ISA 444: Business Forecasting

09: Nonseasonal Smoothing and Forecasting

Fadel M. Megahed, PhD

Endres Associate Professor
Farmer School of Business
Miami University

@FadelMegahed
fmegahed
fmegahed@miamioh.edu
Automated Scheduler for Office Hours

Spring 2023

Quick Refresher from Last Class

- ✅ Describe the benefits and drawbacks of judgmental and quantitative forecasting methods.
- ✅ Explain the difference between causal and extrapolative forecasting.
- ✅ Describe and apply smoothing with a cumulative average.
- ✅ Describe and apply forecasting with a moving average.

Recap: Assignment 06 (Violin Plot)

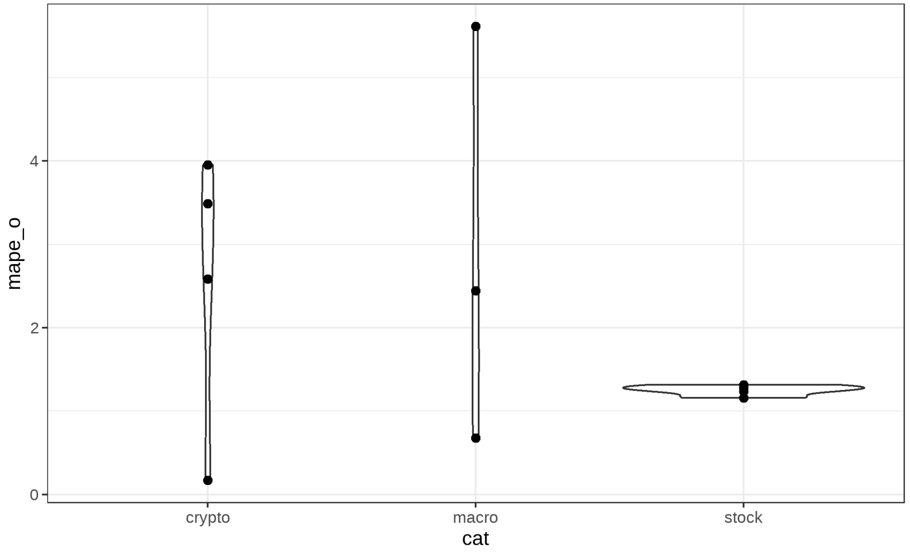
```r
macro = tidyquant::tq_get(
  x = c('UNRATE', 'GNP', 'RHORUSQ156N'),
  from = '2018-01-01', to = '2023-02-12',
  get = 'economic.data'
) |> dplyr::mutate(cat = 'macro')

stocks = tidyquant::tq_get(
  x = c("PM", "UPS",  "SYK",  "PCAR"),
  from = '2018-01-01', to = '2023-02-12') |>
  dplyr::select(symbol, date, adjusted) |>
  dplyr::rename(price = adjusted) |>
  dplyr::mutate(cat = 'stock')


crypto = tidyquant::tq_get(
  c('BTC-USD', 'ETH-USD', 'USDC-USD', 'DOGE-USD'),
  from = '2018-01-01', to = '2023-02-12') |>
  dplyr::select(symbol, date, adjusted) |>
  dplyr::rename(price = adjusted) |>
  dplyr::mutate(cat = 'crypto')


all_data = dplyr::bind_rows(crypto, macro, stocks) |>
  dplyr::group_by(symbol) |>
  dplyr::mutate(
    naive_o = dplyr::lag(price),
    error_o = price - naive_o,
    pe_o = 100*error_o/price
  ) |>
  dplyr::group_by(symbol, cat) |>
  dplyr::summarise(
    me_o = mean(error_o, na.rm = T),
    mape_o = mean(abs(pe_o), na.rm = T)
  )
```

```r
tail(all_data, n = 11)
```

```
## # A tibble: 11 × 4
## # Groups:   symbol [11]
##    symbol      cat            me_o mape_o
##    <chr>       <chr>         <dbl>  <dbl>
##  1 BTC-USD     crypto   4.35        2.58
##  2 DOGE-USD    crypto   0.0000392   3.95
##  3 ETH-USD     crypto   0.397       3.49
##  4 GNP         macro  301.          2.44
##  5 PCAR        stock    0.0256      1.24
##  6 PM          stock    0.0182      1.16
##  7 RHORUSQ156N macro    0.0895      0.676
##  8 SYK         stock    0.0896      1.31
##  9 UNRATE      macro   -0.01        5.61
## 10 UPS         stock    0.0610      1.28
## 11 USDC-USD    crypto  -0.00000135  0.172
```

