Advanced Data Analysis with R - Part Time Series Analysis

Summer Term 2025

Johannes, Sebastian, and Kai (held by Kai)

Table of contents

Preface

Welcome

to the time series part of the course advanced data analysis with R. In the **three** time series lessons, we will

- **understand** why time series are an exciting type of data for us and where we usually come in touch with them,
- get familiar with the **properties** of time series data and with their most relevant differences to other types of data that we already know,
- learn how to analyse time series data **descriptively** and with simple **time series regression models**,
- and we will learn how to account for/ **correct for** time-dynamic covariates in regression models.

All you need is this document and the respective data. You find both on GitHub. However, this document will probably develop within the next few weeks. I let you know, once it is finalised. Credits to Jasper Fuchs (ETH Zürich) who revised the first version of the material.

Who I am

- Kai Husmann
- Department Forest Economics and Sustainable Land-use Planning (Prof. Carola Paul)
- Projects and topics of Forest Econometrics
- Contact: kai.husmann@uni-goettingen.de and StudIP

Where are we in the course

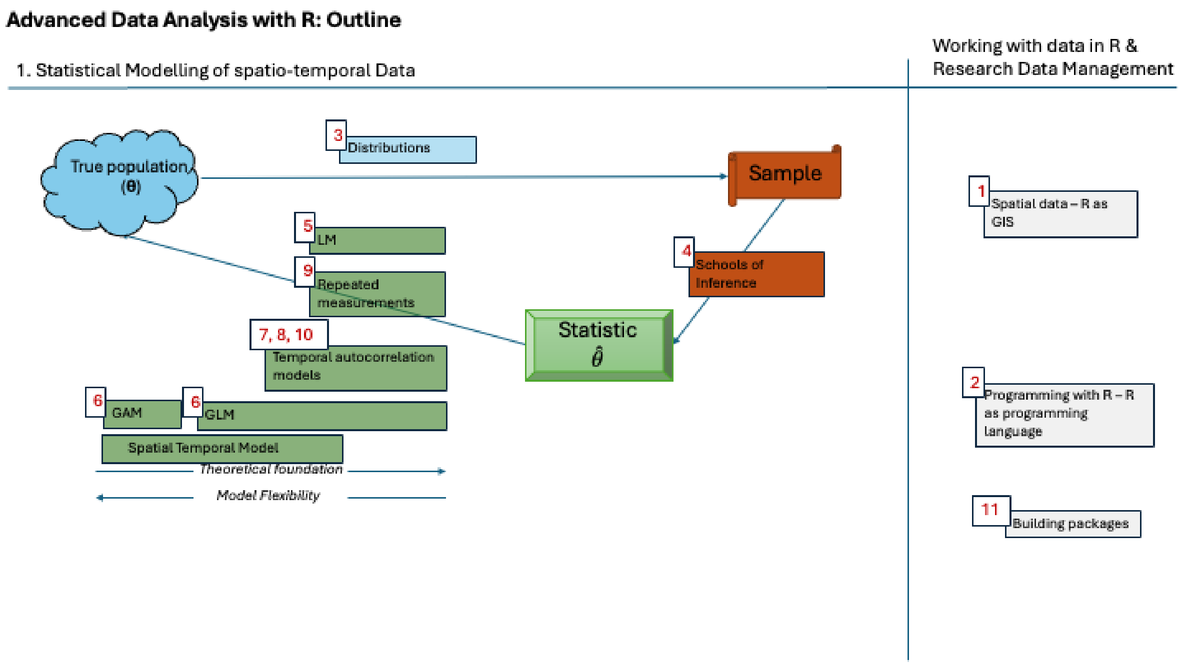


Figure 1: Overview. The red numbers refer to the weeks in which the topics are discuddes.

Motivation for Time Series Analysis

Time series data are ubiquitous in many fields, including economics and finance (where most of today's methods originate from), biology, and environmental sciences. In the context of resilience and resilience of ecosystems, time series data and time series methods have also become increasingly important in the context of ecology, agriculture, and also **forestry**. Time series methods are particularly promising in this area, as the time pattern (direct response, delayed response, …) and the time horizon (how long is the recovery period, is there a recovery, …) of the responses of ecosystem variables to disturbances are usually the main interest. Furthermore, many ecosystem variables themselves show a time trend. As time series models have evolved from the field of economics, they are also in a forestry context often used to describe the dynamics of economic variables, such as market reactions in the sense of e.g. *how does the (timber) price react on supply and demand changes?* and does this relationship persist sudden and extreme supply changes (e.g. due to storms) (e.g. Fuchs et al. 2022)? Is it resistant and resilient to calamities? Time series models are more and more used to describe the dynamics of ecological variables as well, such as the relationship between tree growth and climate variables, or the relationship between tree mortality and tree health variables (e.g. Lemoine 2021).

Time series exercise 1

Consider Figure 2.

1. Do you think, there is a relation between harvested volume and revenue or between share of damaged wood and revenue?
2. How would you analyse these relationships? Suggest a statistical model that you are already familiar with.

