

ISA 444: Business Forecasting

17: Review and Stationarity

Fadel M. Megahed, PhD

Endres Associate Professor
Farmer School of Business
Miami University

 @FadelMegahed

 fmegahed

 fmegahed@miamioh.edu

 Automated Scheduler for Office Hours

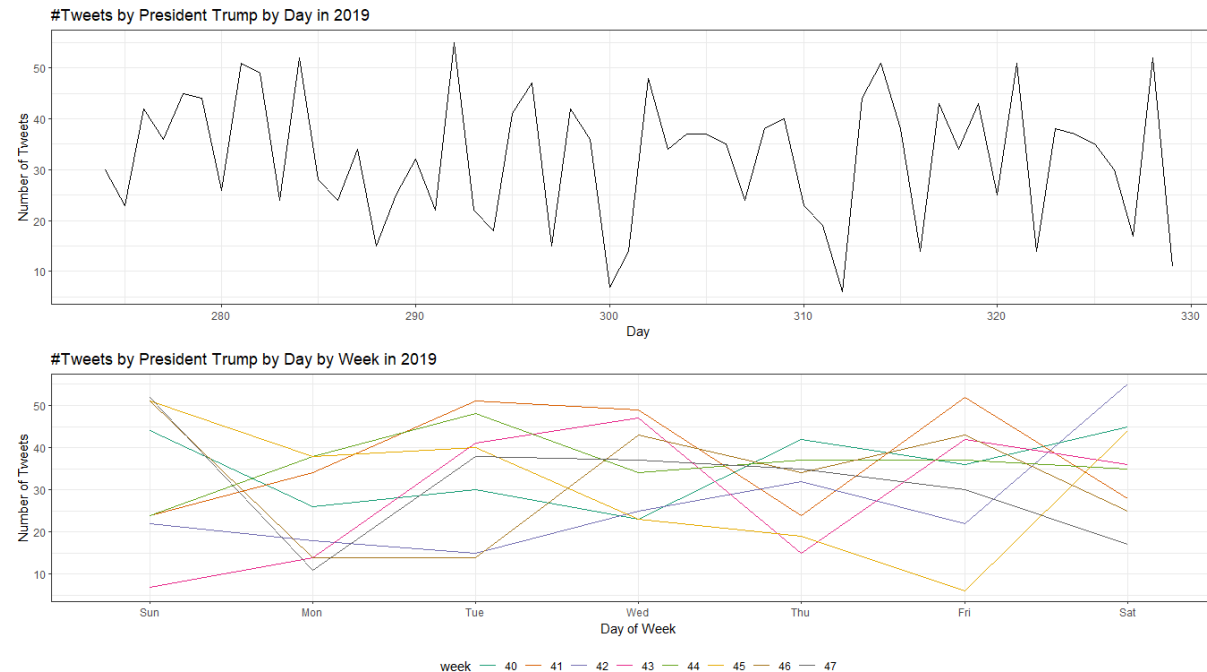
Spring 2023

Exam 02 Review

Q1

Q1 Solution

Assuming that this sample is a representative sample of former President Trump's tweeting patterns, which of the following conclusions are reasonable to make about the daily tweeting patterns of the former President:



Exam 02 Review

Q1

Q1 Solution

Based on examining the charts and/or inspecting the [trumpTweetCount.csv](#), one can observe the following:

- The data seems to be varying around a stationary mean of 21.3 and the variance seems to be near constant as well.
- There is no evidence for seasonality (e.g., lines cross on the num tweets vs. day of week chart, indicating that there are no fixed patterns; Friday is sometimes the day of highest tweet volume and sometimes among the lowest)

Exam 02 Review

Q2

Q2 Solution

By closely investigating the actual tweet count data available at `trumpTweetCount.csv`, select all possible accuracy metrics that can be used to evaluate the goodness-of-fit produced by a specific forecasting/smoothing for the column (n)

Exam 02 Review

Q2	Q2 Solution
----	-------------

Based on examining the [trumpTweetCount.csv](#), the following metrics can be used:

- ME can be used to measure the bias in the forecast per [Slide 12 in Class 06](#).
- MAE and RMSE can be used to measure the variability in the forecast per [Slide 13 in Class 06](#).
- MAPE **cannot** be used since the number of tweets = 0 o yearDay 82.
 - Note that this question is similar to Q7 in Exam 01.

Exam 02 Review

Q3

Q3 Solution

Suppose you were to apply Holt's Method as a forecasting procedure ($\alpha = 0.2$, $\beta = 0.1$, $\text{initial} = \text{'optimal'}$) on the GDP Data from FRED. What is your computed forecast for the GDP (i.e., price column) on 2023-07-01? Do not apply any transformations to your dataset.

Exam 02 Review

Q3

Q3 Solution

```
gdp = tidyquant::tq_get('GDP', from = '1947-01-01', to = '2022-12-01', get = 'economic.data')
gdp_holt = forecast::holt(gdp$price, h = 10, alpha = 0.2, beta = 0.1, initial = 'optimal')

# solution 1
summary(gdp_holt) # third value in the printout since we need Q3 data

# solution 2
gdp_holt$mean

# solution 3
gdp_holt
```

Exam 02 Review

Q4

Q4 Solution

Based on the `forecast::accuracy()` function, the fitted holt model from the previous question outperforms the naive forecast.