Recap: Assignment 06 (Violin Plot)

```r
all_data |>
  ggplot2::ggplot(
    ggplot2::aes(x = cat, y = mape_o)
  ) +
  ggplot2::geom_violin() +
  # next two lines of code are not needed
  # included for aesthetics only
  ggplot2::geom_point(size = 2) +
  ggplot2::theme_bw()
```

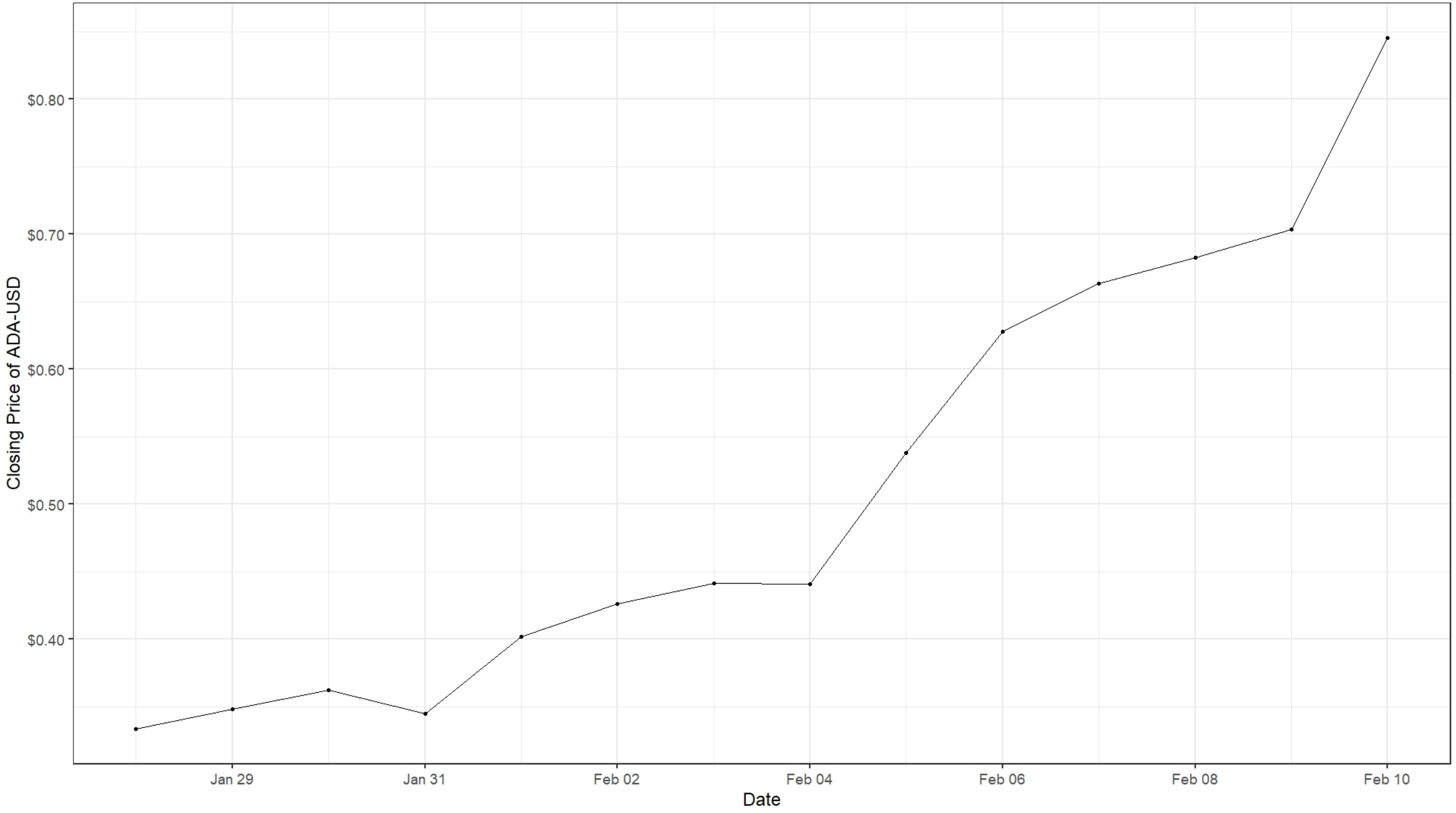


Recap: Assignment 07 (Q3 Insight)

```r
cardano =
  tidyquant::tq_get(
    'ADA-USD', from = '2020-01-01', to = '2023-02-1
  dplyr::select(date, adjusted) |>
  dplyr::mutate(
    ma7 = zoo::rollmeanr(adjusted, k = 7, fill = NA
    ma7_f = dplyr::lag(ma7)
  )

tail(cardano, 10)
```

```
## # A tibble: 10 × 4
##    date       adjusted   ma7 ma7_f
##    <date>        <dbl> <dbl> <dbl>
##  1 2023-02-05    0.392 0.394 0.394
##  2 2023-02-06    0.382 0.395 0.394
##  3 2023-02-07    0.399 0.396 0.395
##  4 2023-02-08    