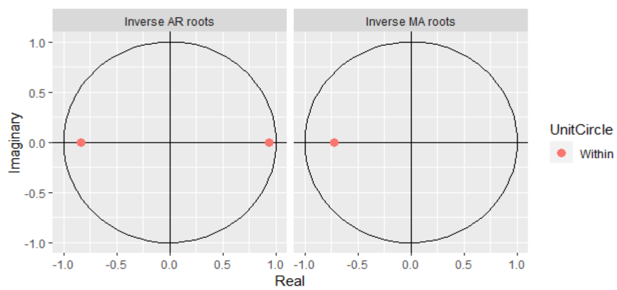
Example (continuation): We check residual plots.



<u>Note</u>: ACF shows some small, but significant autocorrelations, but the seasonal (wave) pattern is no longer there.

Seasonal Time Series – Deterministic

Example (continuation): Finally, we check the stationarity & the invertibility of the ARIMA(2,0,1) process.



Note: All roots inside the unit circle (& real): stationarity and invertibility.

Review: Forecasting

- Forecasting is the primary objective of ARIMA modeling.

- Two types of forecasts.

- **In sample** (prediction): The expected value of the RV (in-sample), the "fitted values," $\hat{Y}_t$.

- **Out of sample** (forecasting): The value of a future RV that is not observed by the sample, $\hat{Y}_{T+\ell}$.

Notation:

- Forecast for $T+\ell$ made at T: $\hat{Y}_{T+\ell}$, $\hat{Y}_{T+\ell|T}$, $\hat{Y}_T(\ell)$.

- $T+\ell$ forecast error: $e_{T+\ell} = e_T(\ell) = Y_{T+\ell} - \hat{Y}_{T+\ell}$

- Mean squared error (MSE): $MSE(e_{T+\ell}) = E[Y_{T+\ell} - \hat{Y}_{T+\ell}]^2$

Review: Forecasting – Basic Concepts

• The optimal point forecast under MSE is the (conditional) mean:

$$\hat{Y}_{T+\ell} = \mathrm{E}[Y_{T+\ell} \,|\, I_T]$$

• Different loss functions lead to different optimal forecast. For example, for the MAE, the optimal point forecast is the median.

• The computation of $\mathrm{E}[Y_{T+\ell} \,|\, I_T]$ depends on the distribution of $\{\varepsilon_t\}$. Then, if

$\{\varepsilon_t\} \sim \mathrm{WN} \quad \Rightarrow \mathrm{E}[\varepsilon_{T+\ell} \,|\, I_T] = 0.$

Review: Forecasting Steps for ARMA Models

• Process:

(1) Find ARIMA model
(Use ACF, PACF or Minic)

$$Y_t = \phi Y_{t-1} + \varepsilon_t$$
$$\Downarrow$$

(2) Estimation
(& Evaluation in-sample)

$$\hat{\phi} \text{ (Estimate of } \phi)$$
$$\Downarrow$$
$$\hat{Y}_t = \hat{\phi} Y_{t-1} \text{ (Prediction)}$$

(3) Forecast
(& Evaluation out-of-sample)

$$\Downarrow$$
$$\hat{Y}_{t+1} = \hat{\phi} \hat{Y}_t \text{ (Forecast)}$$

Review: Forecasting From ARMA Models

• We observe the time series: $\boldsymbol{Y}_T = \{Y_1, Y_2, \ldots, Y_T\}$.
- We determine an ARIMA(p, d, q) model.
- At time T, we want to forecast: $Y_{t+1}, Y_{t+2}, \ldots, Y_{T+\ell}$.
- The information set we have is $I_T = \{Y_1, Y_2, \ldots, Y_T, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T\}$.

• Use the conditional expectation of $Y_{T+\ell}$, given the information at T:

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell} \,|\, Y_T, Y_{T-1}, \ldots, Y_1]$$

Example: We have an AR(1) model.

$$Y_{T+1} = \mu + \phi_1 Y_T + \varepsilon_{T+1}$$

Then, the one-step ahead forecast:

$$\hat{Y}_{T+1} = E[Y_{T+1} \,|\, Y_T, Y_{T-1}, \ldots, Y_1] = \mu + \phi_1 Y_T$$

since $E[\varepsilon_{T+1} \,|\, Y_T, Y_{T-1}, \ldots, Y_1] = 0$.

31

Review: Forecasting From MA(q) Models

• The stationary MA(q) model for Y_t is

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

We produce at time T *l-step ahead* forecasts using:

$$Y_{T+1} = \mu + \varepsilon_{T+1} + \theta_1 \varepsilon_T + \cdots + \theta_q \varepsilon_{T-q+1}$$

$$Y_{T+2} = \mu + \varepsilon_{T+2} + \theta_1 \varepsilon_{T+1} + \cdots + \theta_q \varepsilon_{T-q+2}$$

$$\vdots$$

$$Y_{T+\ell} = \mu + \varepsilon_{T+\ell} + \theta_1 \varepsilon_{T+\ell-1} + \cdots + \theta_q \varepsilon_{T+\ell-q} \qquad (l > 2)$$

Now, we take conditional expectations:

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell} \,|\, I_T] = \mu + E[\varepsilon_{T+\ell} \,|\, I_T] + \theta_1 E[\varepsilon_{T+\ell-1} \,|\, I_T] + \\ + \cdots + \theta_q E[\varepsilon_{T+\ell-q} \,|\, I_T]$$

Note: Forecasts are a linear combination of errors.

Review: Forecasting From MA(q) Models

• Some of the errors are know at T: $\varepsilon_1 = \hat{\boldsymbol{\varepsilon}}_1$, $\varepsilon_2 = \hat{\boldsymbol{\varepsilon}}_2$, ..., $\varepsilon_T = \hat{\boldsymbol{\varepsilon}}_T$, the rest are unknown. Thus,

$$\text{E}[\varepsilon_{T+j}] = 0 \qquad \text{for } j > 1.$$

Example: For an MA(2) we have:

$$\hat{Y}_{T+1} = \mu + \text{E}[\varepsilon_{T+1} | I_T] + \theta_1 \text{E}[\varepsilon_T | I_T] + \theta_2 \text{E}[\varepsilon_{T-1} | I_T]$$
$$\hat{Y}_{T+2} = \mu + \text{E}[\varepsilon_{T+2} | I_T] + \theta_1 \text{E}[\varepsilon_{T+1} | I_T] + \theta_2 \text{E}[\varepsilon_T | I_T]$$
$$\hat{Y}_{T+3} = \mu + \text{E}[\varepsilon_{T+3} | I_T] + \theta_1 \text{E}[\varepsilon_{T+2} | I_T] + \theta_2 \text{E}[\varepsilon_{T+1} | I_T]$$

At time $T = t$, we know ε_t & ε_{t-1}. Set $\text{E}\big[\varepsilon_{t+j} | I_t\big]$=0 for $j > 1$. Then,

$$\hat{Y}_{t+1} = \mu + \theta_1 \text{E}[\varepsilon_t | I_t] + \theta_2 \text{E}[\varepsilon_{t-1} | I_t] = \mu + \theta_1 \hat{\boldsymbol{\varepsilon}}_t + \theta_2 \hat{\boldsymbol{\varepsilon}}_{t-1}$$
$$\hat{Y}_{t+2} = \mu + \theta_2 \text{E}[\varepsilon_t | I_t] = \mu + \theta_2 \hat{\boldsymbol{\varepsilon}}_t$$
$$\hat{Y}_{t+3} = \mu$$
$$\hat{Y}_{t+\ell} = \mu \qquad \text{for } \ell > 2. \quad \Rightarrow \text{MA(2) memory of 2 periods}$$

Review: Forecasting From MA(q) Models

• The example generalizes: An MA(q) process has a memory of only q periods. All forecasts beyond q revert to the unconditional mean, μ.