Exam 02 Review

Q4 Q4 Solution

```
gdp = tidyquant::tq_get('GDP', from = '1947-01-01', to = '2022-12-01', get = 'economic.data')
gdp_holt = forecast::holt(gdp$price, h = 10, alpha = 0.2, beta = 0.1, initial = 'optimal')

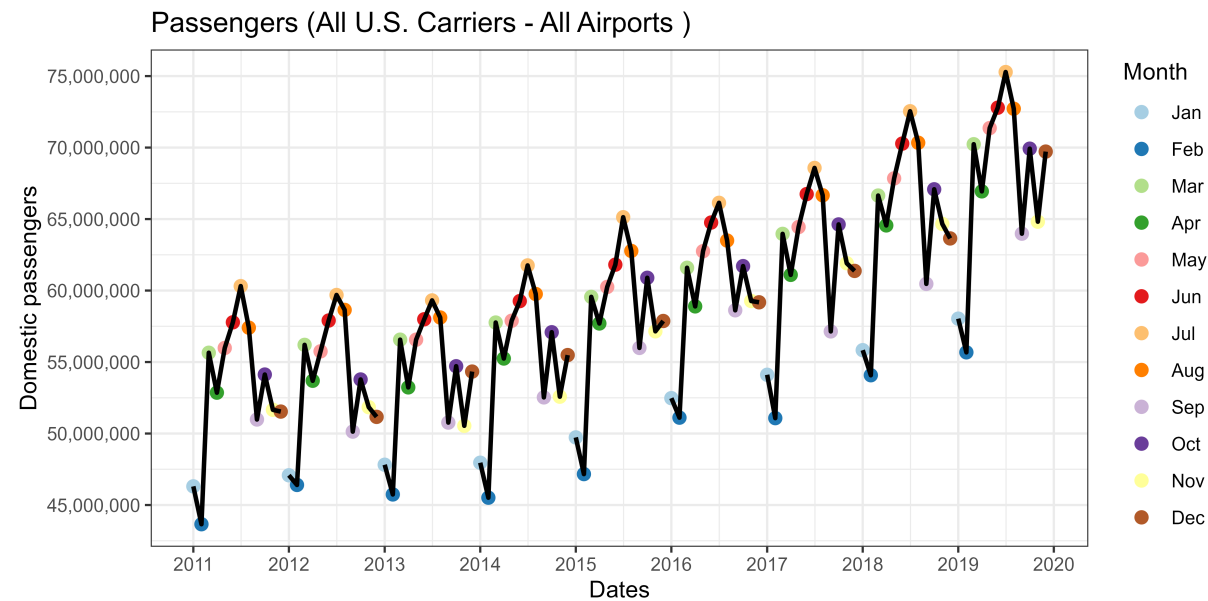
# MASE = 1.06, which means that this is a worse forecast when compared the naive
forecast::accuracy(gdp_holt)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 24.4912 266.3348 109.1605 0.3006439 1.792617 1.060017 0.7181486
```

Exam 02 Review

Q5 and its Solution

Based on the plot below, the airlines dataset can be considered seasonal.



Yes, monthly patterns are repeating from one year to another.

Exam 02 Review

Q6	Q6 Code	Q6 Output
----	---------	-----------

Provide an R code that can be used to replicate the chart above. For the plot, I would like you to:

- Have a date variable, which you will need to create, on the x-axis;
- Have the DOMESTIC on the y-axis;
- Have the line be connected only for the year; and
- Have the points to be colored according to each month.

You do not have to worry about the x-axis/y-axis titles/formatting, plot titles, themes, and my choice of color palette, i.e., focus on achieving the above four requirements.

Exam 02 Review

Q6	Q6 Code	Q6 Output
----	---------	-----------

```
airlines = readr::read_csv('../data/us_passengers_domestic.csv') |>
  # to focus on post financial crisis and pre-COVID19 data
  dplyr::filter(Year > 2010 & Year < 2020)

airlines |>
  dplyr::mutate(
    date = lubridate::ymd( paste(Year, Month, '01', sep = '-') ),
    # you do not have to overwrite month
    Month = lubridate::month(date, label = T)) |>
  ggplot2::ggplot( ggplot2::aes(x = date, y = DOMESTIC, group = Year)) +
  ggplot2::geom_point(ggplot2::aes(color = Month), size = 2.5) +
  ggplot2::geom_line(size = 1) +
  ggplot2::theme_bw() +
  ggplot2::scale_x_date(breaks = scales::pretty_breaks(n=12)) +
  ggplot2::scale_y_continuous(breaks = scales::pretty_breaks(n=6), labels = scales::comma) +
  ggplot2::scale_color_brewer(palette = 'Paired') +
  ggplot2::labs(
    title = 'Passengers (All U.S. Carriers - All Airports )',
    x = 'Dates',
    y = 'Domestic passengers')
```