0.394 0.396 0.396
##  5 2023-02-09    0.363 0.391 0.396
##  6 2023-02-10    0.359 0.384 0.391
##  7 2023-02-11    0.369 0.380 0.384
##  8 2023-02-12    0.364 0.376 0.380
##  9 2023-02-13    0.360 0.373 0.376
## 10 2023-02-14    0.386 0.371 0.373
```

Recap: Assignment 07 (Q4 Insight)



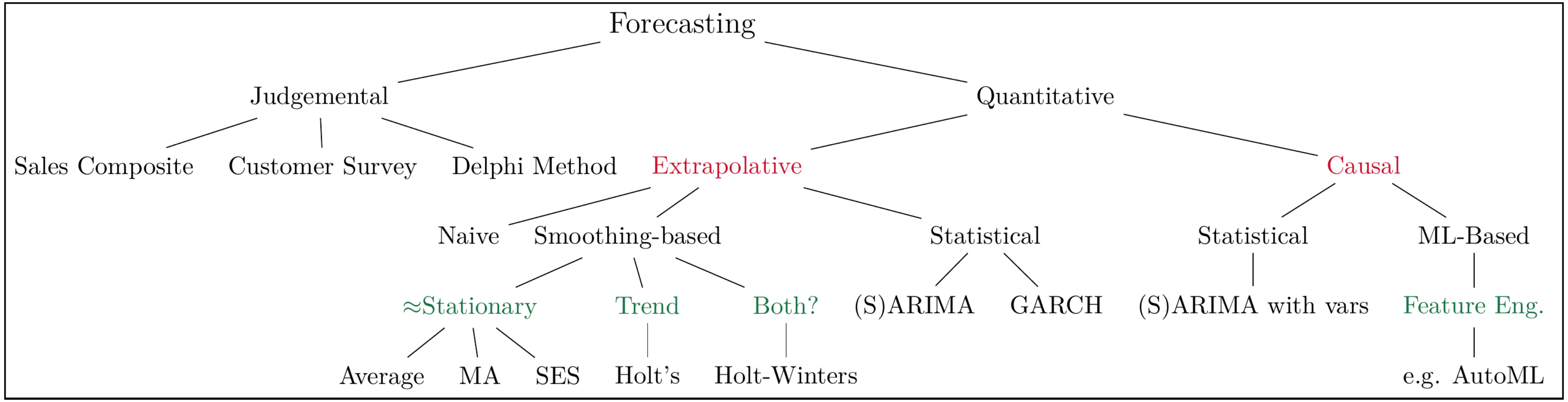
The Rise of Cardano's Closing Price in Early 2021

Learning Objectives for Today's Class

- Recognize time series that are appropriate for simple exponential smoothing (SES).
- Use SES for smoothing and forecasting.

Simple Exponential Smoothing (SES)

Overview of Univariate Forecasting Methods



A 10,000 foot view of forecasting techniques

Notes: My (incomplete) classification of **univariate**forecasting techniques, i.e., they exclude popular approaches used in multivariate time series forecasting.