Following Lütkepohl and Krätzig (2004, 1), a time series is a sequence of observations of one variable over a period in time. The observations are thus ordered in time and usually have equal observation frequency. Most economic measures, like the gross national product, wood prices, or wood material flows, are often provided at an annual base. In contrast, meteorological data, like temperature or precipitation, are often provided at a daily base, or even more frequently. Ecosystem data, like tree dimension's measurements or tree health data, are seldom found in a frequency higher then annually. The forest health survey (crown condition monitoring) in Germany (Waldzustandserhebung) e.g. takes place every year, while the national forest inventory (Bundeswaldinventur) is conducted every 10 years only. This may also explain why the data availability of economic variables is better than that of ecological variables.

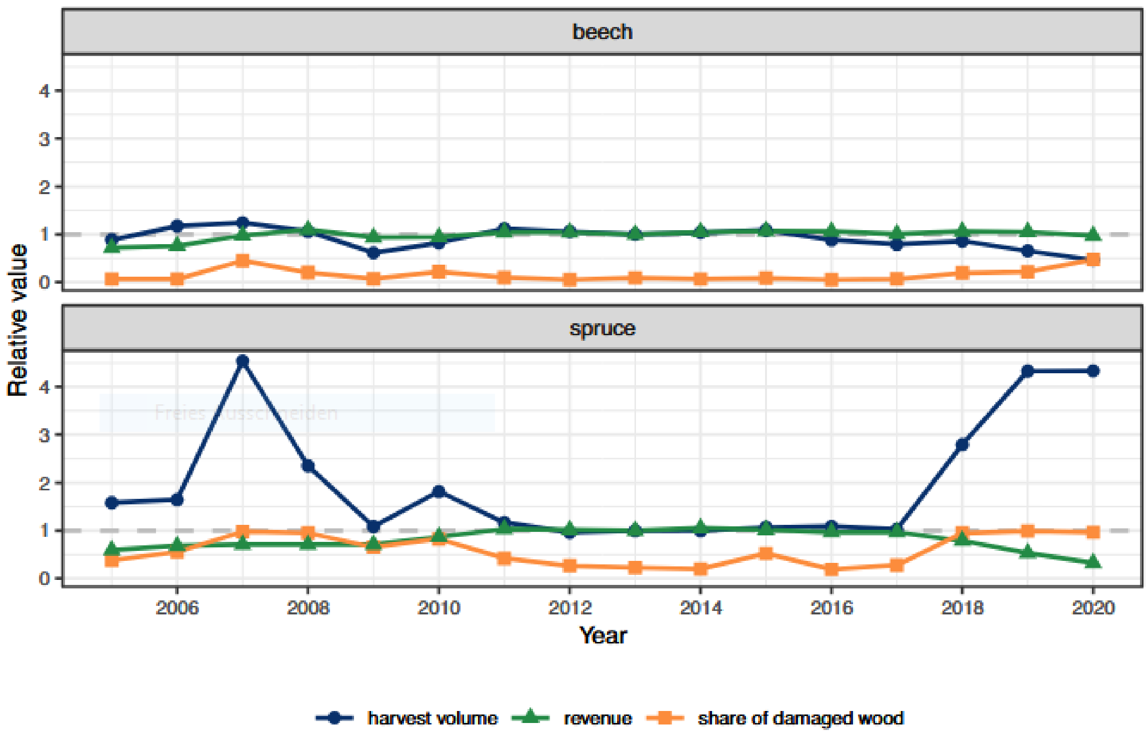


Figure 2: Fuchs et al. (2022): *Is there a relation connection between harvest volume, wood revenue and share of damaged wood? What is your guess?*

The challenge to measure variables in an even frequency without changing the measuring or estimation principles is a higher challenge in environmental sciences than it is in economics.

As with other types of data, time series data can be used in a regression context to describe correlations of the past, to forecast the future, or to estimate parameters for further use in e.g. causal simulation models. In addition, time series are used to integrate or correct for temporal dynamics (autocorrelation) in regression models, particularly in ecosystem sciences. In dynamic ecosystems, the relationships between the variables of interest are often confounded by temporal dynamics. If we want to infer the relationship between crown defoliation and precipitation, for example, we need to consider the state of crown defoliation in the past (diseased trees with high defoliation in the last year will never be 100% healthy in the current year, even if precipitation is currently sufficient).

Typical time series projects start with a descriptive analysis of the time-dynamic patterns (Lütkepohl and Krätzig 2004, 5), whereby, in contrast to the previous descriptive analysis, an important aspect is whether the data are actually time series data.

The typical issues of interest in time series analysis are to do

- descriptive statistics of time-dynamic patterns,
- filtering (however, we won't do this in this course),
- hypotheses testing/ statistical inference,
- forecasting, and
- accounting for time-dynamics of covariates (autocorrelation) in regression models

In the three time series lessons, we will introduce some of the most common methods of univariate time series analysis and provide practised examples in R.

Topic 1: What is a time series?

- Examples of time series
- Relevant **properties** and **assumptions**
- **Differences** (and similarities) to other data types
- Concept of **stationarity**
- Detection of autocorrelation
- Practised programming features for time series in `R`

Topic 2: Analysis of time-dynamic patterns - Detection of autocorrelation - **Descriptive** statistics (classical decomposition) - Statistical **modeling** (exponential smoothing) - Hypotheses **testing** and causality

Topic 3: Accounting for autocorrelation in linear mixed models

- **Detection** of autocorrelated residuals in ordinary models
- Most common **procedures**

Note

With this in mind. What are your wishes and expectations on the course. Let me know by next week.