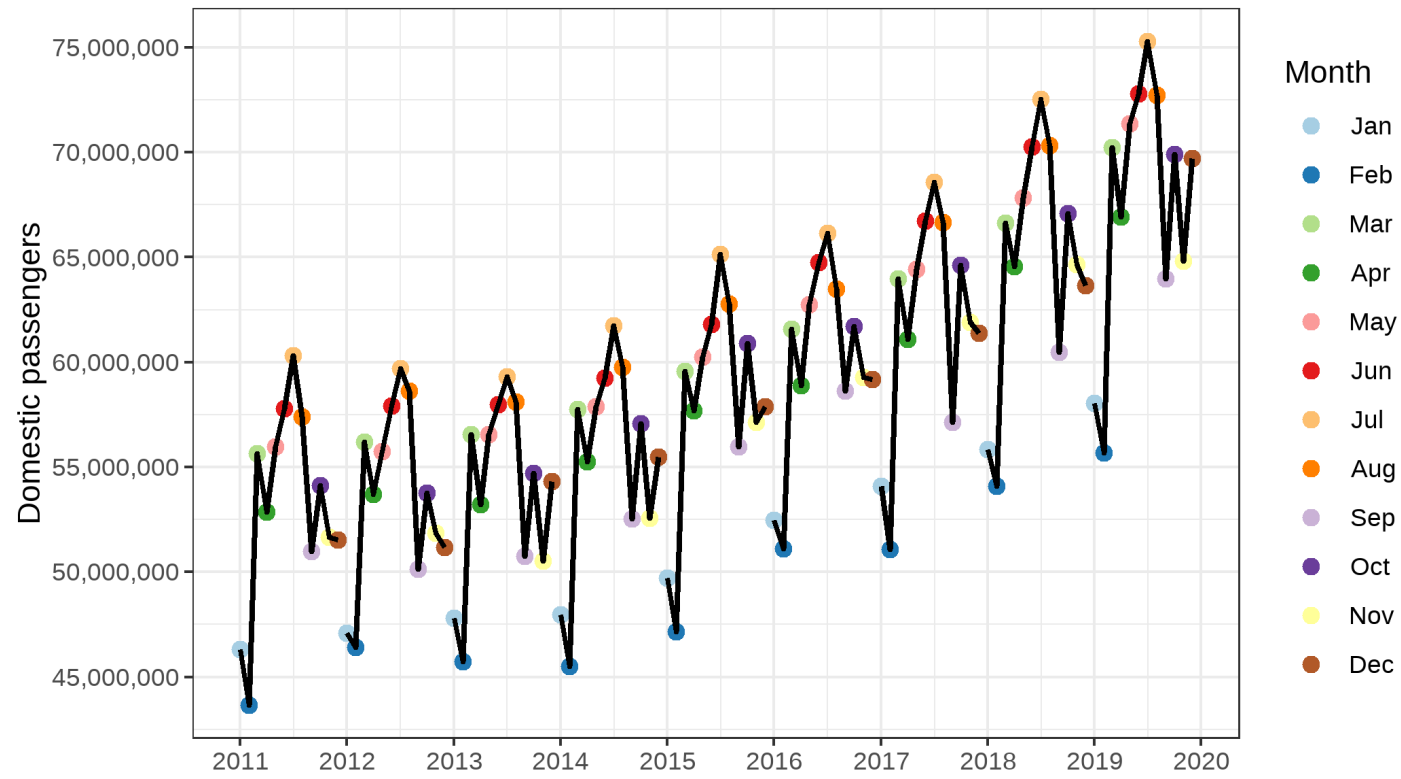
Exam 02 Review

Q6

Q6 Code

Q6 Output

Passengers (All U.S. Carriers - All Airports)



Exam 02: Review

Q7

Q7 Solution

If you were to build on the R code below, what is the `airlines` dataset's seasonal factor for July?:

```
airlines_ts = ts(data = airlines$DOMESTIC, start = c(2011,1), frequency = 12)
decomposed_ts = decompose(airlines_ts, type = 'additive')
forecast::autoplot(decomposed_ts)
```

Please insert your answer to rounded to the nearest digit (i.e. you do not have to return any decimal places for example 12321).

Exam 02: Review

Q7

Q7 Solution

```
# solution 1  
decomposed_ts$seasonal[7] |> scales::comma()
```

```
## [1] "6,573,221"
```

```
# solution 2  
decomposed_ts$figure[7] |> scales::comma()
```

```
## [1] "6,573,221"
```

```
# solution 3 (making your previous figure interactive)  
# forecast::autoplot(decomposed_ts) |> plotly::ggplotly()
```

Exam 02: Review

Q8

Q8 Solution

What is the interpretation of your reported value from the previous question?

Exam 02: Review

Q8

Q8 Solution

The number of air passengers is 6,573,221 higher in July than the average for a particular year, i.e., many more people travel in July.

Exam 02: Review

Q9

Q9 Solution

Should the data be modeled/decomposed using an additive or multiplicative seasonal model?

Exam 02: Review

Q9

Q9 Solution

This is close; however, the range seems to increase with larger values. This can also be confirmed via:

```
forecast::hw(airlines_ts, seasonal = 'additive') |> forecast::accuracy()
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 97030.16 901289.7 704860.4 0.1505585 1.213241 0.3595306
##               ACF1
## Training set 0.003893482
```

```
forecast::hw(airlines_ts, seasonal = 'multiplicative') |> forecast::accuracy()
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 38967.76 824490.4 614646.3 0.05600333 1.066082 0.3135148
##               ACF1
## Training set 0.06342432
```

Exam 02 Review

Q10

Q10 Solution

Based on an EIA Article published on March 14, 2023 [Links to an external site.](#), the plot below shows the monthly averages of U.S. natural gas consumption per day over the past decade (i.e., for Jan 2013, we averaged 31 days of daily consumption). **The plot shows a clear seasonal pattern. Hence, seasonal decomposition can be used to forecast future values in 2023.**

Exam 02 Review

Q10


Q10 Solution

Decomposition methods are not suitable for forecasting per [Class 12 Slide 11](#).

Exam 02 Review

One possible solution for Q11 can be found online at: [exam 02 case study](#)

Quick Refresher from Last Week

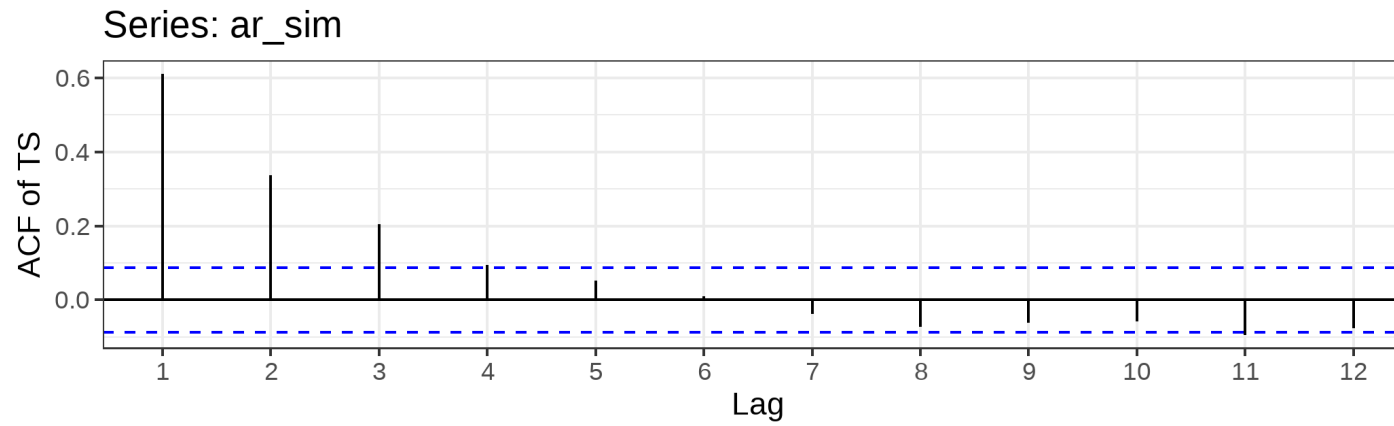
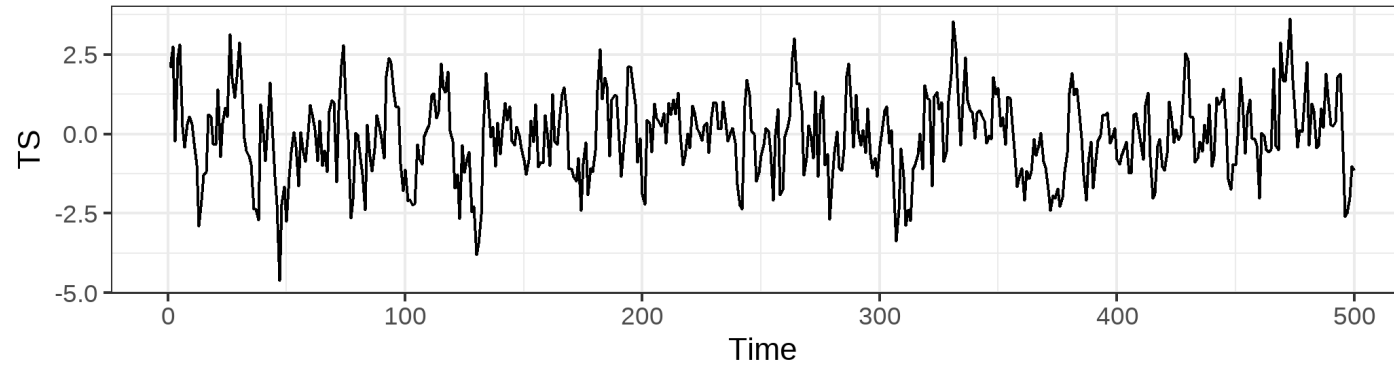
- ✓ Explain what do we mean by population/sample mean, variance, covariance and correlation (**review**).
- ✓ Explain the population autocovariance and autocorrelation.
- ✓ Compute sample estimates of the autocovariance and autocorrelation.
- ✓ Describe the large sample distribution of the autocorrelation function.
- ✓ Explain how sample (partial) autocorrelation is calculated.
- ✓ Use  to compute both the ACF and PACF.