SES: Definition and Basic Principles

Simple Exponential Smoothing (SES) is a method used for one-step-ahead forecasting of a time series when there is no trend or seasonal pattern, but the mean may drift slowly over time. The mean is said to have a "local level".

Similar to the idea behind a moving average, SES averages the values of the times series, but applies weights of decreasing importance to values that are farther away from the forecast. The weights of the observations "exponentially decay" as we move away from them in time.

The SES one-step-ahead forecast is given by:

$$l_{t+1} = l_t + \alpha(y_t - l_t) = \alpha y_t + (1-\alpha)l_t,$$

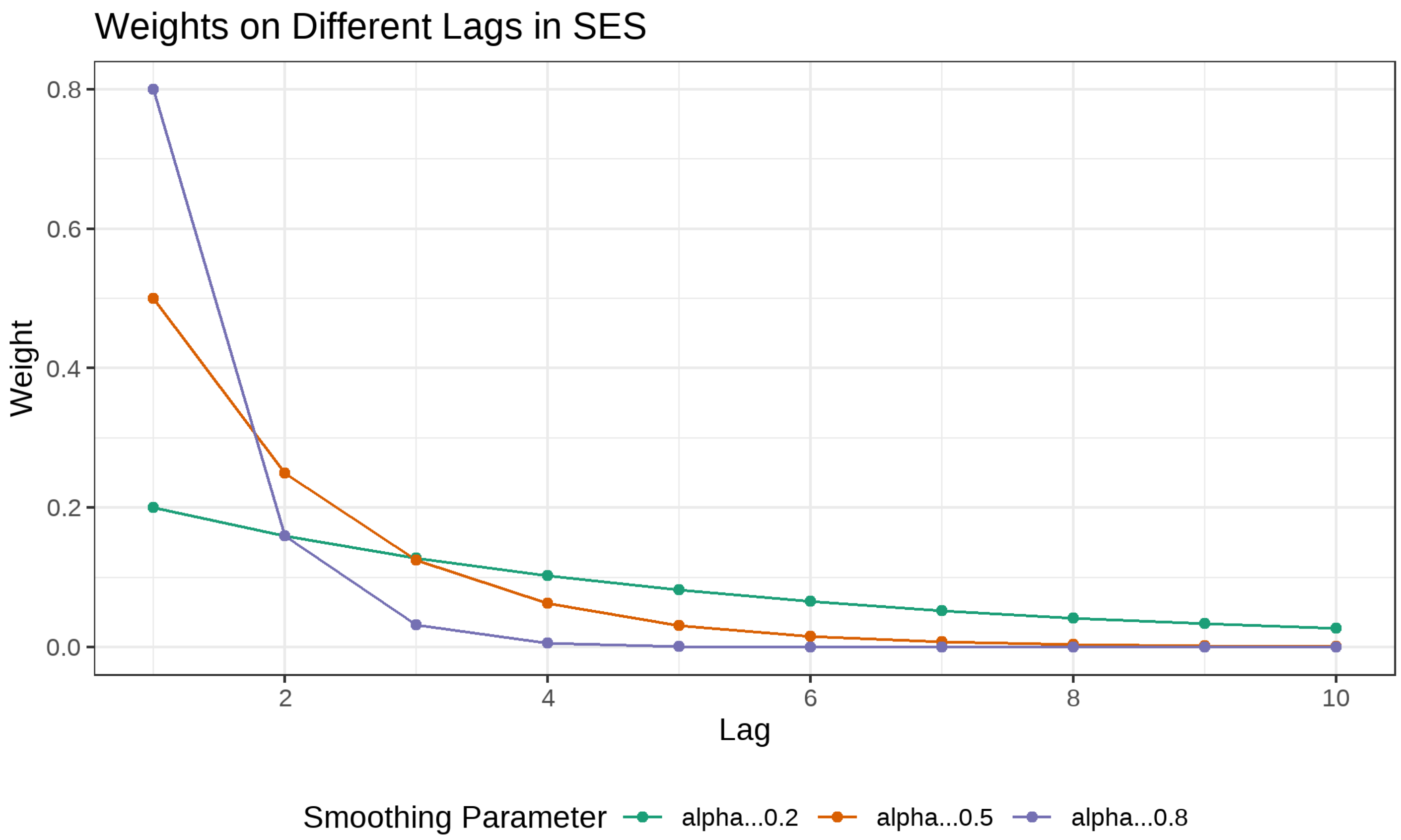
where $0 < \alpha < 1$ is the smoothing parameter, and l_t is the level of the series at time t. Note that l_{1+1} is often denoted as f_{t+1} since it represents our one step-ahead forecast for $t+1$.

