

# MTH 208 Exploratory Data Analysis

## Lesson 08: Time-Series Analysis

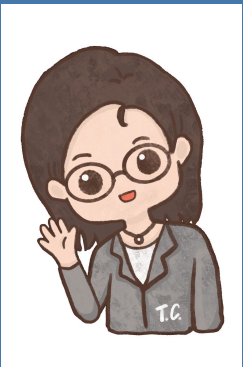
Ying-Ju Tessa Chen, PhD

Associate Professor  
Department of Mathematics  
University of Dayton

 @ying-ju

 ying-ju

 ychen4@udayton.edu



# Learning Objectives

- Introduction to Time-Series Data
- Graphic Displays for Time-Series Data
- Identifying Patterns in Time-Series Data
- Time Series Decomposition

# Introduction to Time-Series Data

## Definition of Time-Series Data

Time-series data is a sequence of data points collected or recorded at successive time intervals. These intervals can be regular or irregular but are typically uniform, such as hourly, daily, monthly, or yearly. Each data point in a time series represents the value of a variable or variables at that point in time, making the temporal ordering of these data points critical to their analysis and interpretation.

## Examples of Time-Series Data

- **Economics:** Gross Domestic Product (GDP) measured quarterly, inflation rates reported monthly, or stock prices recorded daily.
- **Weather Forecasting:** Temperature, humidity, or precipitation levels recorded hourly or daily.
- **Healthcare:** Patient heart rate monitored over minutes or hours, or the spread of a disease tracked daily or weekly across regions.
- **Energy:** Electricity consumption or production levels recorded hourly or daily.

# Importance of Time Series in Various Domains

- **Economics:** Time-series data is pivotal for understanding economic trends, assessing policy impacts, and forecasting future economic conditions. Analysts study patterns in economic indicators to predict recessions, booms, and other economic phenomena.
- **Weather Forecasting:** Accurate weather predictions rely heavily on analyzing time-series data of various meteorological variables. This analysis helps in preparing for severe weather events, agricultural planning, and daily life decisions.
- **Stock Market Analysis:** Investors and financial analysts use time-series data of stock prices and market indices to identify trends, calculate volatility, and develop trading strategies. Time-series analysis in finance is crucial for risk assessment and portfolio management.
- **Healthcare:** In the healthcare domain, time-series data from patient monitoring can provide critical insights for medical diagnosis, treatment planning, and understanding disease progression. It is also used in epidemiology to track the spread of diseases and evaluate public health interventions.
- **Energy Sector:** Time-series analysis helps energy companies and policymakers understand consumption patterns, plan for demand fluctuations, and optimize the generation and distribution of energy resources.