Learning Objectives for Today's Class

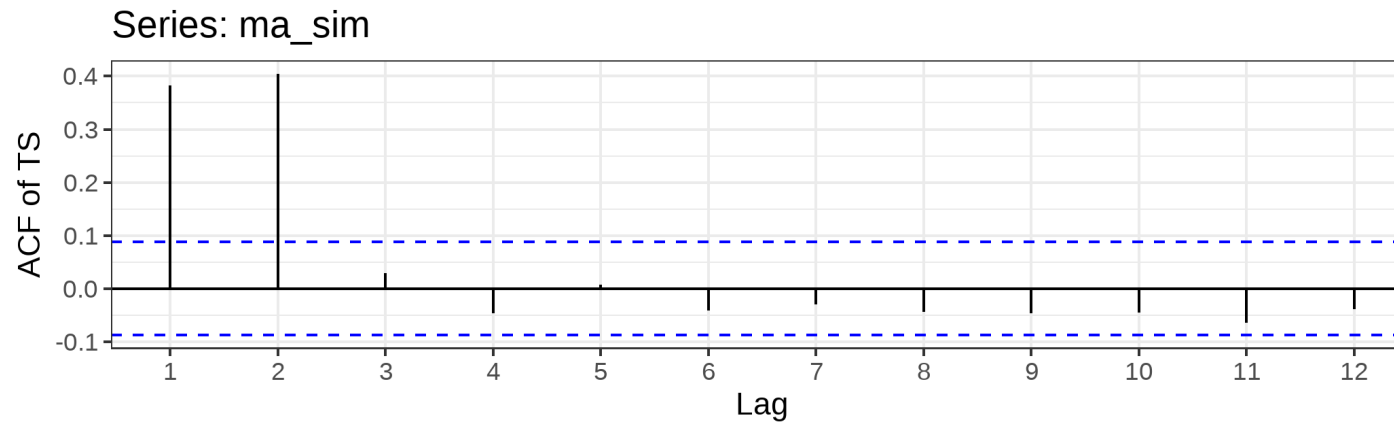
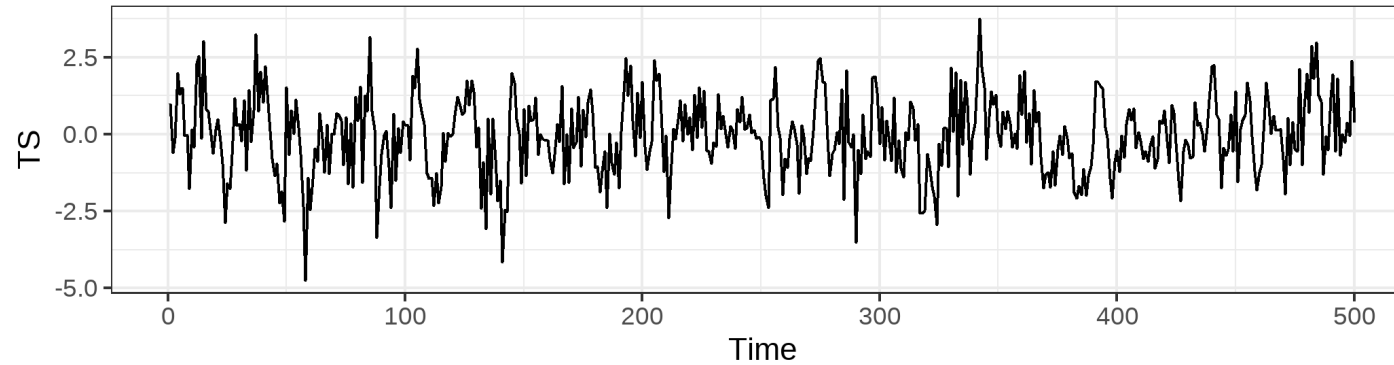
- Utilize time-series plots (line charts and ACF) to identify whether a ts is stationary.
- Apply transformations to a nonstationary time series to bring it into stationarity (**review**).
- Conduct formal tests for stationarity using the ADF and KPSS tests.