SES: Impact of the Smoothing Parameter

$$\begin{aligned}
l_4 &= \alpha y_3 + (1-\alpha) l_3 \\
&= \alpha y_3 + (1-\alpha)[\alpha y_2 + (1-\alpha) l_2] \\
&= \alpha y_3 + \alpha(1-\alpha) y_2 + (1-\alpha)^2 l_2 \\
&= \alpha y_3 + \alpha(1-\alpha) y_2 + (1-\alpha)[\alpha y_1 + (1-\alpha) l_1] \\
&= \alpha y_3 + \alpha(1-\alpha) y_2 + \alpha(1-\alpha)^2 y_1 + (1-\alpha)^3 l_1 \\
&= \alpha y_3 + \alpha(1-\alpha) y_2 + \alpha(1-\alpha)^2 y_1 + (1-\alpha)^3 l_0
\end{aligned}$$

Note that SES needs **two parameters**: (a) the **smoothing parameter** α, and (b) the **initial value for the level** (i.e., l_0).} Note that we will use $l_0 = l_1 = y_1$.

SES: Impact of the Smoothing Parameter



Example 1: Sales Forecast

We will use R to forecast sales, mimicking Table 3.3 in our reference book (See P. 72).

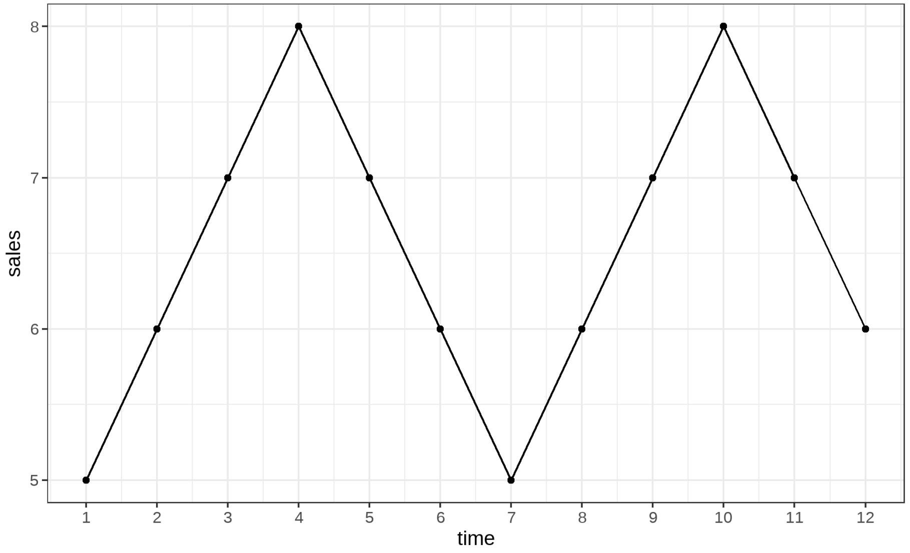
Table 3.3 Illustration of Spreadsheet Calculations for SES Smoothing Constant: Alpha (α) = 0.3

Time	Sales	Forecast	Error	$(Error)^2$	Absolute Error	Absolute Percentage Error
1	5.00					
2	6.00	5.00	1.00	1.00	1.00	16.67
3	7.00	5.30	1.70	2.89	1.70	24.29
4	8.00	5.81	2.19	4.80	2.19	27.38
5	7.00	6.47	0.53	0.28	0.53	7.61
6	6.00	6.63	–0.63	0.39	0.63	10.45
7	5.00	6.44	–1.44	2.07	1.44	28.78
8	6.00	6.01	–0.01	0.00	0.01	0.12
9	7.00	6.01	0.99	0.99	0.99	14.21
10	8.00	6.30	1.70	2.88	1.70	21.21
11	7.00	6.81	0.19	0.04	0.19	2.68
12	6.00	6.87	–0.87	0.75	0.87	14.48
13		6.61				
		Means	**0.49**	**1.46**	**1.02**	**15.26**
		RMSE = 1.21				