Example: We fit an MA(1) to the U.S. stock returns (T=1,975):

```
library(tseries)
library(forecast)
fit_p_ts <- arima(lr_p, order=c(0,0,1))          # fit an MA(1) model
fcast_p <- forecast(fit_p_ts, h=4)               # produce 4-step ahead forecasts
> fit_p_ts
> fcast_p
Coefficients:
        ma1  intercept
     0.2888     0.0037
s.e. 0.0218     0.0012

sigma^2 estimated as 0.001522:  log likelihood = 3275.83,  aic = -6545.67
> fcast_p
    Point Forecast       Lo 80      Hi 80       Lo 95      Hi 95
1796   0.012570813 -0.03742238 0.06256401 -0.06388718 0.08902881
1797   0.003689524 -0.04834634 0.05572539 -0.07589247 0.08327152
1798   0.003689524 -0.04834634 0.05572539 -0.07589247 0.08327152
1799   0.003689524 -0.04834634 0.05572539 -0.07589247 0.08327152
```

Review: Forecasting From AR(p) Models

- The stationary AR(p) model for Y_t is

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

We produce, at time T, ℓ-*step ahead* forecasts using:

$$Y_{T+1} = \mu + \phi_1 Y_T + \phi_2 Y_{T-1} + \cdots + \phi_p Y_{T-p+1} + \varepsilon_{T+1}$$

$$Y_{T+2} = \mu + \phi_1 Y_{T+1} + \phi_2 Y_T + \cdots + \phi_p Y_{T-p+2} + \varepsilon_{T+2}$$

$$\vdots$$

$$Y_{t+\ell} = \mu + \phi_1 Y_{T+\ell-1} + \phi_2 Y_{T+\ell-2} + \cdots + \phi_p Y_{T+\ell-p} + \varepsilon_{t+\ell} \quad (\ell>2)$$

Now, we take conditional expectations:

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell} \mid I_T] = \mu + \phi_1 E[Y_{T+\ell-1} \mid I_T] + \phi_2 E[Y_{T+\ell-2} \mid I_T] + \\ + \cdots + \phi_p E[Y_{T+\ell-p} \mid I_T]$$

Note: The forecasts $\hat{Y}_{T+\ell}$ is a linear combination of past forecast.

Review: Forecasting From AR(p) Models

Example: AR(2) model for $Y_{t+\ell}$ is

$$Y_{t+\ell} = \mu + \phi_1 Y_{t+\ell-1} + \phi_2 Y_{t+\ell-2} + \varepsilon_{t+\ell}$$

Then, taking conditional expectations at $T = t$, we get the forecasts:

$$\hat{Y}_{t+1} = \mu + \phi_1 Y_t + \phi_2 Y_{t-1}$$

$$\hat{Y}_{t+2} = \mu + \phi_1 \hat{Y}_{t+1} + \phi_2 Y_t$$

$$\hat{Y}_{t+3} = \mu + \phi_1 \hat{Y}_{t+2} + \phi_2 \hat{Y}_{t+1}$$

$$\vdots$$

$$\hat{Y}_{t+\ell} = \mu + \phi_1 \hat{Y}_{t+\ell-1} + \phi_2 \hat{Y}_{T+\ell-2}$$

- AR-based forecasts are autocorrelated, they have long memory!

Review: Forecasting From AR(p) Models

Example: We fit an AR(4) to the changes in Oil Prices (T=346):

```
fit_oil_ts <- arima(lr_oil, order=c(4,0,0))
fcast_oil <- forecast(fit_oil_ts, h=12)
> fit_oil_ts

Coefficients:
        ar1      ar2      ar3      ar4    intercept
     0.2946  -0.1027  -0.0571  -0.0983    0.0017
s.e. 0.0521   0.0543   0.0551   0.0539    0.0051

sigma^2 estimated as 0.008812:  log likelihood = 344.57,  aic = -677.14

> fcast_oil
    Point Forecast      Lo 80      Hi 80      Lo 95     Hi 95
365  -5.425015e-02 -0.1745546 0.0660543 -0.2382399 0.1297396
366  -1.578754e-02 -0.1412048 0.1096297 -0.2075966 0.1760216
367   2.455760e-03 -0.1229760 0.1278875 -0.1893755 0.1942871
368   1.356917e-02 -0.1123501 0.1394884 -0.1790077 0.2061460
369   1.160479e-02 -0.1154462 0.1386558 -0.1827029 0.2059125
370   5.060891e-03 -0.1221954 0.1323172 -0.1895608 0.1996826
371   9.059104e-04 -0.1263511 0.1281629 -0.1937169 0.1955287
```

Note: You can extract the point forecasts from the forecast function using $mean. That is, fcast_oil$mean extracts the whole vector of forecasts.

Review: Forecasting From ARMA Models

- The stationary ARMA model for Y_t is

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

- We produce at time T the forecast $Y_{T+\ell}$. Then,

$$Y_{T+\ell} = \theta_0 + \phi_1 Y_{T+\ell-1} + \cdots + \phi_p Y_{T+\ell-p} + \varepsilon_{T+\ell} + \theta_1 \varepsilon_{T+\ell-1} + \cdots + \theta_q \varepsilon_{T+\ell-q}$$

- Taking conditional expectations:

$$\hat{Y}_{T+\ell} = \theta_0 + \phi_1 \hat{Y}_{T+\ell-1} + \cdots + \phi_p \hat{Y}_{T+\ell-p} + E[\varepsilon_{T+\ell}|I_T] + \theta_1 \mathrm{E}[\varepsilon_{T+\ell-1}\,|\,I_T] + \cdots + \\ + \theta_q \mathrm{E}\left[\varepsilon_{T+\ell-q}\,|\,I_T\right]$$

- An ARMA forecasting is a combination of past $\hat{Y}_{T+\ell-i}$ forecasts and observed past $\hat{\varepsilon}_{t+\ell-i}$.

Review: Forecasting From ARMA Models

• We use the MA(∞) (Wold) – representation of a stationary ARMA process to get the forecast error. The Wold representation:

$$Y_{T+\ell} = \mu + \varepsilon_{T+\ell} + \Psi_1\varepsilon_{T+\ell-1} + \Psi_2\varepsilon_{T+\ell-2} + \cdots + \Psi_\ell\varepsilon_T + \cdots$$

The forecast error is:

$$e_T(\ell) = Y_{T+\ell} - \hat{Y}_{T+\ell} = \sum_{i=0}^{\ell-1} \Psi_i\varepsilon_{T+\ell-i}$$

Example 2: One-step ahead forecast (ℓ = 2).

$$Y_{T+2} = \mu + \varepsilon_{T+2} + \Psi_1\varepsilon_{T+1} + \Psi_2\varepsilon_T + \Psi_3\varepsilon_{T-1} + \cdots$$
$$\hat{Y}_{T+2} = \mu + \Psi_2\varepsilon_T + \Psi_3\varepsilon_{T-1} + \cdots$$
$$e_T(2) = Y_{T+2} - \hat{Y}_{T+2} = \varepsilon_{T+2} + \Psi_1\varepsilon_{T+1}$$
$$Var\big(e_T(2)\big) = \sigma^2 * (1 + \Psi_1^2)$$

Note: $\lim_{\ell\to\infty} \hat{Y}_T(\ell) = \mu$

Review: Forecasting From ARMA Models

• The forecast error is:

$$e_T(\ell) = \sum_{i=0}^{\ell-1} \Psi_i\varepsilon_{T+\ell-i}$$

Note: When the expectation of the forecast error is zero:

E[$e_T(\ell)$] = 0 ⇒ we say the forecast is **unbiased**.

• The variance of the forecast error:

$$Var\big(e_T(\ell)\big) = Var\left(\sum_{i=0}^{\ell-1} \Psi_i\varepsilon_{T+\ell-i}\right) = \sigma^2 \sum_{i=0}^{\ell-1} \Psi_i^2$$

Note: $\lim_{\ell\to\infty} Var[e_T(\ell)] = \gamma_0 < \infty$

Review: Forecasting From ARMA Models

• The Wold representation depends on an infinite number of parameters, but, in practice, they decay rapidly. Then, as we forecast into the future, the forecasts are not very interesting:

$$\lim_{\ell \to \infty} \hat{Y}_T(\ell) = \mu$$

We have unconditional forecasts.