A Visual Exploration of Stationary and Nonstationary TS

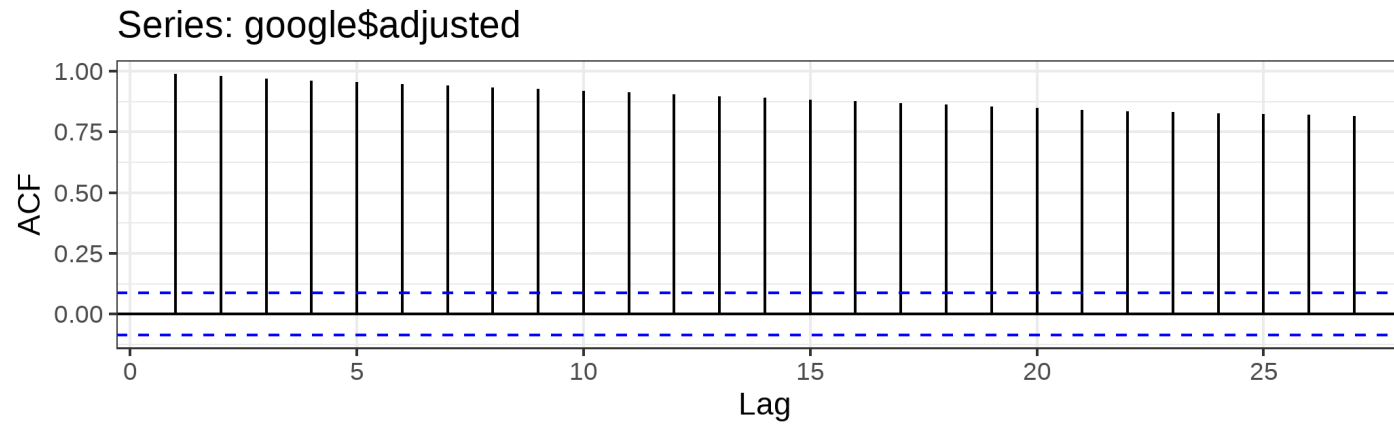
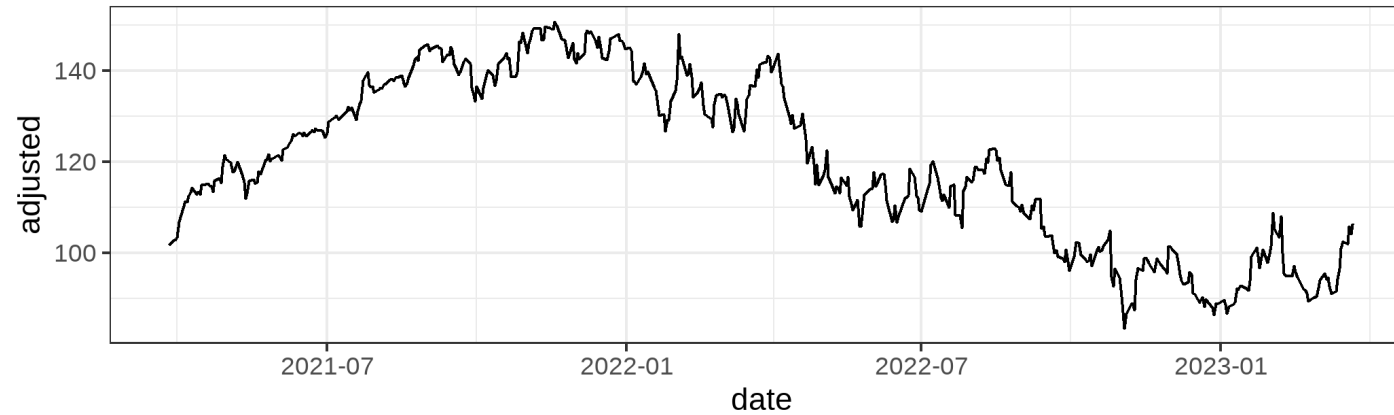
Stationary TS: ACF Dies Down



Stationary TS: ACF Cuts Off



Nonstationary TS: Random Walk



A Formal Definition for Stationarity

Weak Stationarity: A weakly stationary time series is a finite variance process such that:

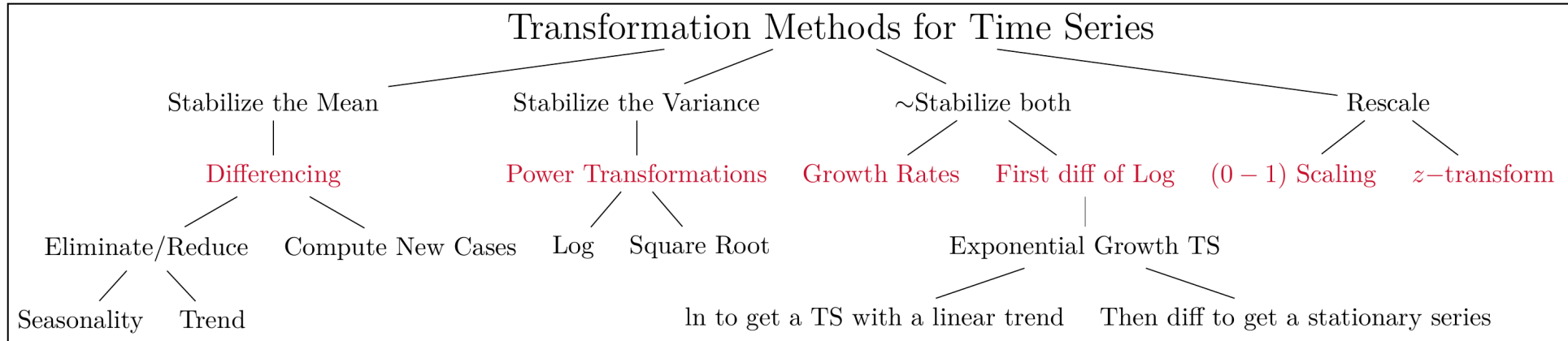
- The mean, μ_t , is constant and does not depend on the time t ; and
- The autocovariance function, $\gamma(s, t)$ depends on s and t only through their difference $|s - t|$.

We will use the term **stationary** to refer to weak stationarity.

- The concept of weak stationarity forms the basis of much of the foundation for time series modeling.
- The fundamental properties (1 & 2) required for weak stationarity are satisfied by many of the models that are widely used.

Differencing

Recap: Guidelines for Transforming TS Data



A classification of common transformation approaches for time series data

What to do when we have a Nonstationary series?

In order to model a time series, it must usually be in a state of stationarity. If the time series is not stationary, you must transform it to achieve stationarity.

Successive **differencing** is typically used to achieve stationarity.