Illustration of Spreadsheet Calculations for SES Smoothing Constant: alpha = 0.3

Example 1: Sales Forecast

```
# data from columns 1 and 2
sales_tbl = tibble::tibble(
  time = 1:12,
  sales = c(5, 6, 7, 8, 7, 6, 5, 6, 7, 8, 7, 6)
  )

# a quick plot to visualize the data
sales_tbl |>
  # setting the canvas
  ggplot2::ggplot(ggplot2::aes(x= time, y = sales))
  # lines with dots shown
  ggplot2::geom_line() +
  ggplot2::geom_point() +
  # cleaning up the chart (x_axis pretty and black 
  ggplot2::scale_x_continuous( breaks = scales::pre
  ggplot2::theme_bw()
```



Example 1: Sales Forecast

```
ses_fit =
  forecast::ses(
    # recommended to convert y into a ts when possi
    y = ts(sales_tbl$sales, start = 1, end = 12),
    h = 4, # forecast four values ahead
    # 95%PI
    level = 95,
    # use first value for l_0 and l_1
    initial = 'simple',
    # alpha = 0.3
    alpha = 0.3
  )

names(ses_fit)
```

```
##  [1] "model"     "mean"       "level"    "x"
##  [7] "fitted"    "residuals"  "method"   "series
```

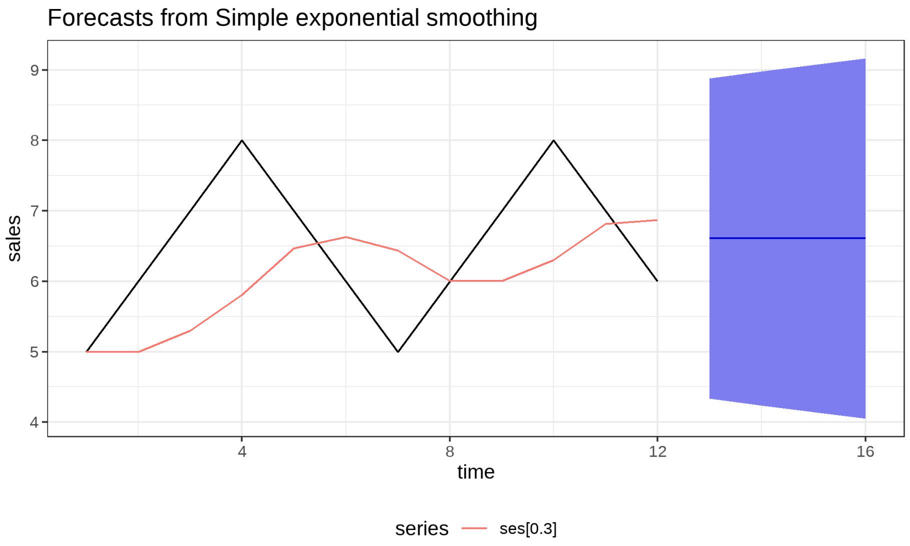
Example 1: Sales Forecast

```
sales_tbl =
  sales_tbl |>
  dplyr::mutate(
    ses_f = ses_fit$fitted,
    error = sales - ses_f,
    error2 = error^2,
    abs_error = abs(error),
    abs_pe = 100*(abs(error/sales))
  )

sales_tbl |>
  dplyr::summarise(
    # needs to be before the other summaries
    # to compute correctly in R
    rmse = sqrt( mean(error^2, na.rm = T) ),
    # a bit fancy but the code below allows you
    # to use the custom mean function with input
    # x to change for each column to compute the
    # mean across the columns from error to
    # abs_pe
    dplyr::across(
      .cols = error:abs_pe,
      .fns = function(x) mean(x, na.rm = T)
    )
  )
```

```
## # A tibble: 1 × 5
##    rmse error error2 abs_error abs_pe
##   <dbl> <dbl>  <dbl>     <dbl>  <dbl>
## 1  1.16 0.447   1.34     0.937   14.0
```