• This is why ARIMA forecasting is useful only for short-term.

Review: Forecasting From ARMA Models: C.I.

• A 100(1- α)% prediction interval for $Y_{T+\ell}$ (ℓ-steps ahead) is

$$\hat{Y}_T(\ell) \pm z_{\alpha/2} \sqrt{Var(e_T(\ell))}$$

or, $$\hat{Y}_T(\ell) \pm z_{\alpha/2}\, \sigma \sqrt{\sum_{i=0}^{\ell-1} \Psi_i^2}$$

Example: 95% C.I. for the 2-step-ahead forecast:

$$\hat{Y}_T(2) \pm 1.96\, \sigma \sqrt{1 + \Psi_1^2}$$

• When computing prediction intervals from data, we substitute estimates for parameters, giving approximate prediction intervals.

Note: MSE[$\varepsilon_{T+\ell}$] = MSE[$e_{T+\ell}$] = $\sigma^2 \sum_{i=0}^{\ell-1} \Psi_i^2$

Forecasting From Simple Models: ES

• Industrial companies, with a lot of inputs and outputs, want quick and inexpensive forecasts. Easy to fully automate. In general, we use past Y_t to forecast future Y_t's, usually referred as the *level's forecasts*.

• Exponential Smoothing Models (ES) fulfill these requirements.

• In general, these models are limited and not optimal, especially compared with Box-Jenkins methods.

• Goal of these models: Suppress the short-run fluctuation by smoothing the series. For this purpose, a weighted average of all previous values works well.

• There are many ES models. We will go over the Simple Exponential Smoothing (SES) & Holt-Winter's Exponential Smoothing (HW ES).

Simple Exponential Smoothing: SES

• We "*smooth*" the series Y_t to produce a quick forecast, S_{t+1}, also called *level's forecast*. Smooth? The graph of S_t is less jagged than the graph of the original series, Y_t.

• We use the observed time series at time t: $Y_1, Y_2, \ldots, Y_t$.

• The equation for the **level**: $S_t = \alpha Y_{t-1} + (1-\alpha) S_{t-1}$

where

- α: The smoothing parameter, $0 \leq \alpha \leq 1$.
- Y_t: Value of the observation at time t.
- S_t: Value of the smoothed observation at time t –i.e., the forecast.

• The equation can also be written as an *updating equation*:

$$S_t = S_{t-1} + \alpha(Y_{t-1} - S_{t-1}) = S_{t-1} + \alpha * (\text{past forecast error})$$

SES: Forecast and Updating

- From the updating equation for S_t:

$$S_t = S_{t-1} + \alpha(Y_{t-1} - S_{t-1})$$

today, at time t, we compute the forecast:

$$S_{t+1} = S_t + \alpha(Y_t - S_t)$$

That is, a simple updating forecast: last period forecast + adjustment.

For the next period, $t+2$, we have (since $S_{t+1} = Y_{t+1}$)

$$S_{t+2} = S_{t+1} + \alpha(Y_{t+1} - S_{t+1}) = S_{t+1}$$

Then, at time t, the ℓ-step ahead forecast is:

$S_{t+\ell} = S_{t+1}$ ⇒ A naive forecast!

Note: SES forecasts are not very interesting after $\ell > 1$.

SES: Forecast and Updating

Example: An industrial firm uses SES to forecast sales:

$$S_{t+1} = S_t + \alpha * (Y_t - S_t)$$

The firm estimates α = **0.25**. The firm observes Y_t = **5** and, last period's forecast, S_t = **3**.

Then, the forecast for time $t+1$ is:

S_{t+1} = **3** + **0.25** * (**5** – **3**) = **3.50**

The forecast for time $t+1$ (& any period after time $t+1$) is:

$S_{t+\ell} = S_{t+1}$ = **3.50** for $\ell > 1$.

Later, the firm observes: Y_{t+1} = **4.77**, Y_{t+2} = **3.15**, & Y_{t+3} = **1.85**.
Then, the MSE:

MSE = $\frac{1}{3}$ * [(**4.77** – **3.50**)^2 + (**3.15** - **3.50**)^2 + (**1.85** - **3.50**)^2] = **1.486**.

SES: Forecast and Updating

Example (continuation):

Note: If α = 0.75, then

S_{t+1} = 3 + 0.75 * (5 – 3) = 4.50

A bigger α gives more weight to the more recent observation –i.e., Y_t.

Again, the forecast for time *t*+1 and any period after time *t*+1 is:

$S_{t+\ell} = S_{t+1} = 4.50$ for ℓ > 1.

SES: Exponential?

- Q: Why Exponential?

For the observed time series $\{Y_1, Y_2, \ldots, Y_t, Y_{t+1}\}$, using backward substitution, $S_{t+1} = \hat{Y}_t(1)$ can be expressed as a weighted sum of previous observations:

$$S_{t+1} = \alpha Y_t + (1-\alpha)S_t = \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha)S_{t-1}]$$
$$= \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + (1-\alpha)^2 S_{t-1}$$
$$\Rightarrow \hat{Y}_t(1) = S_{t+1} = c_0 Y_t + c_1 Y_{t-1} + c_2 Y_{t-2} + \cdots$$

where c_i's are the weights, with

$$c_i = \alpha(1-\alpha)^i;\ i = 0, 1, \quad \ldots;\ 0 \leq \alpha \leq 1.$$

- We have decreasing weights, by a constant ratio for every unit increase in lag.

Then, $\hat{Y}_t(1) = \alpha(1-\alpha)^0 Y_t + \alpha(1-\alpha)^1 Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \cdots$

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

48

SES: Exponential Weights

- $c_i = \alpha\,(1-\alpha)^i$; $i = 0, 1, \ldots;\ 0 \le \alpha \le 1.$

$c_i = \alpha(1-\alpha)^i$	α = 0.25	α = 0.75
c_0	0.25	0.75
c_1	0.25 * 0.75 = 0.1875	0.75 * 0.25 = 0.1875
c_2	.25 * 0.75^2 = 0.140625	0.75 * 0.25^2 = 0.046875
c_3	.25 * 0.75^3 = 0.1054688	0.75 * 0.25^3 = 0.01171875
c_4	.25 * 0.75^4 = 0.07910156	0.75 * 0.25^4 = 0.002929688
⋮		
c_{12}	.25 * 0.75^{12} = 0.007919088	0.75 * 0.25^{12} = 4.470348e-08

Decaying weights. Faster decay with greater α, associated with faster learning: we give more weight to more recent observations.

- We do not know α; we need to estimate it.

49

SES: Selecting α

- Choose α between 0 and 1.

- If α = 1, it becomes a naive model; if α ≈ 1, more weights are put on recent values. The model fully utilizes forecast errors.

- If α is close to 0, distant values are given weights comparable to recent values. Set α ≈ 0 when there are big random variations in Y_t.

- α is often selected as to minimize the MSE.

- In empirical work, 0.05 ≤ α ≤ 0.3 are used (α ≈ 1 is used rarely).

Numerical Minimization Process:

- Take different α values ranging between 0 and 1.

- Calculate 1-step-ahead forecast errors for each α.

- Calculate MSE for each case.

Choose α which has the min MSE: $e_t = Y_t - S_t \Rightarrow \min \sum_{t=1}^{n} e_t^2 \Rightarrow \alpha$

50

SES: Selecting α – MSE

$$S_{t+1} = \alpha Y_t + (1-\alpha)S_t$$

Time	Y_t	S_{t+1} (α=0.10)	$(Y_t - S_t)^2$
1	5	-	-
2	7	(0.1)5 +(0.9)5 = 5	4
3	6	(0.1)7 + (0.9)5 = 5.2	0.64
4	3	(0.1)6 + (0.9)5.2 = 5.28	5.1984
5	4	(0.1)3 + (0.9)5.28 = 5.052	1.107
		TOTAL	**10.945**

$$MSE = \frac{SSE}{n-1} = 2.74$$

• Calculate this for α = 0.2, 0.3,…, 0.9, 1 and compare the MSEs. Choose α with minimum MSE.

