ISA 444: Business Forecasting

06: The Naive Forecast, Measures of Forecast Accuracy and the Prediction Interval

Fadel M. Megahed, PhD

Endres Associate Professor
Farmer School of Business
Miami University

@FadelMegahed
fmegahed
fmegahed@miamioh.edu

Automated Scheduler for Office Hours

Fall 2023

Quick Refresher from Last Class

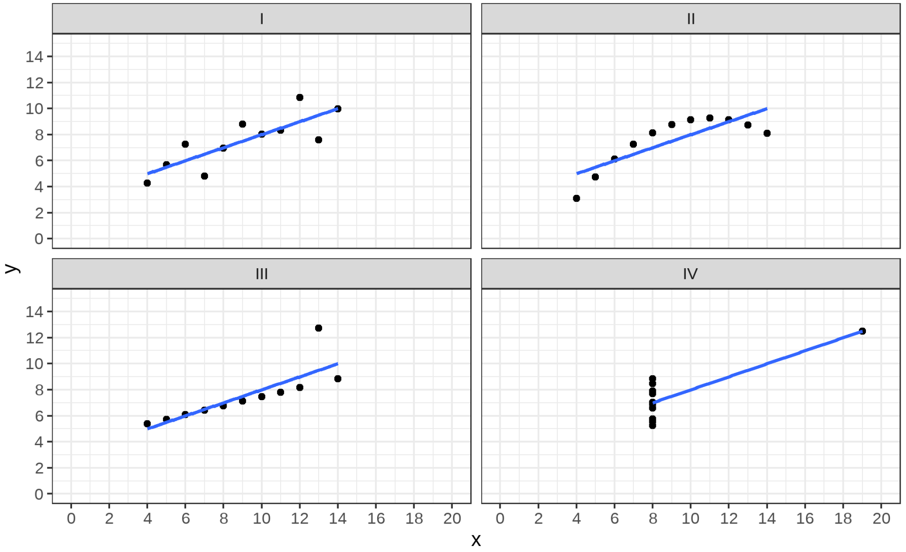
- ✅ Use numerical summaries to describe a time series.
- ✅ Explain what do we mean by correlation.
- ✅ aaply transformations to a time series.

Recap: Viz + Numerical Summary = Big Picture

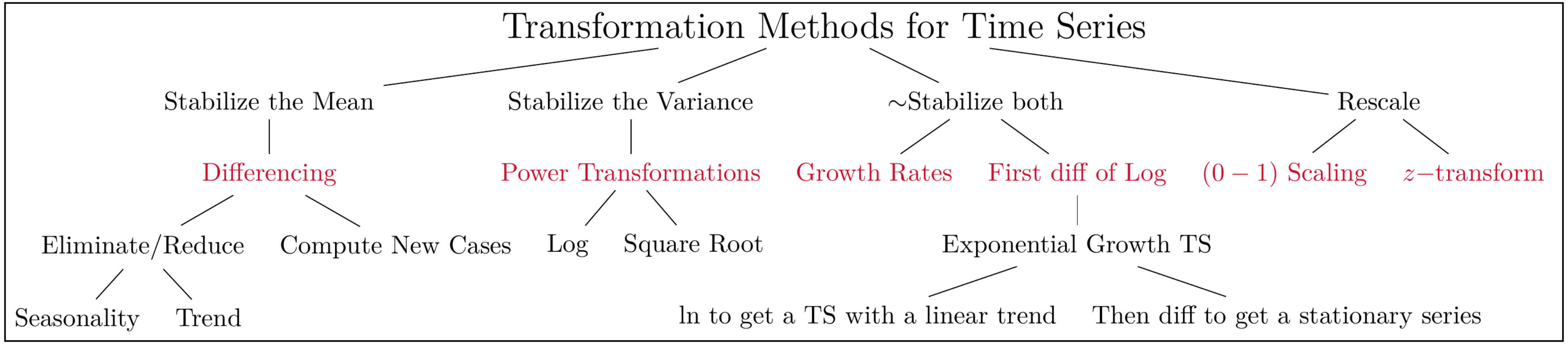
set	x.mean	x.sd	y.mean	y.sd	corr
I	9	3.32	7.5	2.03	0.82
II	9	3.32	7.5	2.03	0.82
III	9	3.32	7.5	2.03	0.82
IV	9	3.32	7.5	2.03	0.82