# Graphic Displays for Time Series Data

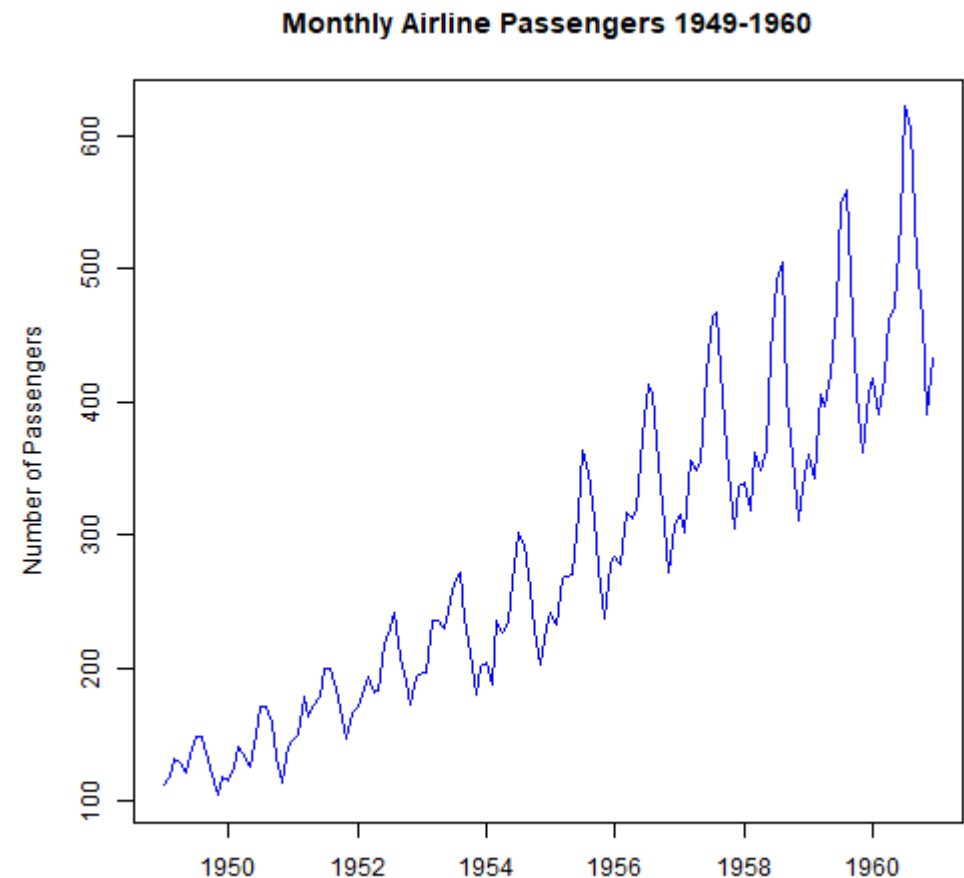
## Line Charts for Time-Series Data

- **Purpose:** Line charts are fundamental for displaying time-series data, illustrating how a variable changes over time. They are particularly useful for identifying long-term trends and patterns.
- **Key Features to Focus on**
  - **Trend:** Observe whether the variable is increasing, decreasing, or remaining constant over time.
  - **Seasonality:** Look for regular patterns or cycles that repeat over a specific period, such as yearly or monthly.
  - **Anomalies:** Spot any unexpected spikes, drops, or unusual patterns that deviate from the overall trend.

# An Example of Time Series Plot

We use the [AirPassengers](#) dataset, a classic time series of monthly totals of international airline passengers from 1949 to 1960, to show an example of time series plot. Here is the R code.

```
data("AirPassengers")  
plot(AirPassengers, type = "l", col = "blue",  
     main = "Monthly Airline Passengers 1949-1960",  
     xlab = "Year", ylab = "Number of Passengers")
```



# Graphic Displays for Time Series Data (Continued)

## Heatmaps for visualizing seasonality

- **Purpose:** Heatmaps can effectively visualize seasonality in time-series data, with color intensities representing the magnitude of a variable. They are excellent for spotting patterns across different timescales (e.g., days of the week or months of the year).
- **Key Features to Focus on**
  - **Color Variation:** Pay attention to changes in color to identify periods of higher or lower values.
  - **Seasonal Patterns:** Look for consistent color patterns across similar periods (e.g., summers warmer than winters).
  - **Outliers:** Identify any cells that have markedly different colors compared to the overall pattern.