Note: $Y_{t-1} = 5$ is set as the initial value for the recursive equation.

51

SES: Initial Values

• We have a recursive equation, we need initial values, S_1 (or Y_0).

• Approaches:

– Set S_1 equal to Y_1. Then, $S_2 = Y_1$.

– Take the average of, say first 4 or 5 observations. Use this average as an initial value.

– Estimate S_1 (similar to the estimation of α.)

52

SES: Forecasting Examples

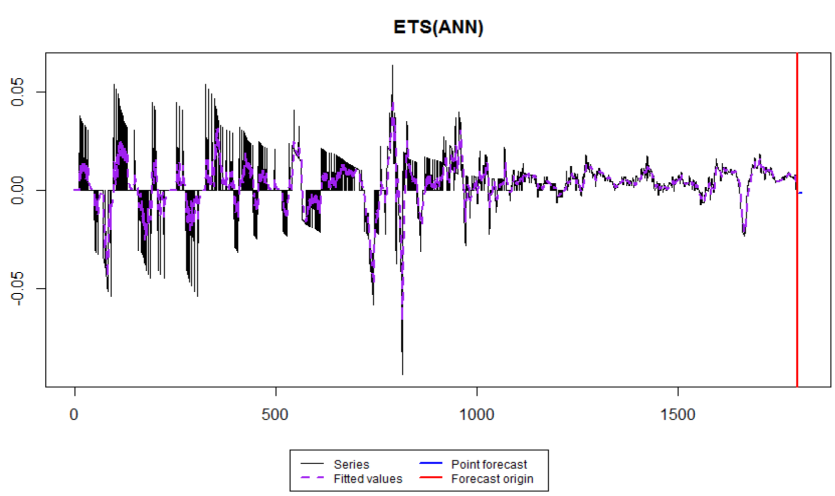
Example 1: We want to forecast log changes in **U.S. monthly dividends** (T=1796) using SES. First, we estimate the model using the R function *HoltWinters*(), which has as a special case SES: set beta=FALSE, gamma=FALSE. We use estimation period T=1750.

```
mod1 <- HoltWinters(lr_d[1:1750], beta=FALSE, gamma=FALSE)
> mod1
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = lr_d[1:1750], beta = FALSE, gamma = FALSE)

Smoothing parameters:
 alpha: 0.289268                          ⇒ Estimated α
 beta : FALSE
 gamma: FALSE

Coefficients:
        [,1]
a 0.004666795                             ⇒ Forecast
```

53

SES: Forecasting Examples

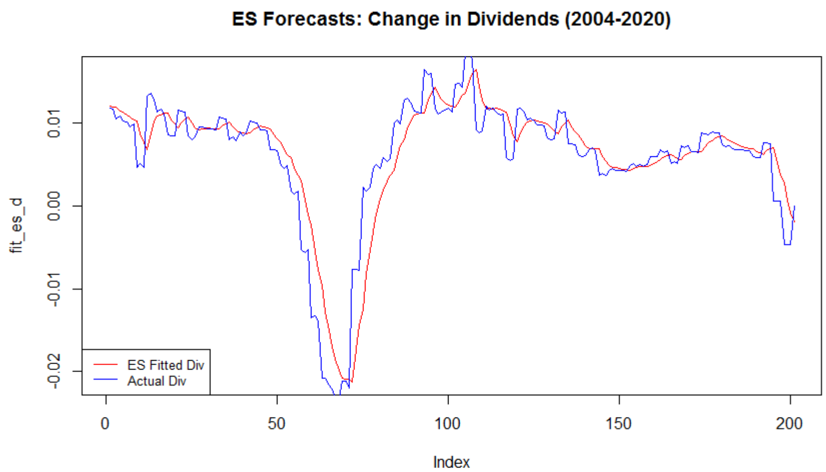
Example 1 (continuation):



54

SES: Forecasting Examples

Example 1 (continuation):



55

SES: Forecasting Examples

Example 1 (continuation): Now, we do one-step ahead forecasts

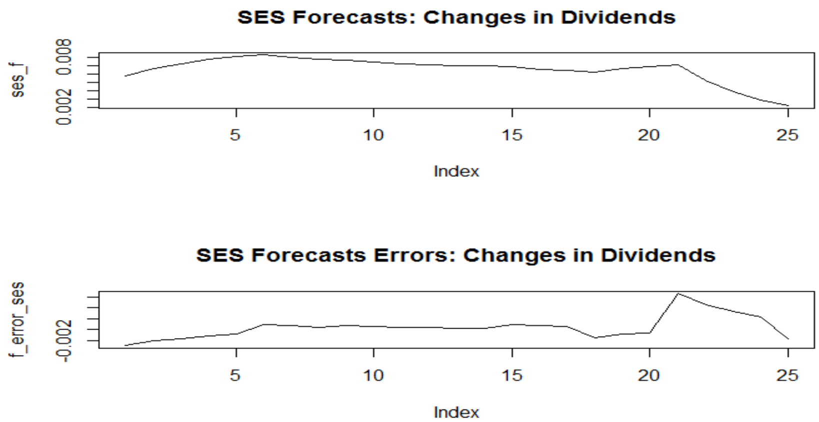
```
T_last <- nrow(mod1$fitted)                    # number of in-sample forecasts
h <- 25                                        # forecast horizon
ses_f <- matrix(0,h,1)                         # Vector to collect forecasts
alpha <- 0.29
y <- lr_d
T <- length(lr_d)
sm <- matrix(0,T,1)
T1 <- T – h + 1                                # Start of forecasts
a <- T1                                        # index for while loop
sm[a-1] <- mod1$fitted[T_last]                 # last in-sample forecast
while (a <= T) {
        sm[a] = alpha * y[a-1] + (1-alpha) * sm[a-1]
a <- a + 1
}

ses_f <- sm[T1:T]
ses_f
f_error_ses <- sm[T1:T] - y[T1:T]              # forecast errors
MSE_ses <- sum(f_error_ses^2)/h                # MSE
plot(ses_f, type="l", main ="SES Forecasts: Changes in Dividends")
```

56

SES: Forecasting Examples

Example 1 (continuation):

```
> ses_f
f_error_ses <- sm[T1:T] - y[T1:T]
> plot(ses_f, type="l", main ="SES Forecasts: Changes in Dividends")
```



SES: Forecasting U.S. Dividends

Example 1 (continuation): *h-step-ahead* forecasts

> forecast(mod1, h=25, level=.95)

	Point Forecast	Lo 95	Hi 95
1751	0.004666795	-0.01739204	0.02672563
1752	0.004666795	-0.01829640	0.02762999
1753	0.004666795	-0.01916647	0.02850006
1754	0.004666795	-0.02000587	0.02933947
1755	0.004666795	-0.02081765	0.03015124
1756	0.004666795	-0.02160435	0.03093794
1757	0.004666795	-0.02236816	0.03170175
1758	0.004666795	-0.02311098	0.03244457
1759	0.004666795	-0.02383445	0.03316804
1760	0.004666795	-0.02454001	0.03387360
1761	0.004666795	-0.02522891	0.03456250
1762	0.004666795	-0.02590230	0.03523589
1763	0.004666795	-0.02656117	0.03589476
1764	0.004666795	-0.02720642	0.03654001

...

Note: Constant forecasts, but C.I. gets wider (as expected) with h. 58

SES: Forecasting Examples

Example 2: We want to forecast **log monthly U.S. vehicles** (1976-2020, T=537) using SES.