Example 1: Sales Forecast

```
forecast::autoplot(ses_fit) +
  forecast::autolayer(ses_fit$fitted, series = 'ses
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = 'bottom') +
  ggplot2::labs(y = 'sales', x = 'time')
```

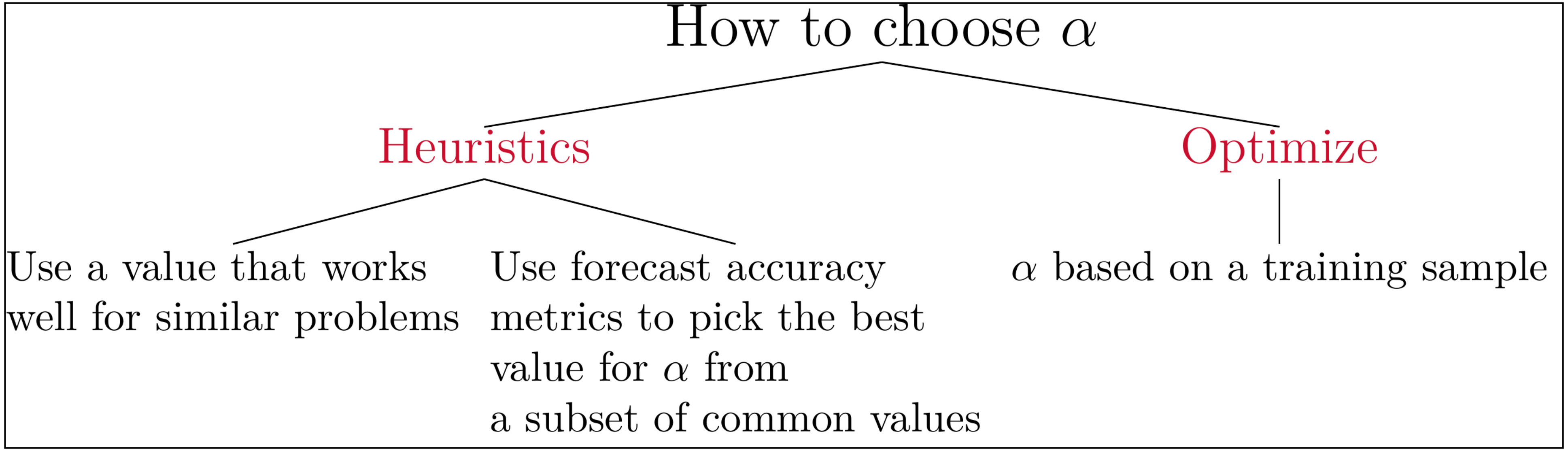


Poll: Why are Our Results Different from Table 3.3?

Go to www.menti.com/al1sej81axe4

Why are our results in Slide 11 Different tha
from Table 3.3?