# An Example of Heatmaps

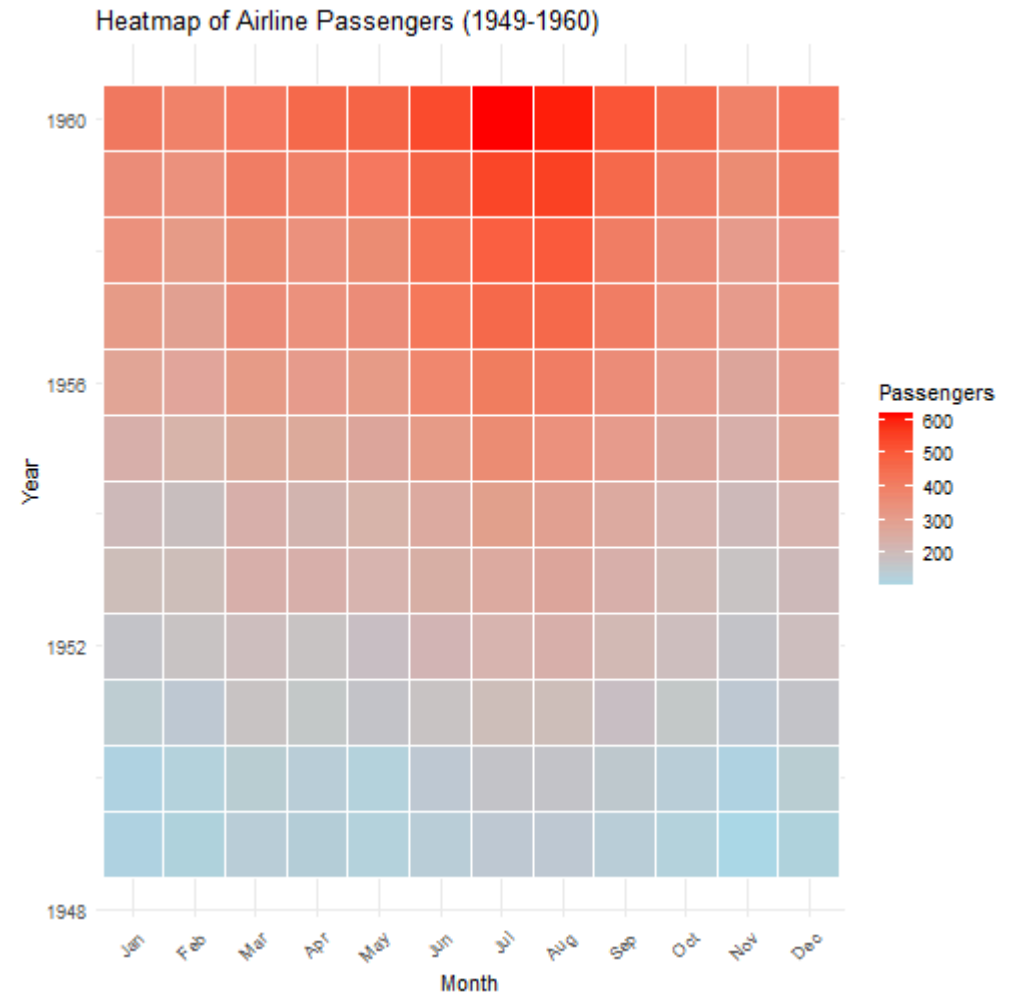
Visualizing the seasonality of the AirPassengers dataset by month:

```
library(ggplot2)
library(reshape2)

# Convert the AirPassengers time series to a data frame
ap_df <- data.frame(Month = time(AirPassengers),
                    Passengers = as.numeric(AirPassengers))

# Create a Year and Month column from the time series
ap_df$Year <- as.integer(floor(ap_df$Month))
ap_df$Month <- factor(month.abb[cycle(AirPassengers)],
                     levels = month.abb)

# Proceed directly to ggplot
ggplot(ap_df, aes(x = Month, y = Year,
                  fill = Passengers)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightblue", high = "red") +
  labs(title = "Heatmap of Airline Passengers (1949-1960)",