103 https://flinga.fi/s/FQ3KSVC

Properties of Time Series Data - What is a Time Series?

We start with an example. The formal properties of time series are illustrated using the example of the wood price of oak (Quercus robur and Quercus petraea) in Germany. Data taken from https://www-genesis.destatis.de/ (Code: 61231-0001). You find it as `stemwood_prices_annually.csv` in the data folder. We also use weather data of the weather station Göttingen (`month_mean_temp_goe.csv`, https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/monthly/kl/historical/).

This chapter introduces the specific properties of time series data. We will compare the time series as a random variable with other types that you are already familiar with. We introduce the most famous descriptive methods for time series data, thereby introducing the concept of autocorrelation. By doing so, we will also introduce and discuss some practical tools for data handling and visualization of time series data in `R`. We will already discuss, which attributes models need to bring to analyse univariate time series data and in which situations the time series properties have to be considered in regression models. Remembering Figure 3, we have three stages in the process of statistical inference. The population, which we usually want to make estimations for, the sample, which we actually have, and the estimated population, which allows us to create simulation based confidence intervals and which we used to illustrate the concept of unbiasedness (see also Chapter 4 Random Variables).

So far, a central assumption was that all observations from the population come from an arbitrary but common distribution and can be independent sampled. This assumption is not valid for time series data, as the observations are ordered in time and thus not independent from each other. In the case of normal distribution, e.g. $X_1, \dots, X_n \overset{iid}{\sim} N(\mu, \sigma^2)$. This independence enabled unbiased estimates of the unconditioned average, e.g. the arithmetic mean, the unconditioned deviation, e.g. the standard deviation (Chapter 5 Random Variables), and to regress the data with other random variables (Chapter 5 Statistical Inference and 5 The Linear Model). In time series data, however, the observations are not independent, and the assumption of independence is violated. This prohibits to calculate averages and deviation that do not consider this dependency. Regressions with other covariates would be confounded by that dependency. All these methods would lead to biased estimates. However, we will not provide the proof of biases estimates in this course. Nor will we cover the simulation of time series. However, it is possible to simulate a time series by a *Brownian Motion* (e.g. Hamilton 2020, chap. 17.1 and 17.2) if you are interested.

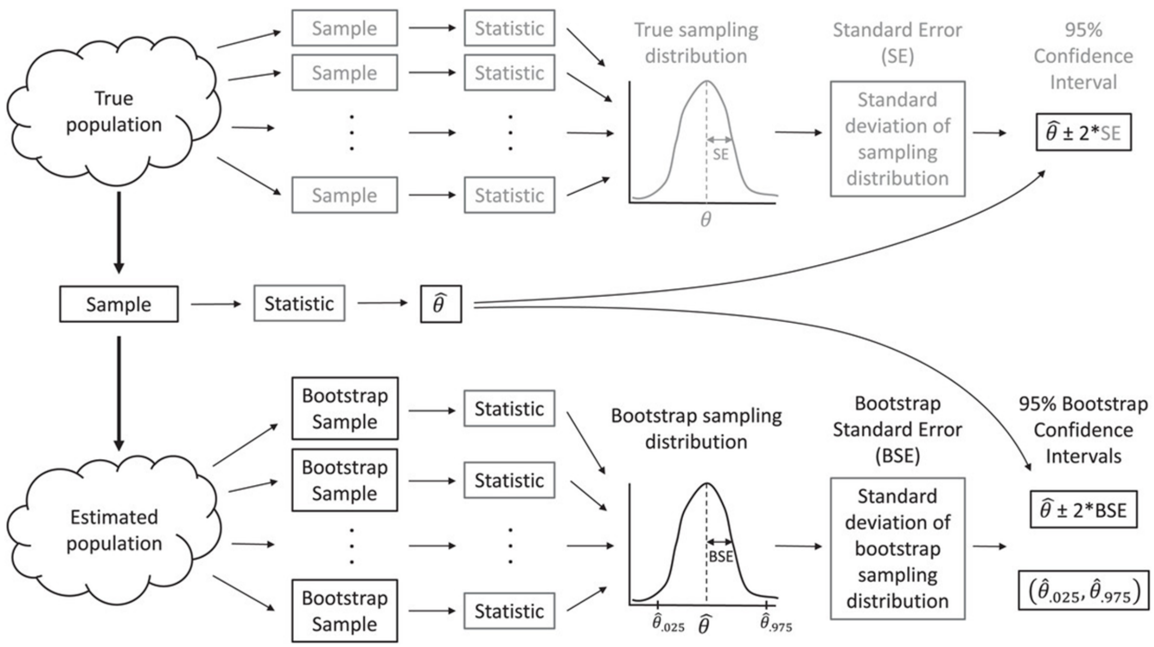


Figure 3: Fieberg, Vitense, and Johnson (2020) Resampling-based methods for biologists. See Chapter Resampling-based methods.