First Differences:

$$y'_t = y_t - y_{t-1}$$

Second Differences:

$$y''_t = y'_t - y'_{t-1}$$

GNP Example

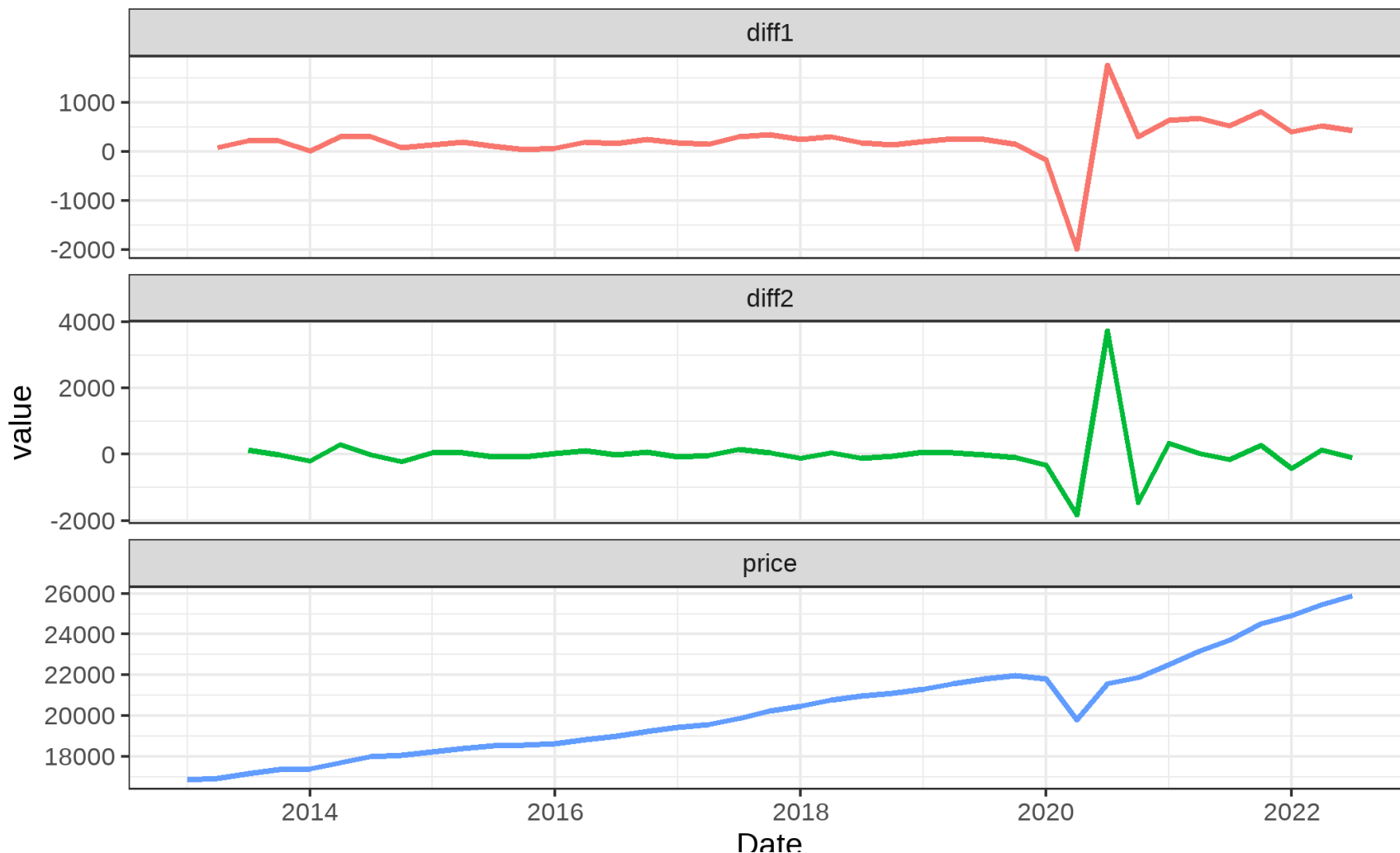
```
gnp = tidyquant::tq_get('GNP', get = 'economic.data')

gnp = gnp |>
  dplyr::mutate(
    diff1 = price - dplyr::lag(price),
    diff2 = c(NA, NA, diff(price, differences = 2))
  )

df = tidyr::pivot_longer(data= gnp, cols = c(3,4,5))

df |>
  ggplot2::ggplot(ggplot2::aes(x= date, y = value, group = name, color = name)) +
  ggplot2::geom_line(size = 1) +
  ggplot2::facet_wrap(~ name, ncol = 1, scales = "free_y") +
  ggplot2::theme_bw() +
  ggplot2::theme(legend.position = 'none') + ggplot2::labs(x = 'Date')
```

GNP Example



So Why Does Differencing Work?

Because many nonstationary time series have features of a random walk.

Random Walk Model

Random Walk with Drift: A random walk is a model such that successive differences (first differences) are independent.

The classic random walk model:

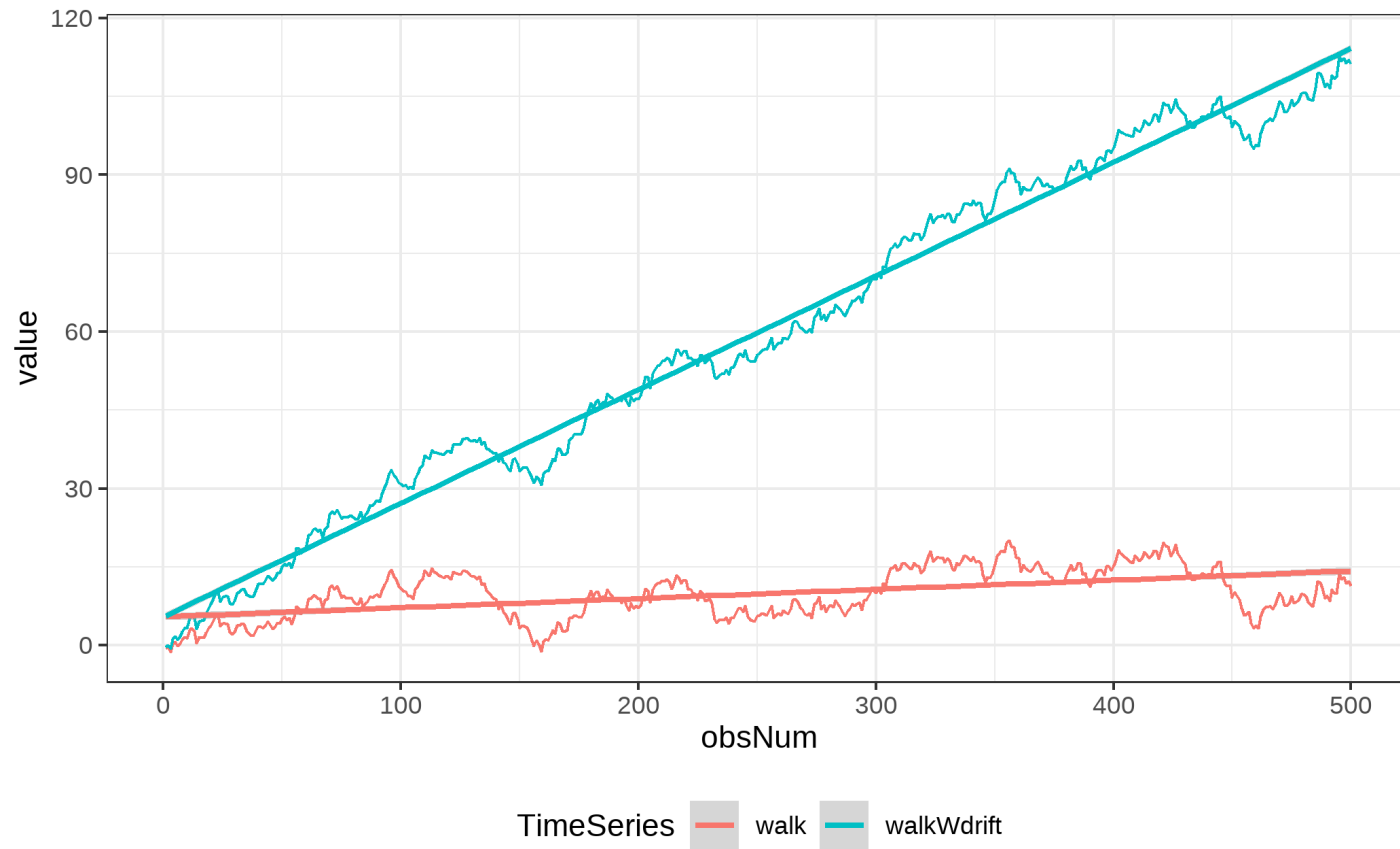
$$Y_t = Y_{t-1} + \epsilon_t$$

A random walk with a drift:

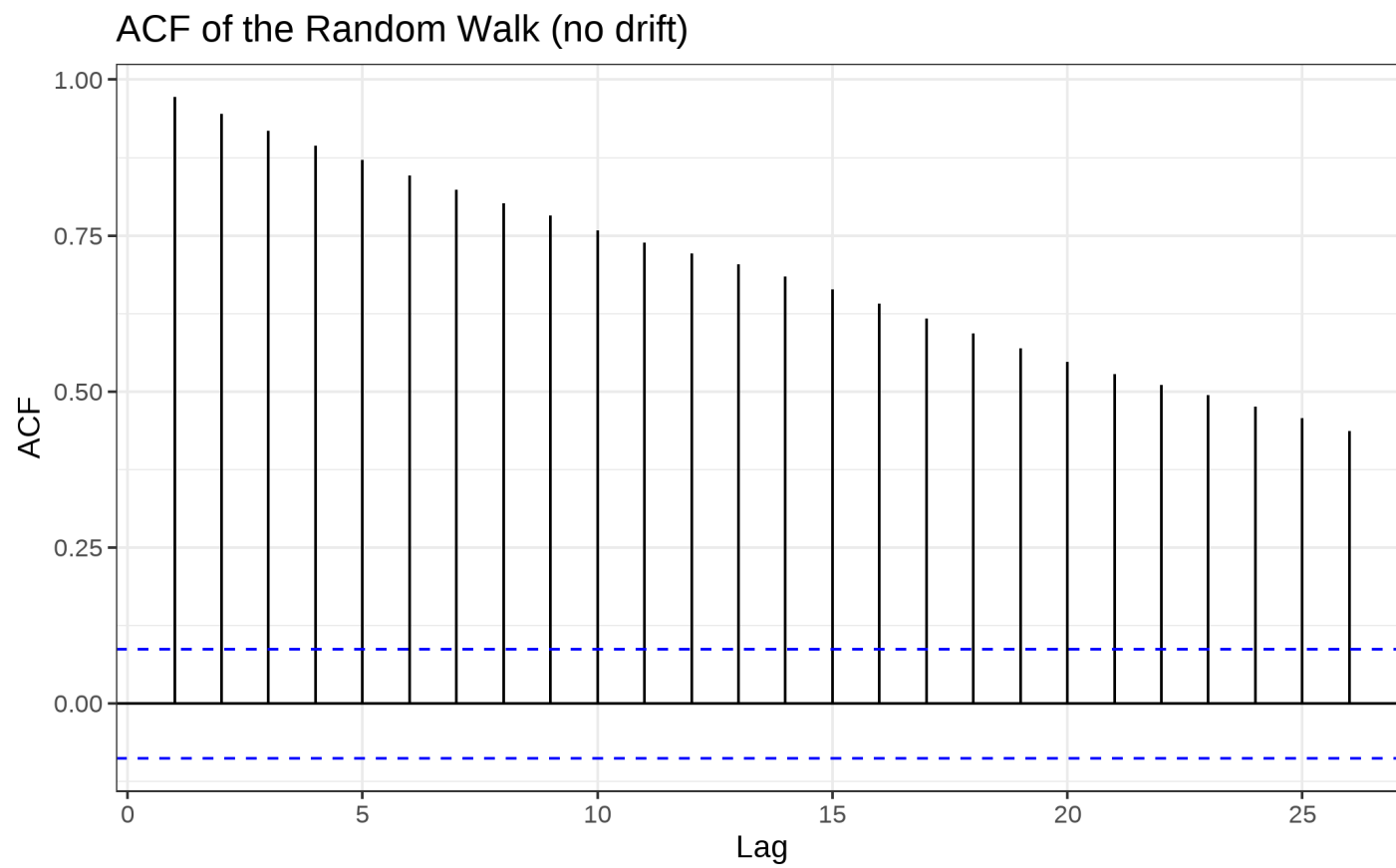
$$Y_t = \delta + Y_{t-1} + \epsilon_t$$

Notes: When $\delta = 0$, the value of the current observation is just the value of the prior observation plus random noise.