       x = "Month", y = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```





# Graphic Displays for Time Series Data (Continued)

## Lag Plots to Identify Autocorrelation

- **Purpose:** Lag plots are used to identify autocorrelation in time-series data, where a variable's current value is compared with its values at previous times (lags).
- **Key Features to Focus on**
  - **Scatter Pattern:** A linear pattern indicates autocorrelation; the stronger the linear relationship, the stronger the autocorrelation.
  - **Outliers:** Points that fall far from the general pattern may indicate anomalies.
  - **Randomness:** A cloud of points without a discernible pattern suggests little to no autocorrelation.

# An Example of Lag Plots

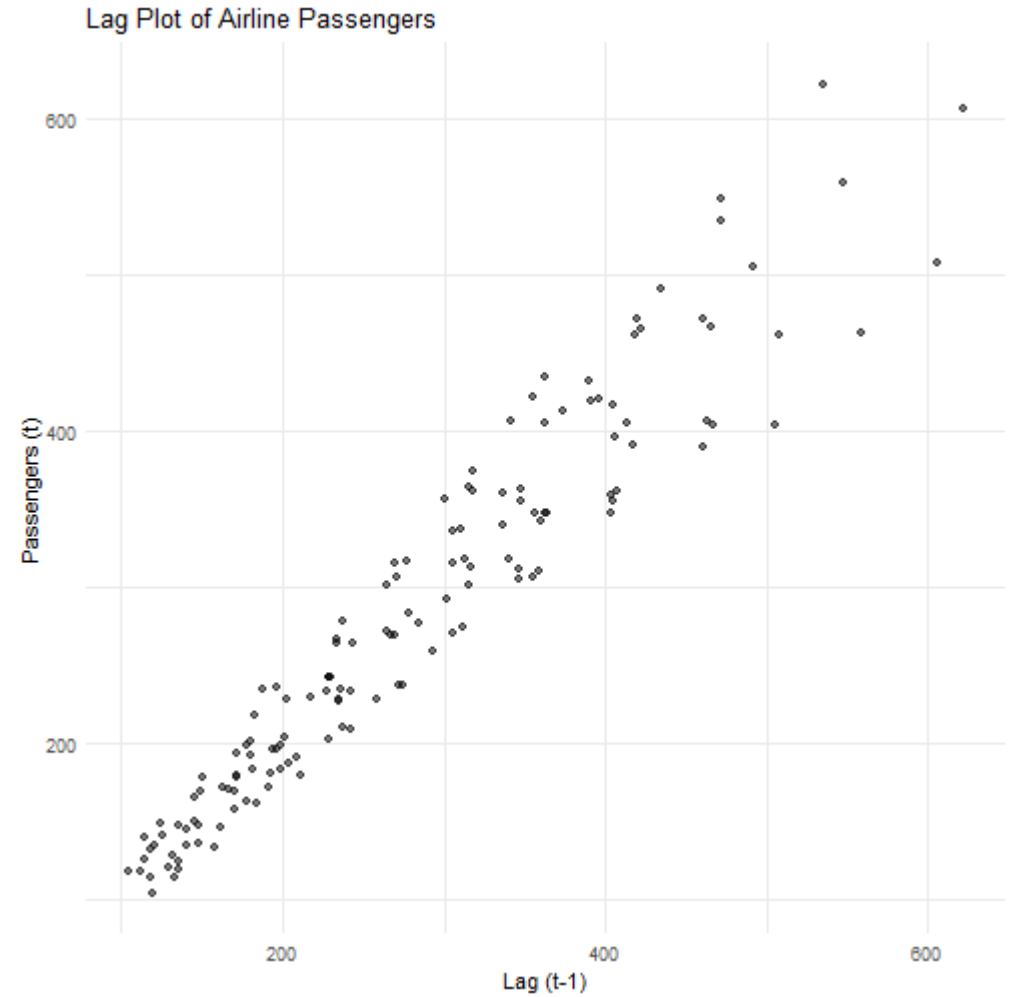
Creating a lag plot of the `AirPassengers` dataset:

```
library(ggplot2)

# Create a lagged version of the dataset
lagged_data <- cbind(AirPassengers,
  lag = c(NA, AirPassengers[-length(AirPassengers)]))

lagged_data <- as.data.frame(lagged_data)

# Plot
ggplot(lagged_data, aes(x = lag, y = AirPassengers)) +
  geom_point(alpha = 0.5) +
  labs(title = "Lag Plot of Airline Passengers",
    x = "Lag (t-1)", y = "Passengers (t)") +
  theme_minimal()
```



# Identifying Patterns in Time-Series Data

- **Trend:** The long-term movement in time-series data.
- **Seasonality:** Regular, predictable patterns within a fixed period, such as daily, monthly, or quarterly.
- **Cyclical patterns:** Fluctuations occurring at irregular intervals, influenced by economic or other factors.

# Advanced Insights into Time-Series Patterns

- **Decomposing Time-Series Data**

- **Purpose:** Break down a time series into its constituent components (trend, seasonality, and residuals) to analyze them separately.
- **Method:** Use methods like classical decomposition or STL (Seasonal and Trend decomposition using Loess) in R for flexible decomposition of different time-series components.

- **Smoothing Techniques:**

- **Purpose:** Smooth out short-term fluctuations to highlight longer-term trends or cycles.
- **Examples:** Moving averages, exponential smoothing, or LOWESS (Locally Weighted Scatterplot Smoothing). Each technique has its application, with moving averages being simplest for identifying the general direction of the time series and LOWESS/LOESS for more flexible fitting that can adapt to changes in the trend.

- **Differencing:**

- **Purpose:** Reduce or eliminate trend and seasonality in a time series to study the stochastic (random) components or make the series stationary.
- **Application:** Apply first differencing (subtracting the current period's observation from the previous period's observation) or seasonal differencing (subtracting the current observation from the observation at the same season last cycle) to remove trends and seasonality, facilitating easier identification of patterns.

# Incorporating Autocorrelation

## **Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF):**

These tools help identify the type of model that might best describe the time series. For instance, they can indicate if a series has autoregressive (AR) or moving average (MA) characteristics.

ACF shows the correlation of the time series with its own lagged values, while PACF shows the partial correlation of the series with its own lagged values, controlling for the values of the time series at all shorter lags.

## **References:**

- [Autocorrelation and Partial Autocorrelation in Time Series Data](#)
- [Interpreting ACF or Auto-correlation plot](#)

# Time Series Decomposition

**Introduction to Time Series Decomposition** Time series decomposition involves breaking down a time series into several components, each representing underlying patterns within the data. Typically, these components include the **trend**, **seasonality**, and **residuals** (irregular component). Decomposition allows analysts to understand complex time series by examining these simpler, underlying structures.

- **Trend:** The long-term progression of the data, showing an overall upward or downward movement. It represents the increase or decrease in the data's value over time.
- **Seasonality:** Regular and predictable changes that repeat over a calendar cycle, such as daily, monthly, or quarterly.
- **Residuals:** The irregular or stochastic component left after the trend and seasonal components have been removed from the time series. Residuals represent the randomness or noise in the data.

# Additive vs. Multiplicative Decomposition Models

- **Additive Model:** Assumes that the components of the time series add together as follows:

$$Y_t = T_t + S_t + R_t,$$

where  $Y_t$  is the data at time  $t$ ,  $T_t$  is the trend component,  $S_t$  is the seasonal component, and  $R_t$  is the residual component. This model is suitable when seasonal variations are roughly constant over time.

- **Multiplicative Model:** Assumes that the components multiply together:

$$Y_t = T_t \times S_t \times R_t.$$

This model is appropriate when the seasonal variations increase or decrease over time proportionally to the trend.

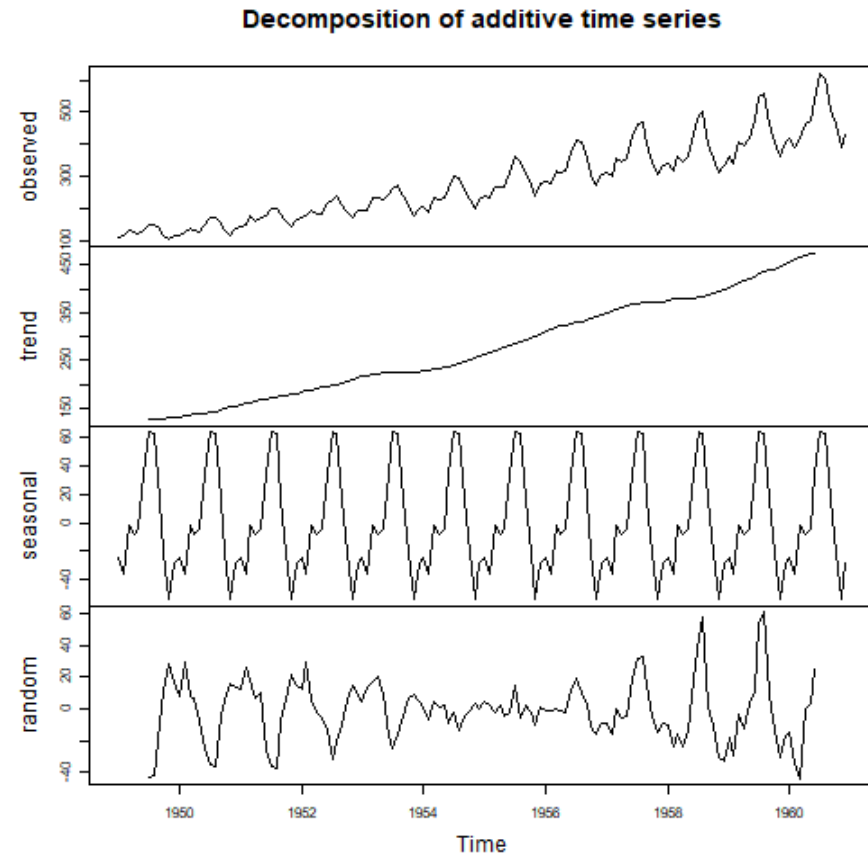
# Hands-on Practice with R's `decompose()` Function

The `decompose()` function in R provides an easy way to perform classical decomposition of time series into trend, seasonal, and random components, supporting both additive and multiplicative models. Below, we show an example using the [AirPassengers](#) Dataset.