In practice, the arithmetic mean will estimate the center point of a time series (remember the Central Limit Theorem) but ignore the dependent part of the data, i.e. the autocorrelation. The same applies for the standard deviation, which will be constant over time. Consider e.g. the stem timber price index of oak (Figure 4). The arithmetic mean is usually not a suitable descriptive statistic for time series data. Instead of *what is the average of the data*, questions like *is there a seasonal trend?* or *To what extent does the data from the past describe my current situation in terms of time horizon and relevance?* are relevant when analysing time-dynamic data.

Ask yourself

- Do I expect autocorrelation in the data?
- Does it possibly confound my statistic of interest?
- Am I interested in analysing the autocorrelation?

Another typical question could be *is there a linear trend?*, which brings us back to the ordinary linear regression (Chapter 6). The linear regression is suitable to describe the global linear trend of a time series (Figure 5). It will thus detect a long-term development of a series. However, the autocorrelation is ignored. Thus, typical question like *how is my recent observation related to last observations of my series?* or *Is there a seasonal pattern?* cannot be analysed

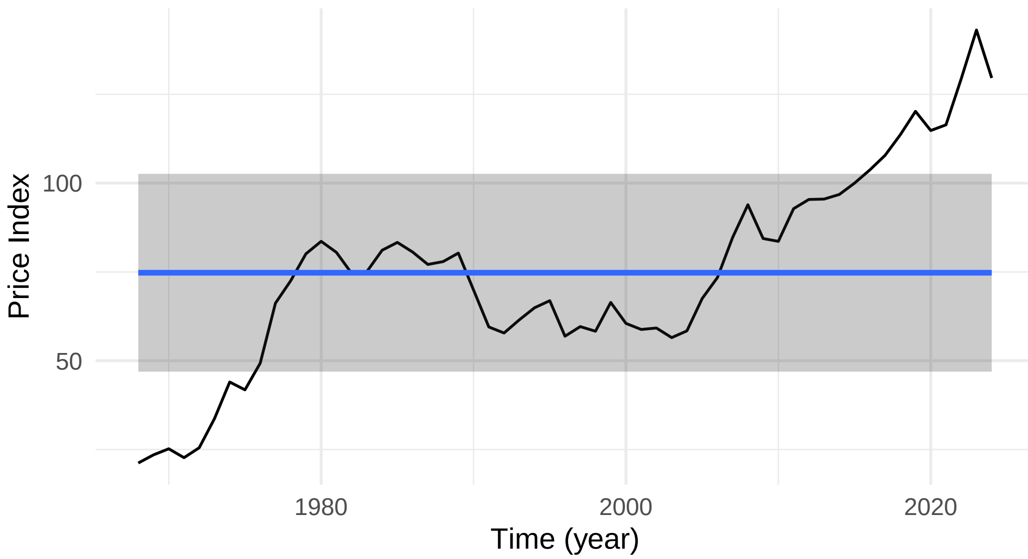


Figure 4: The black line shows the stem timber price index of oak (Quercus robur and Quercus petraea) in Germany from 1968 - 2024. Data taken from https://www-genesis.destatis.de/ (Code: 61231-0001). You find it as `stemwood_prices_annually.csv` in the data folder. The blue line shows the arithmetic mean, and the grey band shows the standard deviation.

by linear regression[1].

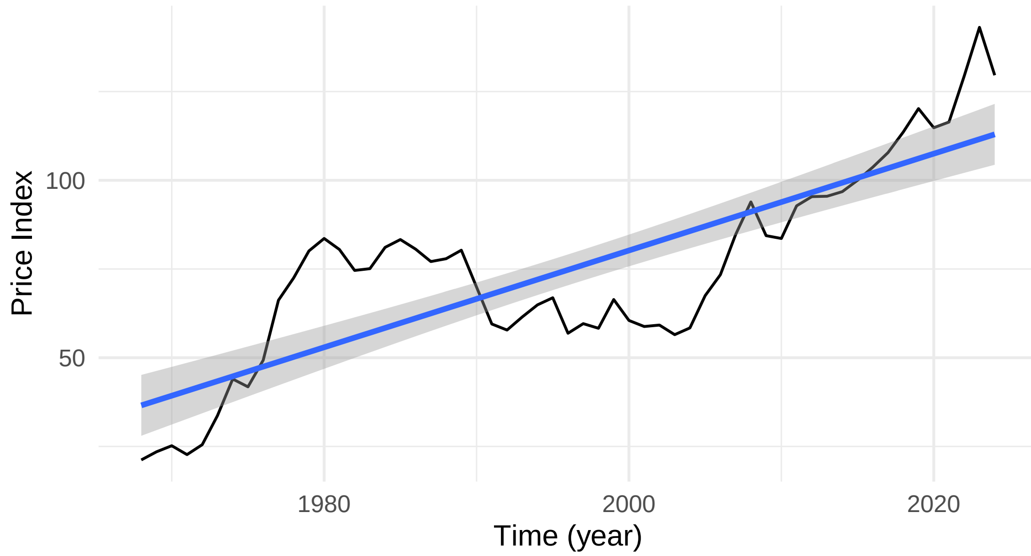


Figure 5: The black line shows the stem timber price index of oak (Quercus robur and Quercus petraea) in Germany from 1968 - 2024. Data taken from https://www-genesis.destatis.de/ (Code: 61231-0001). You find it as `stemwood_prices_annually.csv` in the data folder. The blue line shows the linear regression and its standard error (grey).