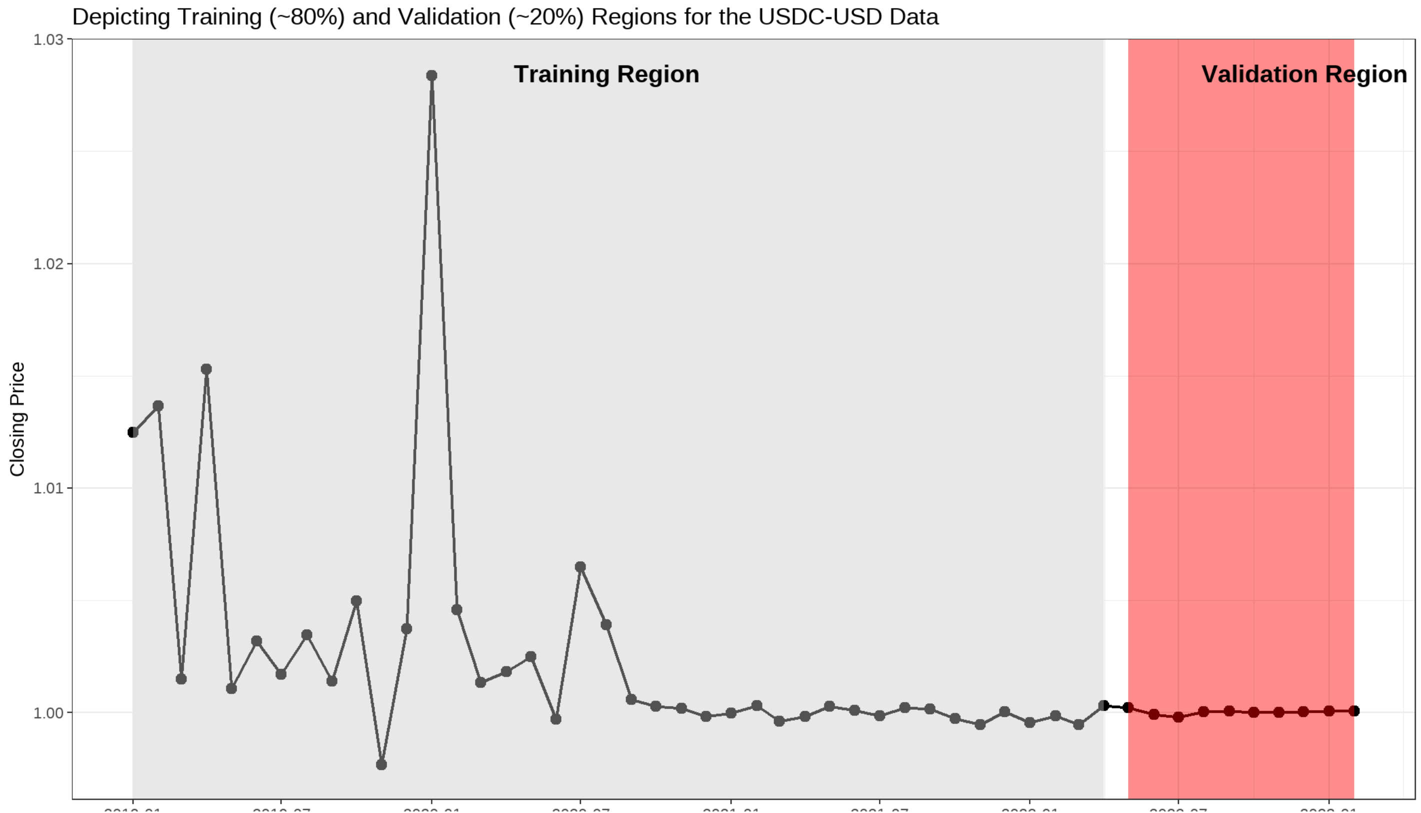
Let us Talk about α



Strategies for choosing an appropriate alpha value.

Training and Validation Samples

Often you determine your smoothing parameter based on a training or baseline sample of observations, not the entire series. Then you apply the model using the smoothing parameter to the new observations and evaluate the fit on the out-of-sample observations.



Training and Validation Samples

(1) Determine the size of the training, or baseline sample.
a. Training sample size is usually 70-80% of the total available data.
b. Training sample should maintain time order. With time series, the training sample usually consists of observations at the beginning of the sample, while validation sample consists of observations at the end of the available data.

(2) Select the smoothing parameter based on the observations in the training sample only.

(3) Evaluate the “in-sample” performance of the forecast using RMSE and graphs using the training sample.

(4) Apply the model chosen in #2 to the validation sample.

(5) Evaluate the “out-of-sample” performance of the forecast using RMSE and graphs.

Source: Slide is based on [Dr. Allison Jones-Farmer's Handouts]{https://miamioh.edu/fsb/directory/?up=/directory/farmerl2}{} for ISA 444, Fall 2022.

Demo: Optimizing the Smoothing Parameter

To illustrate the aforementioned concepts, let us examine the data for the WFJ Sales Example (i.e., Example 3.2 in our textbook). Per the textbook example, we will use the first the 26 observations as the estimation sample.

We will live code the example below in class.

```
wfj_sales = readxl::read_excel('../../data/WFJ_sales.xlsx') |>
  dplyr::select(c(1,2))
```

Recap

Summary of Main Points

By now, you should be able to do the following:

- Recognize time series that are appropriate for simple exponential smoothing (SES).
- Use SES for smoothing and forecasting.

Things to Do to Prepare for Our Next Class

- **Recommended:** Thoroughly read Chapter 3.1-3.3 of our reference book.
- **Required:** Complete assignment08.