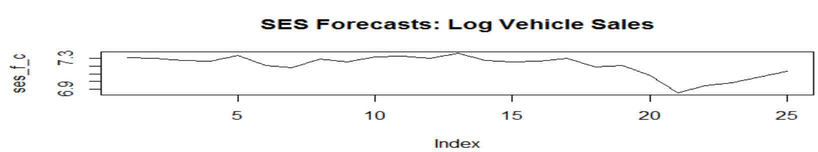
```
mod_car <- HoltWinters(l_car[1:512], beta=FALSE, gamma=FALSE)
> mod_car
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = l_car[1:512], beta = FALSE, gamma = FALSE)

Smoothing parameters:
 alpha: 0.4888382                          ⇒ Estimated α
 beta : FALSE
 gamma: FALSE

Coefficients:
     [,1]
a 7.315328
```

59

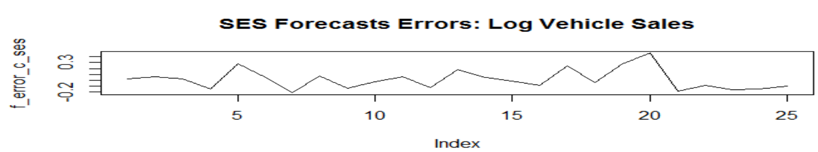
SES: Forecasting Examples

Example 2 (continuation): Now, we do one-step ahead forecasting

```
ses_f_c <- sm_c[T1:T]
f_error_c_ses <- sm_c[T1:T] - y[T1:T]
> plot(ses_f_c, type="l", main ="SES Forecasts: Log Vehicle Sales")
```




```
> plot(f_error_c_ses, type="l", main ="SES Forecasts Errors: Log Vehicle Sales")
```




```
MSE_ses <- sum(f_error_c_ses^2)/h
> MSE_ses
[1] 0.027889
```

60

SES: Remarks

• Some computer programs automatically select the optimal α using a line search method or non-linear optimization techniques.

• We have a recursive equation, we need initial values for S_1.

• This model ignores trends or seasonalities. Not very realistic, especially for manufacturing facilities, retail sector, and warehouses.

• Deterministic components, D_t, can be easily incorporated.

• The model that incorporates both a trend and seasonal features is called *Holt-Winter's ES*.

61

Holt-Winters (HW) Exponential Smoothing

• In the model for Y_t, in addition to the level (S_t), we introduce **trend** (T_t) & **seasonality** (I_t) factors. Since we produce smooth forecasts for T_t & I_t, this method is also called *triple exponential smoothing*.

• The h-step ahead forecast is a combination of the smooth forecasts of S_t (Level), T_t (Trend) & I_{t+h-s} (Seasonal).

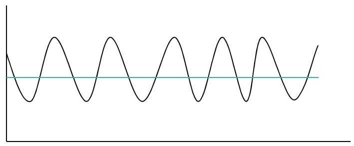
• Both, T_t & I_t, can be included as *additively* or *multiplicatively* factors. In this class, we consider an additive trend and the seasonal factor as additive or multiplicative. We produce *h*-step ahead forecasts:

- For the additive model: $\hat{Y}_t(h) = S_t + h\,T_t + I_{t+h-s}$
- For the multiplicative model: $\hat{Y}_t(h) = (S_t + h\,T_t) * I_{t+h-s}$

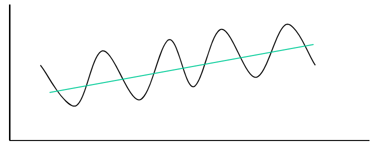
Note: Seasonal factor is multiplied in the h-step ahead forecast.

62

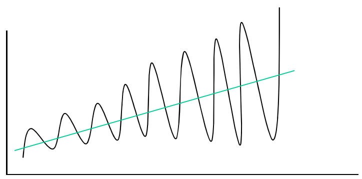
Holt-Winters (HW) ES: Trend & Seasonality



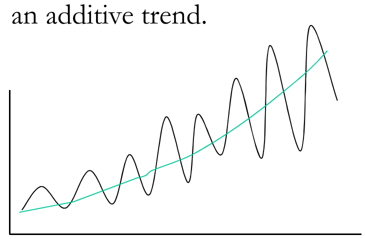
1. No trend and additive seasonal variability.



2. Additive seasonal variability with an additive trend.



3. Multiplicative seasonal variability with an additive trend.



4. Multiplicative seasonal variability with a multiplicative trend.

Note: We will use Model 2 (Additive) and Model 3 (Multiplicative).

Holt-Winters (HW) ES: Additive

• Additive model (additive trend & additive seasonality) forecast:

$$\hat{Y}_t(h) = S_t + h\,T_t + I_{t+h-s}$$

where s is the number of periods in seasonal cycles (=4 for quarters).

• Components:

- **The level**, S_t: A weighted average of "*seasonal adjusted*" Y_t (=$Y_t - I_{t-s}$), and the non-seasonal forecast $(S_{t-1} + T_{t-1})$:

$$S_t = \alpha(Y_t - I_{t-s}) + (1-\alpha)(S_{t-1} + T_{t-1})$$

- **The trend**, T_t: A weighted average of T_{t-1} and the change in S_t.

$$T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1}$$

- **The seasonality**, I_t: A weighted average of seasonal index of s last year, I_{t-s}, and the current seasonal index $(Y_{t-1} - S_{t-1} - T_{t-1})$:

$$I_t = \gamma(Y_t - S_{t-1} - T_{t-1}) + (1-\gamma)I_{t-s}$$

64

Holt-Winters (HW) ES: Additive

- Then, the model for the h-step ahead forecast

$$\hat{Y}_t(h) = S_t + h\,T_t + I_{t+h-s}$$

has three equations:

Level: $S_t = \alpha(Y_t - I_{t-s}) + (1-\alpha)(S_{t-1} + T_{t-1})$

Trend: $T_t = \beta\,(S_t - S_{t-1}) + (1-\beta)\,T_{t-1}$

Seasonal: $I_t = \gamma(Y_t - S_{t-1} - T_{t-1}) + (1-\gamma)I_{t-s}$

- We have only three smoothing parameters:

  α = level coefficient
  β = trend coefficient
  γ = seasonality coefficient

65

Holt-Winters (HW) ES: Multiplicative

- In the multiplicative seasonal case (with an additive trend), we have the h-step ahead forecast:

$$\hat{Y}_t(h) = (S_t + h\,T_t) * I_{t+h-s}$$

- Details for *multiplicative* seasonality –i.e., Y_t/I_t– and *additive* trend

- The forecast, S_t, now shows the average Y_t adjusted ($\frac{Y_t}{I_{t-s}}$).
- The trend, T_t, is a weighted average of T_{t-1} and the change in S_t.
- The seasonality is also a weighted average of I_{t-s} and the Y_t/S_t.

- Then, the model has three equations:

$$S_t = \alpha\,\frac{Y_t}{I_{t-s}} + (1-\alpha)\,(S_{t-1} + T_{t-1})$$
$$T_t = \beta\,(S_t - S_{t-1}) + (1-\beta)\,T_{t-1}$$
$$I_t = \gamma\,\frac{Y_t}{S_t} + (1-\gamma)\,I_{t-s}$$

66

Holt-Winters (HW) ES: Multiplicative

• We think of (Y_t/S_t) as capturing *seasonal effects*.
s = # of periods in the seasonal cycles
(s = 4, for quarterly data; s = 12, for monthly)

• Again, we have only three parameters:
α = smoothing parameter
β = trend coefficient
γ = seasonality coefficient

• Q: How do we determine these 3 parameters?
- Ad-hoc method: α, β and γ can be chosen as values between
$0.02 < \alpha, \gamma, \beta < 0.2$
- Optimal method: Minimization of the MSE, as in SES.

67

Holt-Winters (HW) ES: Multiplicative

Example: An industrial firm uses HW ES to forecast sales next two quarters (h = 1, 2, & 3; with s = 4):

$$\hat{Y}_t(h) = \hat{Y}_{t+h} = (S_t + h\, T_t) * I_{t+h-s}$$

with S_t, T_t, & I_t factors given by:

$$S_t = \alpha \frac{Y_t}{I_{t-s}} + (1-\alpha)(S_{t-1} + T_{t-1})$$
$$T_t = \beta (S_t - S_{t-1}) + (1-\beta) T_{t-1}$$
$$I_t = \gamma \frac{Y_t}{S_t} + (1-\gamma) I_{t-s}$$

The firm estimates: α = 0.25; β = 0.1; & γ = 0.4. It observes Y_t = 5; last quarter's smoothed forecasts: S_{t-1} = 3, T_{t-1} = 1.2; & last year's seasonal factors: I_{t-4} = 1.1, I_{t-3} = 0.7, I_{t-2} = 1.2, & I_{t-3} = 0.8.