Showing 1 to 4 of 4 entries

Previous 1 Next



Recap: Guidelines for Transforming TS Data



A classification of common transformation approaches for time series data

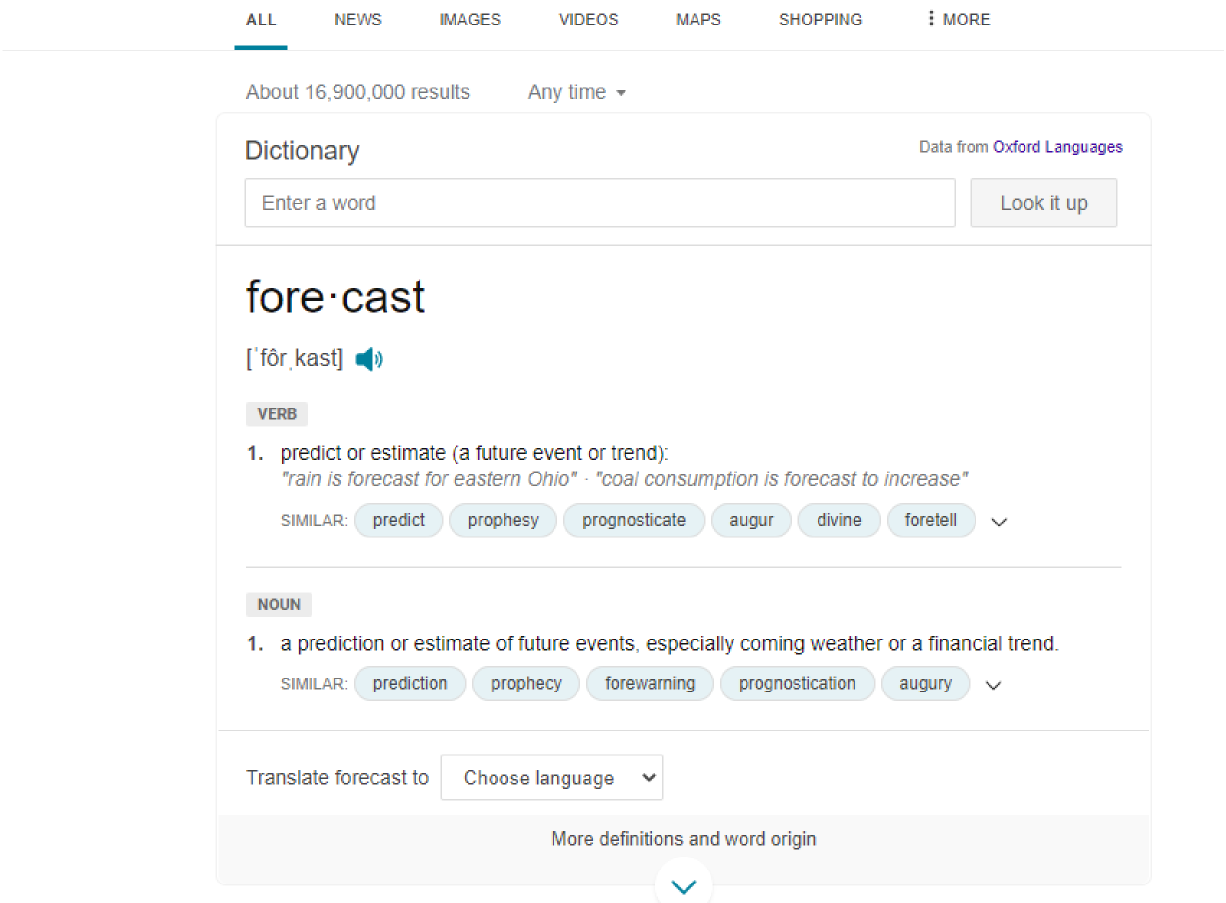
My (incomplete) attempt to provide you with a taxonomy for time series data transformations.