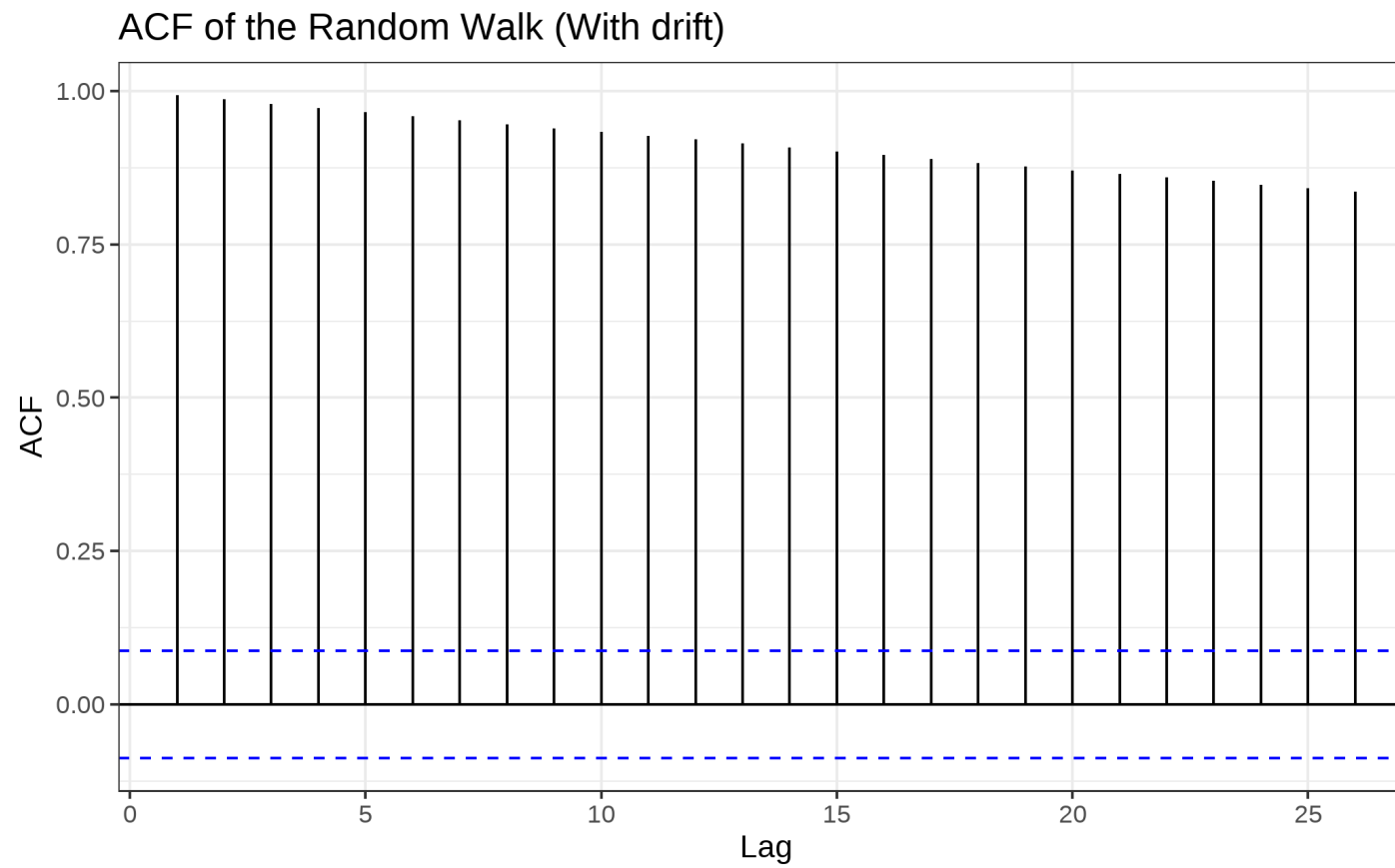
Random Walk



Random Walk



Random Walk



Formal Tests for Stationarity

Augmented Dickey Fuller (ADF) Test

The **Augmented Dickey Fuller (ADF) Test** tests whether or not there is a unit root. The hypotheses are as follows:

Ho: The series is nonstationary

Ha: The series is stationary

Thus, if we have a *SMALL p-value*, we reject the null hypothesis and conclude **STATIONARITY**.

Augmented Dickey Fuller (ADF) Test

```
gnp = tidyquant::tq_get('GNP', get = 'economic.data')
```

```
tseries::adf.test(gnp$price)
```

```
##  
##      Augmented Dickey-Fuller Test  
##  
## data:  gnp$price  
## Dickey-Fuller = -1.4239, Lag order = 3, p-value = 0.799  
## alternative hypothesis: stationary
```

So what do we conclude from the test above?

... Insert conclusion here ...

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

The **Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test** tests whether or not there is a unit root.

The hypotheses are as follows:

Ho: The series is stationary

Ha: The series is nonstationary

Thus, if we have a *SMALL p-value*, we reject the null hypothesis and conclude **NONSTATIONARITY**.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

```
gnp = tidyquant::tq_get('GNP', get = 'economic.data')
```

```
tseries::kpss.test(gnp$price)
```

```
##  
##      KPSS Test for Level Stationarity  
##  
## data:  gnp$price  
## KPSS Level = 1.0318, Truncation lag parameter = 3, p-value = 0.01
```

So what do we conclude from the test above?

... Insert conclusion here ...

Successive KPSS Tests

As a followup to the case when the `kpss.test()` is rejected (or alternatively when you do not reject the `adf.test()`), you can utilize the `ndiffs()` from the package `forecast`, which uses a series of the KPSS tests in a sequence to determine the appropriate number of first differences required for a nonseasonal time series.

`ndiffs()` returns the number of first differences needed to achieve stationarity according to the KPSS test.

```
forecast::ndiffs(gnp$price)
```

```
## [1] 1
```

According to the `ndiffs()` function, **1 successive differences are recommended to transform the GNP data into a stationary ts.**

Recap

Summary of Main Points

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

- Utilize time-series plots (line charts and ACF) to identify whether a ts is stationary.
- Apply transformations to a nonstationary time series to bring it into stationarity (**review**).
- Conduct formal tests for stationarity using the ADF and KPSS tests.

Things to Do to Prepare for Our Next Class

- **Required:** Complete [assignment12](#).