• Components forecasts:

$$S_t = 0.25 \frac{5}{1.1} + (1-0.25) * (3+1.3) = 4.2864$$

Holt-Winters (HW) ES: Multiplicative

Example (continuation):

$$S_t = 0.25 * \frac{5}{1.1} + (1 - 0.25) * (3 + 1.2) = 4.2864$$

$$T_t = 0.1 * (4.2864 - 3) + (1 - 0.1) * 1.2 = 1.2086$$

$$I_t = 0.4 * \frac{5}{4.2864} + (1 - 0.4) * 1.1 = 1.1266$$

The forecast for h = 1 (next quarter) is:

$$\hat{Y}_{t+1} = (4.2864 + 1.2086) * 0.7 = 4.8125$$

The forecast for h = 2 & 3 are:

$$\hat{Y}_{t+2} = (4.2864 \ + 2 \ * 1.2086) * 1.2 = 7.8475.$$

$$\hat{Y}_{t+3} = (4.2864 \ + 3 \ * 1.2086) * 0.8 = 6.1329.$$

HW ES: Initial Values

• Initial values for algorithm

- We need at least one complete season of data to determine the initial estimates of I_{t-s}.
- Initial values for *multiplicative* model:

$$S_0 = \sum_{t=1}^{s} Y_t \,/s$$

$$T_0 = \frac{1}{s}\left(\frac{Y_{s+1} - Y_1}{s} + \frac{Y_{s+2} - Y_2}{s} + \cdots + \frac{Y_{s+s} - Y_s}{s}\right)$$

$$\text{or } T_0 = \left[\left\{\sum_{t=1}^{s} Y_t/s\right\} - \left\{\sum_{t=s+1}^{2s} Y_t/s\right\}\right]/s$$

70

HW ES: Initial Values

• Algorithm to compute initial values for seasonal component I_s. Assume we have T observation and quarterly seasonality (s=4):

(1) Compute the averages of each of T years.

$$A_t = \sum_{i=1}^{4} Y_{t,i}/4\,, \quad t = 1, 2, \cdots, 6 \quad \text{(yearly averages)}$$

(2) Divide the observations by the appropriate yearly mean: $Y_{t,i}/A_t$.

(3) I_s is formed by computing the average $Y_{t,i}/A_t$ per year:

$$I_s = \sum_{i=1}^{T} Y_{t,s}/A_t \quad s = 1, 2, 3, 4$$

71

HW ES: Damped Model

• We can damp the trend as the forecast horizon increases, using a parameter ϕ. For the multiplicative model we have:

$$S_t = \alpha \frac{Y_t}{I_{t-s}} + (1-\alpha)(S_{t-1} - \phi\, T_{t-1})$$
$$T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1}$$
$$I_t = \gamma \frac{Y_t}{S_t} + (1-\gamma)I_{t-s}$$

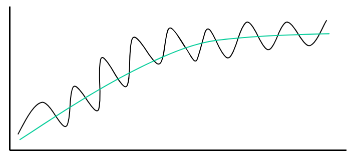
• h-*step ahead* forecast:

$$\hat{Y}_t(h) = \{S_t + \left(1 + \phi + \phi^2 + \cdots + \phi^{2h-1}\right)T_t\} * I_{t+h-s}$$

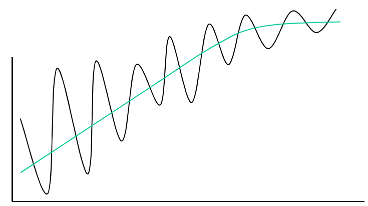
• This model is based on practice: It seems to work well for industrial outputs. Not a lot of theory or clear justification behind the damped trend.

72

ES Models – Damped Model: Types



5. Dampened trend with additive seasonal variability.



6. Multiplicative seasonal variability and dampened trend.

• Overall, we have different models, incorporating different features:
- Trend: Additive or multiplicative, dampened or not
- Seasonal variability: Additive or multiplicative

• Q: With all these models, which one we should use? It depends on the data at hand.

HW ES: Example – Log U.S. Vehicles Sales

Example: We want to forecast log U.S. monthly vehicle sales with HW. We use the R function *HoltWinters()*.

```
l_car_18 <- l_car[1:512]
l_car_ts <- ts(l_car_18, start = c(1976, 1), frequency = 12)     # convert lr_d in a ts object
hw_d_car <- HoltWinters(l_car_18, seasonal="additive")
> hw_d_car
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = lr_d_ts, seasonal = "additive")

Smoothing parameters:
 alpha: 0.4355244                ⇒ Estimated smoothing parameter
 beta : 0.009373815              ⇒ Estimated trend parameter ≈ 0 (no trend)
 gamma:0.3446495                 ⇒ Estimated seasonal parameter
```

74

HW ES: Example – Log U.S. Vehicles Sales

Example (continuation):

> hw_d_car

Coefficients:

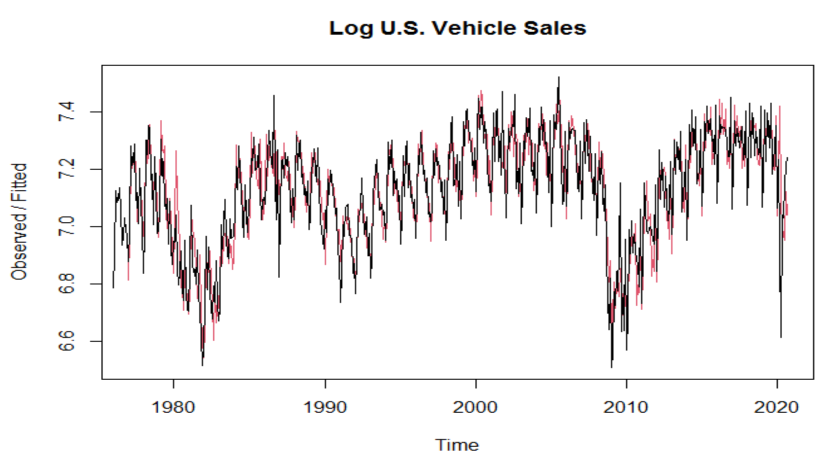
	[,1]	
a	7.177857555	⇒ forecast for level
b	0.0001100345	⇒ forecast for trend
s1	-0.075314457	⇒ forecast for seasonal month 1
s2	-0.084468361	⇒ forecast for seasonal month 2
s3	0.049447067	
s4	-0.273299309	
s5	-0.138251757	
s6	-0.026603921	
s7	-0.144953062	
s8	0.079214066	
s9	0.037899454	
s10	0.020477134	
s11	0.089309775	
s12	-0.012530316	

75

HW ES: Example – Log U.S. Vehicles Sales

Example (continuation):

plot(hw_d_car)



SES: Forecasting Log U.S. Vehicles Sales

Example (continuation): Now, we forecast one-step ahead forecasts

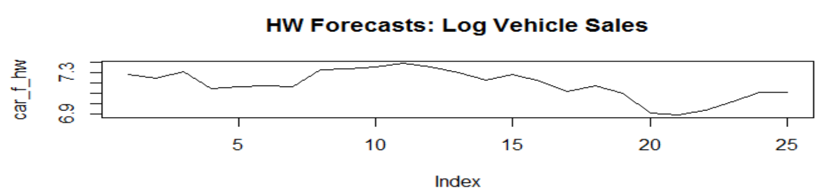
```
T_last <- nrow(hw_d_car$fitted)
h <- 25
ses_f_hw <- matrix(0,h,1)
 alpha <- 0.4355244
 beta <- 0.009373815
 gamma <- 0.3446495
y <- l_car
T <- length(l_car)
sm <- matrix(0,T,1)
Tr <- matrix(0,T,1)
I <- matrix(0,T,1)
T1 <- T-h+1
a <- T1
sm[a-1] <- 7.177857555
Tr[a-1] <- -0.000309358
I[501:512] <- c(-0.075314457,-0.084468361,0.049447067,-0.273299309,-0.138251757, -
0.026603921, -0.144953062,0.079214066,0.037899454,0.020477134,0.089309775,-
0.012530316)
```