Learning Objectives for Today's Class

- Compute the nonseasonal naive forecast.
- Apply and interpret measures of forecast accuracy.
- Interpret prediction intervals for a simple forecast.

The Naive Forecast

Recap: What is Forecasting?



A Naïve Forecast

- A naïve forecast for an observation, Y_t , is the observation prior, Y_{t-1}.
- For some types of time series (e.g. **Random Walks**), a naïve forecast is the **best possible forecast one can make**.
- For almost any time series, the naive forecast should be included as a **benchmark/baseline.** How much is your forecast better than the naïve forecast? Is it worth it?
- In the case of seasonal data, a naïve forecast could be the observation from the prior period.
  - For example, in the case of monthly data, the naïve forecast for the observation $Y_{Jan2024}$ could be $Y_{Jan2023}$. In this case, we would denote the frequency, m=12, and the naïve forecast for Y_t is the observation m periods prior, or Y_{t-m}.

Intuition and Code Behind the Naïve Forecast

```r
aapl = tidyquant::tq_get('AAPL')

aapl |>
  # selecting the two columns of interest
  dplyr::select(date, adjusted) |>
  # printing the last 5 observations for demo
  dplyr::slice_tail(n = 5) |>
  # printing empty space under a naive_f column
  dplyr::mutate(naive_f = '---')
```

```
## # A tibble: 5 × 3
##   date       adjusted naive_f
##   <date>        <dbl> <chr>
## 1 2023-09-07     178. ---
## 2 2023-09-08     178. ---
## 3 2023-09-11     179. ---
## 4 2023-09-12     176. ---
## 5 2023-09-13     174. ---
```

Intuition and Code Behind the Naïve Forecast

Join at https://www.menti.com/alqezoh8oh13

I know how to compute the naive fore and Python

Measures of Forecast Accuracy

- The measures of accuracy we will discuss all deal with the difference between the **actual observed value** (Y_t) and the **forecasted value** (F_t) for time t.
- In order to measure forecast accuracy, we assume we have m actual values available, thus we have $Y_{t+1}, Y_{t+2}, \ldots, Y_{t+m}$ and forecasts $F_{t+1}, F_{t+2}, \ldots, F_{t+m}$.
- This is important because we will be averaging the forecast errors over m.
- **Forecast Error:** $e_{t+i} = Y_{t+i} - F_{t+i}$.

Measures of "Average" Forecast Performance

- A positive average error measure indicates that: for your m forecasts, on average your **actuals** (Y_{t+i}) are **larger than** their corresponding **forecasted values** (F_{t+i}), i.e., you are underestimating.
- A negative average error measure indicates that you are overestimating on average.
- If your average error **(percent)** measure is ~0; you have an **unbiased** forecast.
  - An unbiased average measure on its own is meaningless; look at its variability.
- **Mean Error:**

$$ME = \frac{\sum_{i=1}^{m} e_{t+i}}{m}.$$

- **Mean Percentage Error:**

$$MPE = \frac{100}{m} \sum_{i=1}^{m} \frac{e_{t+i}}{Y_{t+i}}.$$

Measures of "Variability" in Forecast Performance

- **Absolute Forecast Error:**

$$|e_{t+i}| = |Y_{t+i} - F_{t+i}|.$$

- **Squared Forecast Error:**

$$(e_{t+i})^2 = (Y_{t+i} - F_{t+i})^2.$$

- **Mean Absolute Error:**

$$MAE = \frac{\sum_{i=1}^{m} |e_{t+i}|}{m}.$$

- **Root Mean Squared Error:**

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m} (e_{t+i})^2}{m}}.$$

Measures of "Relative" Forecast Performance

- **Mean Absolute Percentage Error:**

$$MAPE = \frac{100}{m} \sum_{i=1}^{m} \frac{|e_{t+i}|}{m}.$$

- **Relative Mean Absolute Error:**

$$RelMAE = \frac{\sum_{i=1}^{m} |e_{t+i}|}{\sum_{i=1}^{m} |Y_{t+i} - Y_{t+i-1}|}.$$