More formally, a time series is a sequence of T observations $y_t, t = 1, 2, \dots, T$ that are ordered (dependent) in time an which emerge from one random variable (Lütkepohl and Krätzig 2004, 11). Considering this ordering and some heterogeneity assumptions that we will come back to later, in a time series, any observation at any time t is a (so far unknown) function of its history as

$$y_t = f_t(t, y_{t-1}, y_{t-2}, \dots).$$

If we consider this time dependent function as a common function over the entire series, the discrepancy between this function and the actual observation is a stochastic component u_t, which is usually assumed to be an iid error process with mean zero and constant variance σ^2. Thus, the function can be rewritten as

$$y_t = f(t, y_{t-1}, y_{t-2}, \dots) + u_t,$$

[1]Note that many time series methods are actually specific variants of linear regression. We use the term *linear regression* here to mean *ordinary linear regression* without any correction, generalized term, mixed term, etc.

which means that the entire time series can also be described by a function f and a stochastic component u_t, just as we can do it for any regression. In practice, the function f is limited to a significant lag order P, thus

$$y_t \approx f(t, y_{t-1}, y_{t-2}, \dots, y_{t-P}) + u_t.$$

Theorem 0.1. *This representation allows to further distinguish f into a deterministic part $g(t)$ and an autocorrelative part, as*

$$y_t \approx g(t), \alpha_1 y_{t-1} + \alpha_2 y_{t-2}, \dots, \alpha_P y_{t-P} + u_t.$$

$g(t)$ is able to capture e.g. seasonality and/ or a common linear trend and/ or a constant. $g(t)$ captures all components that commonly apply independent from recent historic observations. A time series model with a linear trend (and optionally a constant) and without autocorrelation is thus an ordinary linear model (optionally with intercept) (see Figure 5). The linear regression in the example was able to describe this common trend, but the temporal dynamics remained in the residuals u_t.

Exemplary Time Series and Components of Time Series

When creating time series models, it is particularly important to analyse the characteristics of the series and also to take into account the theoretically assumed characteristics, as different models exist for different data-generating processes in time series statistics (Lütkepohl and Krätzig 2004, 8). The most relevant components that are to be investigated or hypothesised prior modelling are the

- constant components (intercept and/or slope),
- the seasonal component, and
- the autocorrelation component.

We use another example to illustrate the three components and thereby also learn some features for practised programming in `R`. `R` accounts for the properties of time series and provides functions for practical programming with time series data, which enables an effective working flow, beginning with a time series data type. The native function `ts` converts a data frame into a time series data type. `ts` requires the data to be ordered and a regular time pattern.

```r
stemwood_prices <- read_csv2("data/stemwood_prices_annually.csv")
stemwood_prices <- stemwood_prices |> select(-time) |>
  # Time is not required any more as a column as it is included in the ts object
  ts(start = min(stemwood_prices$time), frequency = 1)
# frequency = 1 as we have annual data
```

The `autoplot` function is a wrapper for the `ggplot2` package, which provides complete plots for particular data types. The class of the object transmitted to the function determines the type of the plot, which can then be further modified using the well-known `ggplot2` syntax. To include the time series feature in `autoplot`, also the `forecast` package is required. `forecast` is a package that contains numerous tools for time series analysis. To get an nice overview over the time series of the prices of stem wood for oak, beech, and spruce, for example, we can use the `autoplot` as follows.

```r
library(forecast)
plot_stemwood_prices <- stemwood_prices |>
  autoplot(facets = TRUE, colour = TRUE)
```

Adding elements that might help interpreting the time series data, such as vertical lines, can be done straightforwardly using `geom_vline`. The `annotate` function can be used to add labels to the plot. In the following example, we add the most severe storm events after 2000.

```r
plot_stemwood_prices <- plot_stemwood_prices +
  ylab("Price Index") +
  guides(colour = "none") + # legend not necessary as the facets are annotated
  geom_smooth(method = "lm") + # Add linear trends
  theme_minimal() +
  geom_vline(xintercept = c(2000, 2007, 2018)) + # Add storm events
  annotate(x = 2000, y = +Inf, label = "Lothar", vjust = 1, geom = "label") +
  annotate(x = 2007, y = +Inf, label = "Kyrill", vjust = 1, geom = "label") +
  annotate(x = 2018, y = +Inf, label = "Friederike", vjust = 1, geom = "label")

plot_stemwood_prices
```