77

SES: Forecasting Log U.S. Vehicles Sales

Example (continuation):

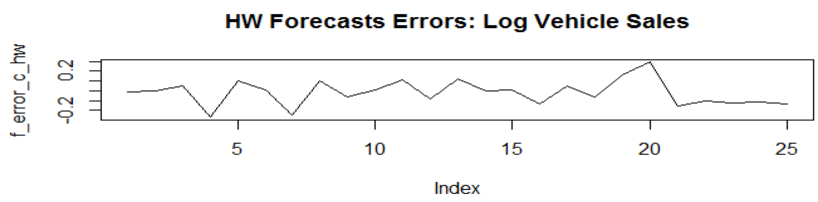
```
while (a <= T) {
        sm[a] = alpha * y[a-1] +  (1-alpha) * sm[a-1]
        Tr[a] = beta * (sm[a] - sm[a-1]) + (1 - beta) * Tr[a-1]
        I[a] = gamma * (y[a] - sm[a]) + (1 - gamma) * I[a - 12]
a <- a + 1
}
hh <- c(1:h)
car_f_hw <- sm[T1:T] + hh*Tr[T1:T] + I[T1:T]
car_f_hw
f_error_c_hw<- car_f_hw - y[T1:T]
plot(car_f_hw, type="l", main ="SES Forecasts: Log Vehicle Sales")
```



SES: Forecasting Log U.S. Vehicles Sales

Example (continuation):

plot(f_error_c_hw, type="l", main ="SES Forecasts Errors: Log Vehicle Sales")




```
MSE_hw <- sum(f_error_c_hw^2)/h
> MSE_hw
[1] 0.01655964
```

79

HW ES: Remarks

- Remarks

- If a computer program selects $\gamma = 0 = \beta$, it has a lack of trend or seasonality. It implies a constant (deterministic) component. In this case, an ARIMA model with deterministic trend may be a more appropriate model.

- For HW ES, a seasonal weight near one implies that a non-seasonal model may be more appropriate.

- We can model seasonalities as multiplicative or additive:

⇒ Multiplicative seasonality: $\text{Forecast}_t = S_t * I_{t-s}$.

⇒ Additive seasonality: $\text{Forecast}_t = S_t + I_{t-s}$.

80

Evaluation of forecasts – Accuracy measures

• The mean squared error (*MSE*) and mean absolute error (*MAE*) are the most popular accuracy measures:

$$\text{MSE} = \frac{1}{m}\sum_{i=T+1}^{T+m}(\hat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=T+1}^{T+m} e_i{}^2$$

$$\text{MAE} = \frac{1}{m}\sum_{i=T+1}^{T+m}|\hat{y}_i - y_i| = \frac{1}{m}\sum_{i=T+1}^{T+m}|e_i|$$

where m is the number of out-of-sample forecasts.

• But other measures are routinely used:

- Mean absolute percentage error (*MAPE*) = $\frac{100}{T-(m-1)}\sum_{i=T+1}^{T+m}|\frac{\hat{y}_i - y_i}{y_i}|$

- Absolute *MAPE (AMAPE*) = $\frac{100}{T-(m-1)}\sum_{i=T+1}^{T+m}|\frac{\hat{y}_i - y_i}{\hat{y}_i + y_i}|$

Remark: There is an asymmetry in MAPE, the level y_i matters.

Evaluation of forecasts – Accuracy measures

- % correct sign predictions (PCSP) = $\frac{1}{T-(m-1)}\sum_{i=T+1}^{T+m} z_i$

where z_i = 1 if $(\hat{y}_{i+l} * y_{i+l}) > 0$
= 0, otherwise.

- % correct direction change predictions (PCDP)= $\frac{1}{T-(m-1)}\sum_{i=T+1}^{T+m} z_i$

where z_i = 1 if $(\hat{y}_{i+l} - y_i) * (y_{i+l} - y_i) > 0$
= 0, otherwise.

Remark: We value forecasts with the right direction (sign) or forecast that can predict turning points. For stock investors, the sign matters!

• MSE penalizes large errors more heavily than small errors, the sign prediction criterion, like MAE, does not penalize large errors more.

Evaluation of forecasts – Accuracy measures

Example: We compute MSE and the % of correct direction change (PCDC) predictions for the one-step forecasts for U.S. monthly vehicles sales based on the SES and HW ES models.

```
> MSE_ses
[1] 0.027889

> MSE_hw
[1] 0.01655964
```

• We calculate PCDC with following script for HW & SES:

```
bb_hw <- (car_f_hw - y[(T1-1):(T-1)]) * (y[T1:T] - y[(T1-1):(T-1)])
indicator_hw <- ifelse(bb_hw > 0,1,0)     # ifelse ("if else") produces a 1 if condition is true
pcdc_hw <- sum(indicator_hw)/h
> indicator_hw
 [1] 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 0 0
> pcdc_hw
[1] 0.76
```

Evaluation of forecasts – Accuracy measures

Example (continuation):

```
bb_s <- (ses_f_c - y[(T1-1):(T-1)]) * (y[T1:T] - y[(T1-1):(T-1)])
indicator_s <- ifelse(bb_s > 0,1,0)
pcdc_s <- sum(indicator_s)/h
> indicator_s
 [1] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 0
> pcdc_s
[1] 0.76
```

Note: Same percentage of correct direction change (PCDC) predictions, but the sequence of correct predictions is not the same.

Evaluation of forecasts – DM Test

• To determine if one model predicts better than another, we define the loss differential between two forecasts:

$$d_t = g(e_t^{M1}) - g(e_t^{M2})$$

where g(.) is the forecasting loss function, M1 and M2 are two competing sets of forecasts –could be from models or something else.

• We only need $\{e_t^{M1}\}$ & $\{e_t^{M2}\}$, not the structure of M1 or M2. In this sense, this approach is "*model-free*."

• Typical (symmetric) loss functions: $g(e_t) = e_t^2$ & $g(e_t) = |e_t|$.

• But other g(.)'s can be used: $g(e_t) = \exp(\lambda\, e_t^2) - \lambda\, e_t^2$ (λ>0).

Note: This is a more general test than MGN: It works for any loss function, not just MSE.

Evaluation of forecasts – DM Test

• Then, we test the null hypotheses of equal predictive accuracy:

H_0: $E[d_t] = 0$
H_1: $E[d_t] = \mu \neq 0$.

- Diebold and Mariano (1995) assume $\{e_t^{M1}\}$ & $\{e_t^{M2}\}$ is covariance stationarity and other regularity conditions (finite Var[d_t], independence of forecasts after ℓ periods) needed to apply CLT. Then,

$$\frac{\bar{d} - \mu}{\sqrt{Var[\bar{d}]/T}} \xrightarrow{d} N(0,1), \qquad \bar{d} = \frac{1}{m}\sum_{i=T+1}^{T+m} d_i$$

• Then, under H_0, the DM test is a simple *z-test*:

$$DM = \frac{\bar{d}}{\sqrt{\widehat{Var}[\bar{d}]/T}} \xrightarrow{d} N(0,1)$$

Evaluation of forecasts – DM Test

where $\hat{V}ar[\bar{d}]$ is a consistent estimator of the variance, usually based on sample autocovariances of d_t:

$$\hat{V}ar[\bar{d}] = \gamma(0) + 2\sum_{j=k}^{\ell}\gamma(j)$$

• There are some suggestion to calculate small sample modification of the DM test. For example, :

$$DM^* = DM/\{[T + 1 - 2\,\ell + \ell\,(\ell - 1)/T]/T\}^{1/2} \sim t_{T-1}.$$

where ℓ-step ahead forecast. If time-varying volatility (ARCH) is suspected, replace ℓ with $[0.5\sqrt{(T)}] + \ell$.