- **Thiel's U:**

$$U = \sqrt{\frac{\sum_{i=1}^{m} (e_{t+i})^2}{\sum_{i=1}^{m} (Y_{t+i} - Y_{t+i-1})^2}}.$$

Computing these Measures in R via the forecast

```r
aapl_df =
  aapl |>
  # the naive forecast = Y_{t-1} = lag(Y_t, 1)
  dplyr::mutate( naive_f = dplyr::lag(adjusted, n =1) )

# printing the first 3 rows
aapl_df |> dplyr::slice_head(n = 3)
```

```
## # A tibble: 3 × 9
##   symbol date        open  high   low close    volume adjusted naive_f
##   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>    <dbl>   <dbl>
## 1 AAPL   2013-01-02  19.8  19.8  19.3  19.6 560518000     16.8    NA
## 2 AAPL   2013-01-03  19.6  19.6  19.3  19.4 352965200     16.6    16.8
## 3 AAPL   2013-01-04  19.2  19.2  18.8  18.8 594333600     16.1    16.6
```

```r
# computing the accuracy metrics via the forecast pkg
forecast::accuracy(
  # we start at row 2 since first fct is NA
  object = aapl_df$naive_f[2:nrow(aapl_df)],
  x = aapl_df$adjusted[2:nrow(aapl_df)]
)
```

```
##                  ME     RMSE       MAE        MPE     MAPE
## Test set 0.05847653 1.627164 0.9113523 0.07051151 1.260787
```

Computing these Measures in R with dplyr

```
aapl_df |>
  dplyr::mutate(
    error = adjusted - naive_f, # actual - forecast
    perc_error = error / adjusted # error/actual
  ) |>
  dplyr::group_by(symbol) |> # not needed but it would allow you to compute for multiple TS
  dplyr::summarise(
    # measures of "average" forecast performance
    me = mean( error, na.rm = T ),
    mpe = 100* mean( perc_error, na.rm = T ),
    # measures of "variability" in forecast performance
    mae = mean( abs(error), na.rm = T ),
    mape = 100* mean( abs(perc_error), na.rm = T),
    rmse = mean( error^2, na.rm = T ) |> sqrt()
  )
```

```
## # A tibble: 1 × 6
##   symbol     me    mpe   mae  mape  rmse
##   <chr>   <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 AAPL   0.0585 0.0705 0.911  1.26  1.63
```

Some Practical Insights

Main Insight(s): (Edit below)

- The **naive forecast** is .... of the series; thus, the forecast error is the ...
- In general, if the $MAE \approx |ME|$, then we conclude that ... . If we were using a naive forecast in such a case, then we can also conclude that ...
- Irrespective of the forecasting method, the MPE and MAPE are useful/valid if and only if ....

The Prediction Interval

Prediction Intervals $\neq$ Confidence Intervals

- Prediction intervals and confidence intervals are **not** the same.
- A **prediction interval** is an interval associated with a **random variable yet to be observed, with a specified probability of the random variable lying within the interval**.
  - For example, I might give an 80% interval for the forecast of GDP in 2024. The actual GDP in 2024 should lie within the interval with probability 0.8.
- A **confidence interval** is an interval associated with **a parameter** (e.g., the mean of a random variable) ... The parameter is assumed to be non-random but unknown, and the confidence interval is computed from data. Because the data are random, the interval is random ... That is, with a large number of repeated samples, 95% of the intervals would contain the true parameter if you built a 95%.
- **Prediction intervals** are wider than confidence intervals since it includes the variance of ϵ (the error in our predictions).

Adapted from Rob Hyndman: The difference between prediction intervals and confidence intervals

Point vs Interval Forecasts

- **Point Forecasts:** future observations for which we report a single forecast observation.
- **Interval Forecast:** a range of values that are reported to forecast an outcome.