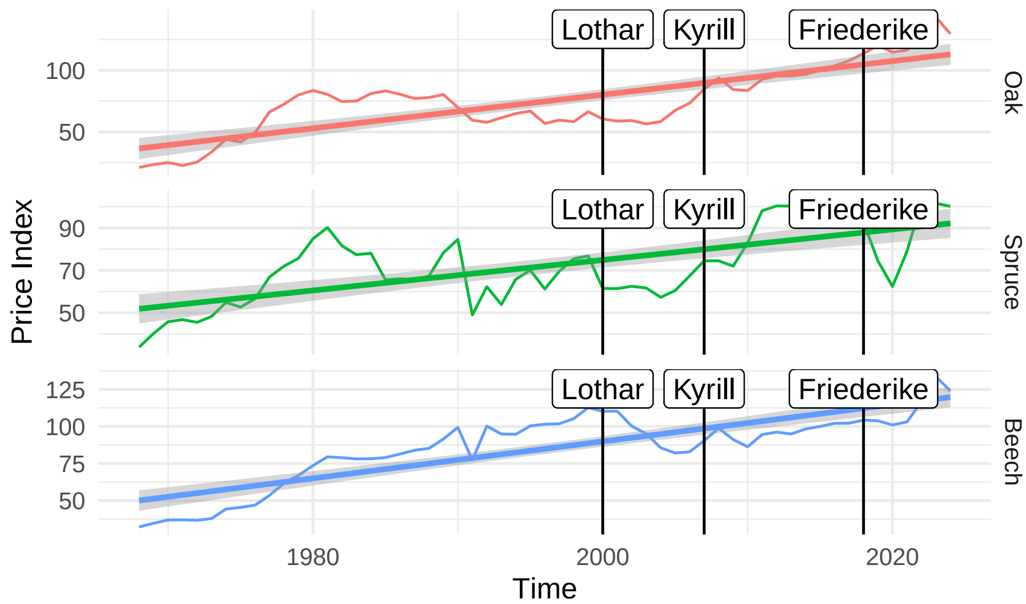


Figure 6: Oak, spruce and beech stem wood price indices in Germany from 1968 - 2024. Data taken from https://www-genesis.destatis.de/ (Code: 61231-0001). An example of three series with similar global trend but differing autocorrelations.

Time series exercise 2

Now it's your turn. Please organize yourself in small groups of 2 - 3 students and choose one species (Norway spruce, Scots pine or European beech) from the forest defoliation data set per group (see Chapter 2_datasets.pdf on Github for more details of the data). It would be great if we would cover all species. Please coordinate with the other groups to ensure that all three types are analysed.

1. Data preparation

- Load the data set into the variable `dat`.
- Filter it to your species.
- Create a univariate time series with the mean loss for each year. Call your `ts` object `dat_*[your species]`.

2. Visualization

- Plot your time series using `autoplot`.

197

According to the UBA (Umweltbundsamt) (https://www.umweltbundesamt.de/themen/wasser/extremereignisseklimawandel/trockenheit-in-deutschland-fragen-antworten#trockenheit-aktuelle-situation), the years 2018, 2019, 2020 and 2022 have been severely dry within the time horizon from 1990 to 2023.

3. Interpretation

- Emphasize the drought years 2018, 2019, 2020 and 2023 in your plot.
- Please present your plot to the colleagues. What are the 3 main findings of your plot?

4. Save your workspace.

Components of Time Series in More Detail

It can be seen that all three series increase by trend (*trend* component), which is emphasised by the 3 linear trend lines. It can also be seen that the series differ fundamentally in terms of their short-time dynamic patterns (autocorrelation). Additionally to the trend, there appears to be a correlation between the observations, a time-dynamic which on a time horizon shorter than the trend. This short term dynamics seem to be different among the three species, in contrast to the trend. While the price index for oak stemwood is relatively stable in terms of short-term time-dynamic pattern (see also Figure 5) and does not react on the events displayed, spruce is more sensible to dynamic pattern including a very severe storm reaction (Friederike) that led to a price decline to the index of 1975. Visual inspection shows that there are obvious trend components and that there might be *autocorrelative* components as well. Seasonal components cannot be followed from this figure. However, the data is annual, and thus, the seasonal component is not expected to be visible in the plot.

Industrial wood could be hypothised to have a *seasonal* trend, as the demand for wood is prospectively higher in winter than in summer, since it is often used as energy wood. In forestry, the timber sales prices are usually negotiated on a long-term basis, meaning that short-term demand and supply rarely have a direct impact and that possible seasonal trends are therefore masked (Fuchs et al. 2022). However, the higher the quality of the wood, the more this applies. Among the prices of all wood assortments, the industrial wood price is thus most likely to have a seasonal component. The time series also shows a linear trend, but evolves more slowly and appears to have a much stronger autocorrelative component. There may also be a seasonal trend, which appears to masked by autocorrelative trends in periods with higher fluctuation.

```r
ind_prices <- read_csv2("data/industrialwood_prices_monthly.csv")
ind_prices <- ind_prices |> select(Spruce) |>
  ts(start = min(ind_prices$year), frequency = 12) # Monthly data

ind_prices |>
  autoplot() +
  guides(colour = "none") + # A legend is not necessary.
  theme_minimal() + geom_smooth(method = "lm") + ylab("Price Index")
```