```
# Load the AirPassengers dataset
data("AirPassengers")

# Decompose the time series using an additive model
decomp_add <- decompose(AirPassengers,
                        type = "additive")

# Plot the decomposed components
plot(decomp_add)
```

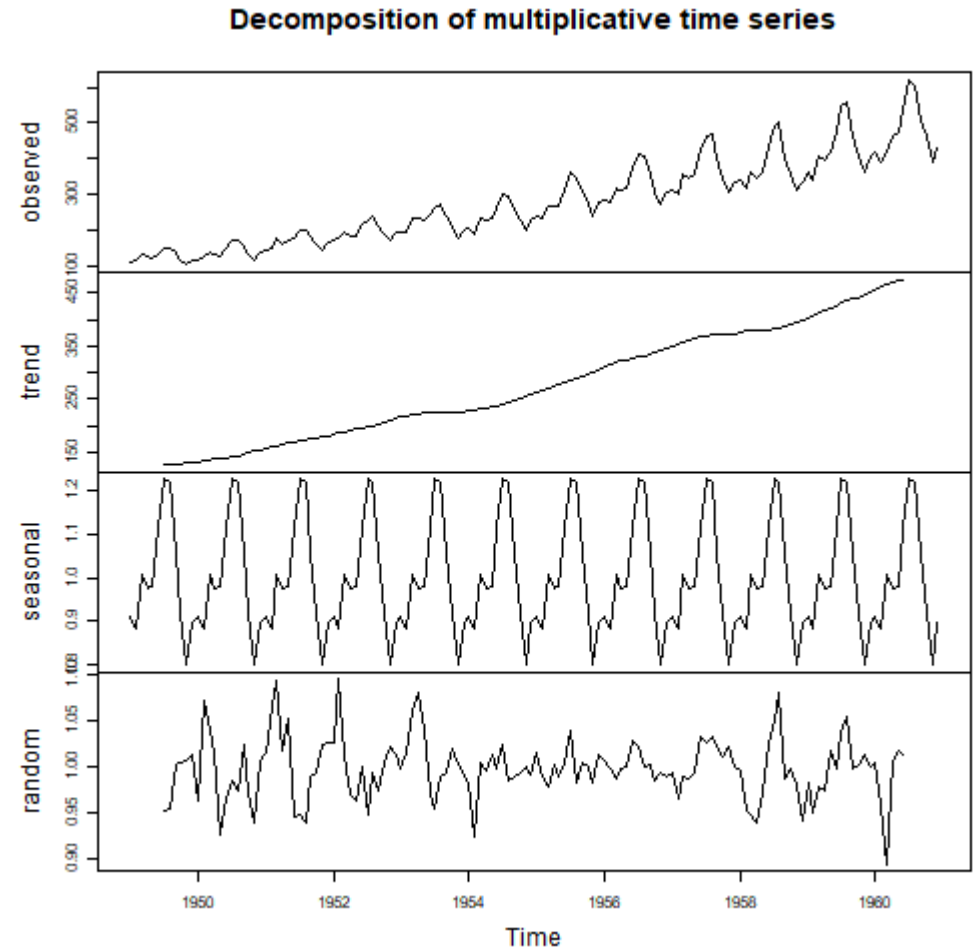




# Plot the decomposed components - Multiplicative Model

```
# Decompose the time series
# using a multiplicative model
decomp_mult <- decompose(AirPassengers,
                          type = "multiplicative")

# Plot the decomposed components
plot(decomp_mult)
```



# References

The lectures of this course are based on the ideas from the following references.

- Exploratory Data Analysis by John W. Tukey
- A Course in Exploratory Data Analysis by Jim Albert
- The Visual Display of Quantitative Information by Edward R. Tufte
- Data Science for Business: what you need to know about data mining and data-analytic thinking by Foster Provost and Tom Fawcett
- Storytelling with Data: A Data Visualization Guide for Business Professionals by Cole Nussbaumer Knaflic