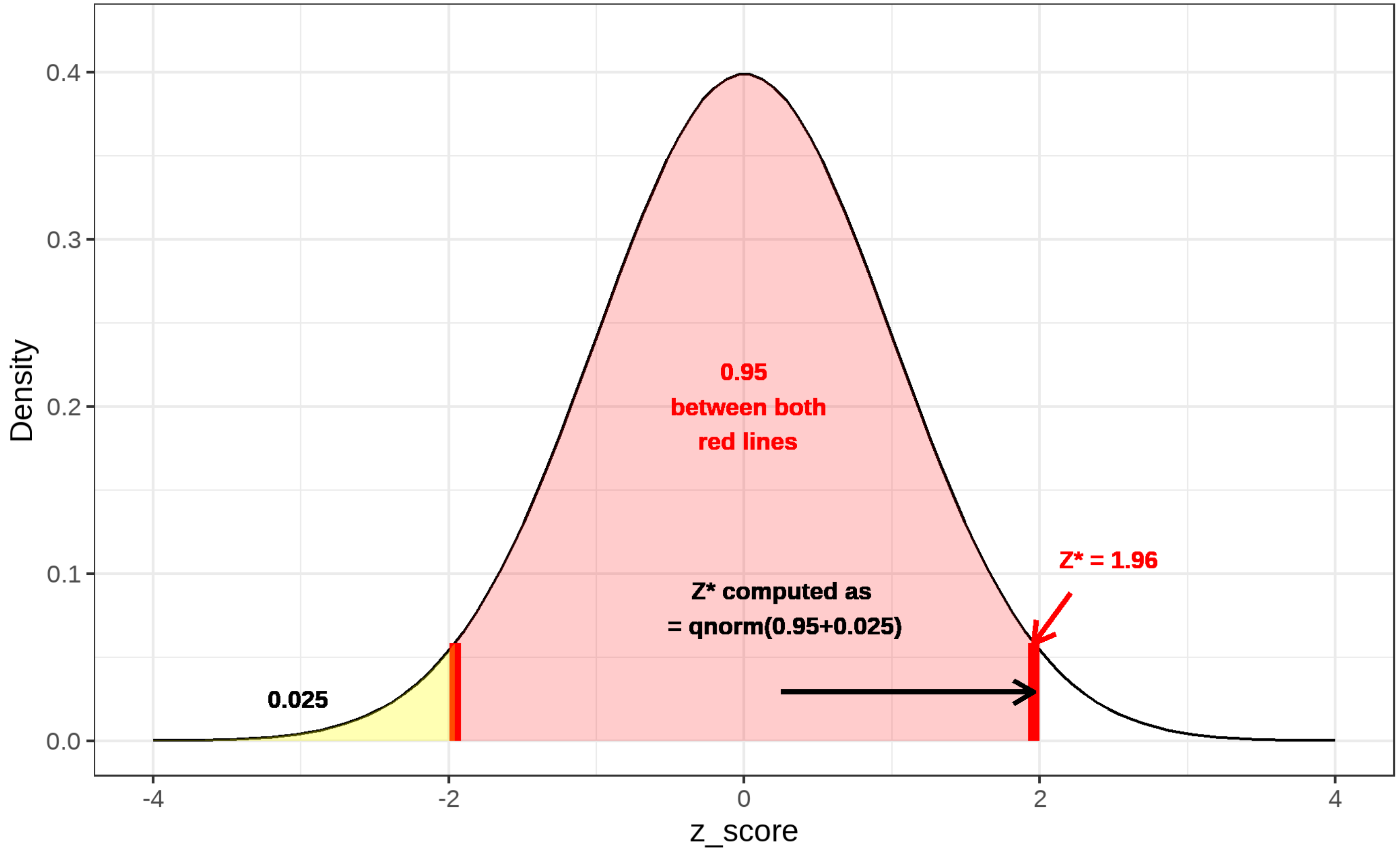
If we assume the forecast errors follow a Normal Distribution, an approximate $100(1-\alpha)$ prediction interval can be computed as follows:

$$\hat{F}_t \pm Z^* * SD,$$

where:

- $\hat{F}_t$ forecast at time t.
- The RMSE can be used as an estimate of the standard deviation of the forecast errors.
- Z^* is the quantile corresponding to $100(1-\frac{\alpha}{2})$.