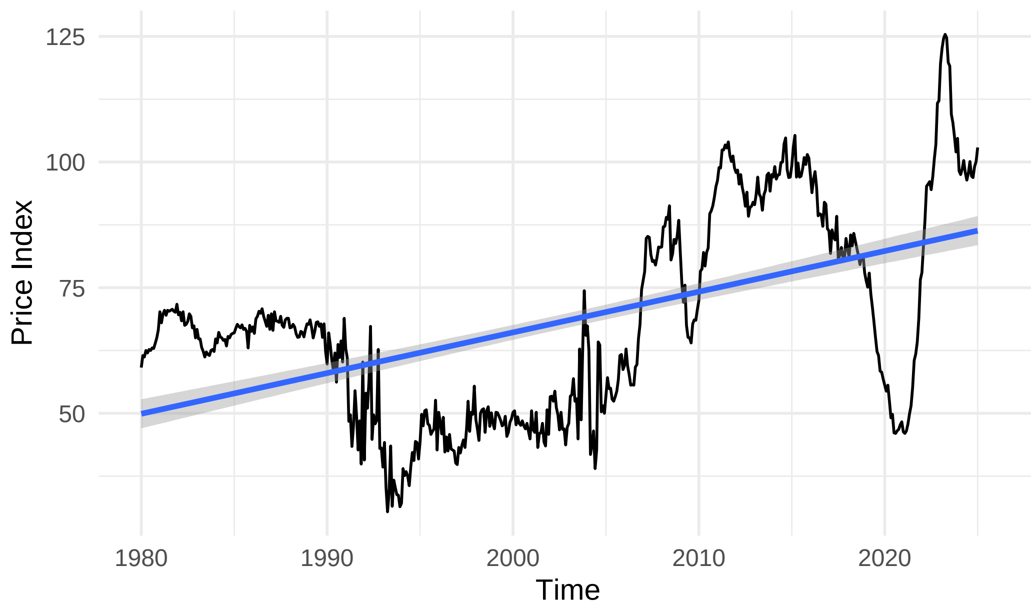


Figure 7: Monthly spruce industrial wood price index in Germany from 1980 - 2024. Data taken from https://www-genesis.destatis.de/ (Code: 61231-0002).

The seasonal trend is (among the linear trend and constant trend) another typical fixed term. Fixed term is defined as common trend that appears for the time series in general (i.e. everything that is not directly correlated with the historic observations). A *saisonal* component can e.g. be expected in the air temperature data of the German Weather Service (DWD). Temperature data with a resolution finer than a year serves as an example of seasonality. We can suppose that the termperature shows strong saisonal trends and that it in average increases (climate change). The climate change is supposed to be very slight when compared to the supposed saisonality trend. The data is available at a daily base and can be used to illustrate the concept of seasonality. The following code visualises the mean air temperature at the weather station of Göttingen.

```r
temp_goe <- read_csv2("data/month_mean_temp_goe.csv")
temp_goe <- temp_goe |> select(mean_daymean_temp) |>
  ts(start = c(min(temp_goe$year), 1), # Starting year = min year
     # Starting month = Jan
     frequency = 12) # monthly data

temp_goe |>
  autoplot() +
  guides(colour = "none") + # legend not necessary as the facets are annotated
  theme_minimal() + geom_smooth(method = "lm") + ylab("Mean Temperature (°C)")
```

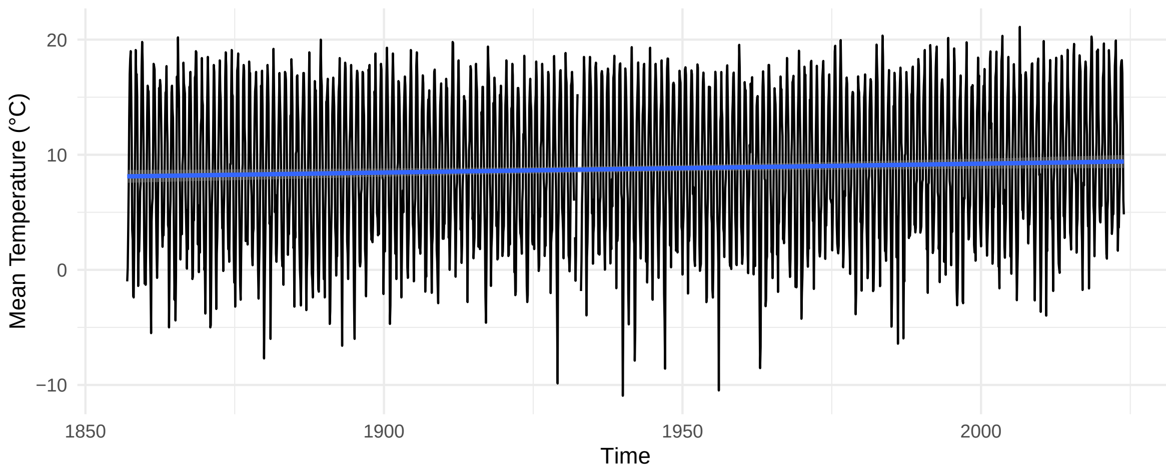


Figure 8: Monthly mean air temperature (mean of day means) at 2m height at the weather station of Göttingen from January 1857 till December 2023. Taken from (https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/monthly/kl/historical/).