Note: If $\{e_t^{M1}\}$ & $\{e_t^{M2}\}$ are perfectly correlated, the numerator and denominator of the DM test are both converging to 0 as $T \to \infty$.

⇒ Avoid DM test when this situation is suspected (say, two nested models.) Though, in small samples, it is OK.

Evaluation of forecasts – DM Test

Example: Code in R

```
dm.test <- function (e1, e2, h = 1, power = 2) {
d <- c(abs(e1))^power - c(abs(e2))^power
 d.cov <- acf(d, na.action = na.omit, lag.max = h - 1, type = "covariance", plot = FALSE)$acf[, , 1]
 d.var <- sum(c(d.cov[1], 2 * d.cov[-1]))/length(d)
 dv <- d.var                                    #max(1e-8,d.var)
 if(dv > 0)
   STATISTIC <- mean(d, na.rm = TRUE) / sqrt(dv)
 else if(h==1)
   stop("Variance of DM statistic is zero")
 else
 {
   warning("Variance is negative, using horizon h=1")
   return(dm.test(e1,e2,alternative,h=1,power))
 }
   n <- length(d)
 k <- ((n + 1 - 2*h + (h/n) * (h-1))/n)^(1/2)
 STATISTIC <- STATISTIC * k
 names(STATISTIC) <- "DM"
}
```


Evaluation of forecasts – DM Test

Example: We compare the SES and HW forecasts for the log of U.S. monthly vehicle sales. We use the *dm.test* function, part of the forecast package.

```
library(forecast)
> dm.test(f_error_c_ses, f_error_c_hw, power=2)

	Diebold-Mariano Test

data:  f_error_c_sesf_error_c_hw
DM = 1.6756, Forecast horizon = 1, Loss function power = 2, p-value = 0.1068
alternative hypothesis: two.sided

> dm.test(f_error_c_ses,f_error_c_hw, power=1)

	Diebold-Mariano Test

data:  f_error_c_sesf_error_c_hw
DM = 1.94, Forecast horizon = 1, Loss function power = 1, p-value = 0.064
alternative hypothesis: two.sided
```

Note: Cannot reject H_0: $MSE_{SES} = MSE_{HW}$ at 5% level

Evaluation of forecasts – DM Test: Remarks

- The DM tests is routinely used. Its "model-free" approach has appeal. There are model-dependent tests, with more complicated asymptotic distributions.

- The loss function does not need to be symmetric (like MSE).

- The DM test is based on the notion of unconditional –i.e., on average over the whole sample- expected loss.

- Following Morgan, Granger and Newbold (1977), the DM statistic can be calculated by regression of d_t, on an intercept, using NW SE. But, we can also condition on variables that may explain d_t. We move from an unconditional to a conditional expected loss perspective.

Combination of Forecasts

• Idea – from Bates & Granger (*Operations Research Quarterly,* 1969):

- We have different forecasts from R models:

$$\hat{Y}_T^{M1}(\ell), \hat{Y}_T^{M2}(\ell), \quad \ldots, \hat{Y}_T^{MR}(\ell)$$

• Q: Why not combine them?

$$\hat{Y}_T^{Comb}(\ell) = \omega_{M1}\hat{Y}_T^{M1}(\ell) + \omega_{M2}\hat{Y}_T^{M2}(\ell) + \ldots + \omega_{MR}\hat{Y}_T^{MR}(\ell)$$

• Very common practice in economics, finance and politics, reported by the press as "consensus forecast." Usually, as a simple average.

• Q: Advantage? Lower forecast variance. Diversification argument.

Intuition: Individual forecasts are each based on partial information sets (say, private information) or models.

91

Combination of Forecasts – Optimal Weights

• The variance of the forecasts is:

$$Var[\hat{Y}_T^{Comb}(\ell)] = \sum_{j=1}^{R} (\omega_{Mj})^2 Var[\hat{Y}_T^{Mj}(\ell)] +$$

$$+ 2\sum_{j=1}^{R}\sum_{i=j+1}^{R} \omega_{Mj}\omega_{Mi} \text{ Covar}[\hat{Y}_T^{Mj}(\ell)\hat{Y}_T^{Mi}(\ell)]$$

Note: Ideally, we would like to have negatively correlated forecasts.

• Assuming unbiased forecasts and uncorrelated errors,

$$Var[\hat{Y}_T^{Comb}(\ell)] = \sum_{j=1}^{R} (\omega_{Mj})^2 \sigma_j^2$$

Example: Simple average: ω_j=1/R. Then,

$$Var[\hat{Y}_T^{Comb}(\ell)] = 1/R^2 \sum_{j=1}^{R} \sigma_j^2 .$$

Combination of Forecasts – Optimal Weights

Example: We combine the SES and HW forecast of log US vehicles sales:

```
f_comb <- (ses_f_c + car_f_hw)/2
f_error_comb <- f_comb - y[T1:T]
> var(f_comb)
[1] 0.0178981
> var(car_f_hw)
[1] 0.02042458
> var(ses_f_c)
[1] 0.01823237
```

Combination of Forecasts – Optimal Weights

- We can derived optimal weights –i,e., ω_j's that minimize the variance of the forecast. Under the uncorrelated assumption:

Under the uncorrelated assumption:

$$\omega_{Mj} *= \sigma_j^{-2} \Big/ \sum_{j=1}^{R} \sigma_j^{-2}$$

- The ω_j*'s are inversely proportional to their variances.

- In general, forecasts are biased and correlated. The correlations will appear in the above formula for the optimal weights. For the two forecasts case:

$$\omega_{Mj} *= (\sigma_1^2 - \sigma_{12})/(\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}) = (\sigma_1^2 - \rho\sigma_1\sigma_2)/(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2)$$

Combination of Forecasts: Regression Weights

• In general, forecasts are biased and correlated. The correlations will appear in the above formula for the optimal weights. Ideally, we would like to have negatively correlated forecasts.

• Granger and Ramanathan(1984) used a regression method to combine forecasts.

- Regress the actual value on the forecasts. The estimated coefficients are the weights.

$$y_{T+\ell} = \beta_1 \hat{Y}_T^{M1}(\ell) + \beta_2 \hat{Y}_T^{M2}(\ell) + ... + \beta_R \hat{Y}_T^{MR}(\ell) + \varepsilon_{T+\ell}$$

• Should use a constrained regression

– Omit the constant

– Enforce non-negative coefficients.

– Constrain coefficients to sum to one

95

Combination of Forecasts: Regression Weights

Example: We regress the SES and HW forecasts against the observed car sales to obtain optimal weights. We omit the constant

```
> lm(y[T1:T] ~ ses_f_c + car_f_hw - 1)

Call:
lm(formula = y[T1:T] ~ ses_f_c + car_f_hw - 1)

Coefficients:
 ses_f_c  car_f_hw
 -0.5426    1.5472
```

Note: Coefficients (weights) add up to 1. But, we see negative weights... In general, we use a constrained regression, forcing parameters to be between 0 and 1 (& non-negative). But, h=25 delivers not a lot of observations to do non-linear estimation.

Combination of Forecasts: Regression Weights

• Remarks:

- To get weights, we do not include a constant. Here, we are assuming unbiased forecasts. If the forecasts are biased, we include a constant.
- To account for potential correlation of errors, we can allow for ARMA residuals or include y_{T+l-1} in the regression.
- Time varying weights are also possible.

• Should weights matter? Two views:

- Simple averages outperform more complicated combination techniques.
- Sampling variability may affect weight estimates to the extent that the combination has a larger MSE.

97

Forecasting: Final Comments

• Since the late 1960s, combination weights have generally been chosen to minimize a symmetric, squared-error loss function.

• But, asymmetric loss functions can also be used. More recent research work find that the optimal weights depend on higher order moments, such a skewness.

98