Recall: Standard Normal Distribution



See the interactive version of this chart at Interactive Standard Normal Dist.

Prediction Intervals for aapl: "By Hand"

```
aapl_df
```

```
## # A tibble: 2,693 × 9
##    symbol date        open  high   low close    volume adjusted naive_f
##    <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>    <dbl>   <dbl>
##  1 AAPL   2013-01-02  19.8  19.8  19.3  19.6 560518000     16.8    NA
##  2 AAPL   2013-01-03  19.6  19.6  19.3  19.4 352965200     16.6    16.8
##  3 AAPL   2013-01-04  19.2  19.2  18.8  18.8 594333600     16.1    16.6
##  4 AAPL   2013-01-07  18.6  18.9  18.4  18.7 484156400     16.0    16.1
##  5 AAPL   2013-01-08  18.9  19.0  18.6  18.8 458707200     16.1    16.0
##  6 AAPL   2013-01-09  18.7  18.8  18.4  18.5 407604400     15.8    16.1
##  7 AAPL   2013-01-10  18.9  18.9  18.4  18.7 601146000     16.0    15.8
##  8 AAPL   2013-01-11  18.6  18.8  18.5  18.6 350506800     15.9    16.0
##  9 AAPL   2013-01-14  18.0  18.1  17.8  17.9 734207600     15.3    15.9
## 10 AAPL   2013-01-15  17.8  17.8  17.3  17.4 876772400     14.9    15.3
## # ℹ 2,683 more rows
```

Prediction Intervals for aapl: "By Hand"

```
# 95% prediction_interval
alpha = 0.95
curve_prob = 1 - ((1-alpha)/2)

# adding the error since it was not saved to aapl_df
aapl_df = aapl_df |> dplyr::mutate( error = adjusted - naive_f   )

# recomputing the rmse and saving it to an object titled rmse
aapl_df |>
  dplyr::group_by(symbol) |>
  dplyr::summarise(rmse = error^2 |> mean(na.rm = T) |> sqrt() ) |>
  dplyr::pull(rmse)  -> # pull rmse value from tibble (i.e., convert to tibble/vec)
  rmse

# computing the prediction intervals for our data
aapl_df =
  aapl_df |>
  dplyr::mutate(
    pi_l = naive_f - (abs(qnorm(curve_prob))*rmse),
    pi_u = naive_f + (abs(qnorm(curve_prob))*rmse)
  )
```

Prediction Intervals for aapl: "By Hand"

```r
aapl_df |>
  ggplot2::ggplot(ggplot2::aes(x = date)) +
  # Plotting the actual in black
  ggplot2::geom_point(ggplot2::aes(y = adjusted)) +
  ggplot2::geom_line(ggplot2::aes(y = adjusted)) +
  # Plotting the forecast in darkgray
  ggplot2::geom_line(ggplot2::aes(y = naive_f), color =
  # Plotting the PI in red
  ggplot2::geom_ribbon(
    ggplot2::aes(ymin = pi_l, ymax = pi_u, fill = '95%
  ) +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = 'none')
```

Recap

Summary of Main Points

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

- Compute the nonseasonal naive forecast.
- Apply and interpret measures of forecast accuracy.
- Interpret prediction intervals for a simple forecast.

Things to Do to Prepare for Our Next Class

- Go over your notes and read through Chapter 2 of our reference book.
- **Potential Practice Problems:**
  - Extract data using either the `tq_get()` (tidyquant package) or the `covid19()` (COVID19 package), and compute the transformations using a manual (i.e., Excel) approach and R/Python.
  - **Reference Book Example:** For the Means approaches in Example 2.7 (P.49), use R/Python to compute the 7 error forecasting metrics (data available here).
  - **Reference Book Exercise 2.12:** Compute the forecast errors for the naive forecast.