We can use the native function `window` to extract a part of the time series.

```r
temp_goe |> window(start = c(2000, 1), end = c(2020, 12)) |>
  autoplot() +
  guides(colour = "none") + # legend not necessary as the facets are annotated
  theme_minimal() + geom_smooth(method = "lm") + ylab("Mean Temperature (°C)")
```

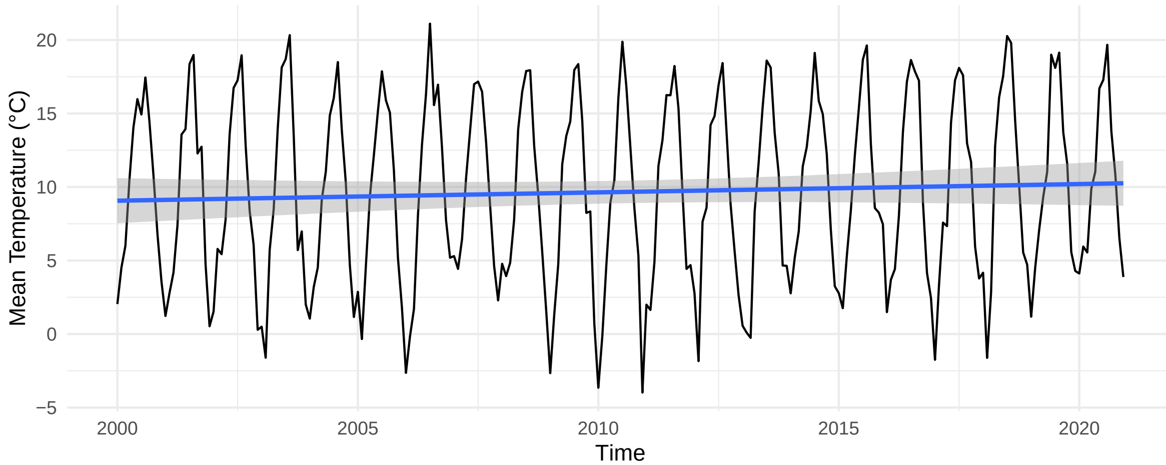


Figure 9: A window of Figure 8 from year 2000 to 2020.

Time series exercise 3

Go to https://flinga.fi/s/FCZ78B4

1. Imagine one time series. Decide which of the components (trend, seasonality, autocorrelation) apply.
2. Describe your time series briefly (heading + ~ 1-2 lines) and write the description into a purple sticky note. Arrange your sticky notes in a horizontal line.
3. Consider whether your data can be analysed using methods that are non-time-series, or if you need to make use of time series methods. Don't write down your answer - just think about it.
4. If possible, suggest a non-time-series method for analysing your data series on a blue sticky note. Stick this note somewhere, but not directly under your purple sticky note.
5. Now let's go through all the notes together in class.

- Which data series is analysable using non-time-series-methods?
- If so, which method would suit?
- What would be the additional information/ the advantage of a time series method
233 instead?

Stationarity

How is the concept of stationarity related to the three components of time series?

To understand time series data and the assumptions of the time series model, a time series can be thought of as a stochastic process with a latent data generating process (the population) and a realisation at the level of the random sample (Figure 3), just as we did for the other data types. In contrast to the populations so far, an observed time series y_t, $t = 1, ..., T$ is regarded as one realization of a finite part of a stochastic process $y_t(\omega)$ (Lütkepohl and Krätzig 2004, 10, 11). We do only describe the stochastic theory very superficially. You can find a deeper insight and references to the textbooks from which the time series theory originate in Lütkepohl and Krätzig (2004, 11). A stochastic time series process is **stationary** if all of its members are mutually independent, which in particular for time series processes means that all members are time invariant. Stationary observations are independent. Such a stationary process, also called white noise, would generate observations that fluctuate around a stable mean and have a constant variance. Such a process would meet all assumptions (iid, common variance) that we have talked about so far (Chapter 2, see also Figure 3) and would not require time series methods. The ordinary (unconditional) arithmetic mean, calculated from any realised series, would be an unbiased estimator of the population mean. The same would appear for the standard deviation. In reality, of course, we never know whether our apparently observed non-stationary time series has arisen from a stationary process, or whether it really has arisen from a non-stationary process (see also Chapter 2). Consider the following 5 simulated observations emerging from a stationary process with mean 0 and a standard deviation of 1. All of these 5 series have a mean close to 0, of course. Indeed, the mean and standard deviation would thus provide unbiased estimates for the population but for the red line, as an example, it is difficult to recognise visually that it has emerged from a stationary process. The line could also be interpreted as a increasing trend or autocorrelation. This problem occurs to any observed time series. While it sometimes seems obvious that there is an autocorrelation component (e.g. Figure 7), a seasonal component (e.g. Figure 9), or a trend component (e.g. Figure 5), in fact this is no clear advice that a series does not evolve from a stationary process. When it comes to testing for stationarity, we must remember that we are only testing the realisation, never the population, in the sense of *how likely is it that this realisation could arise from a stationary process?*

More formally[2], stationarity means that each member of a series in the population has the same expectation and expected variance (homoscedasticity).

$$
\begin{aligned}
E[y_1] &= E[y_2] = \dots = E[y_T] = \mu \\
Var[y_1] &= E[(y_1 - \mu)(y_1 - \mu)] = Var[y_2] = \dots = Var[y_T] = \gamma_0
\end{aligned}
$$

[2] the formulations are mainly taken from Lütkepohl and Krätzig (2004, 12 ff.) but aligned to the nomenclature of the course

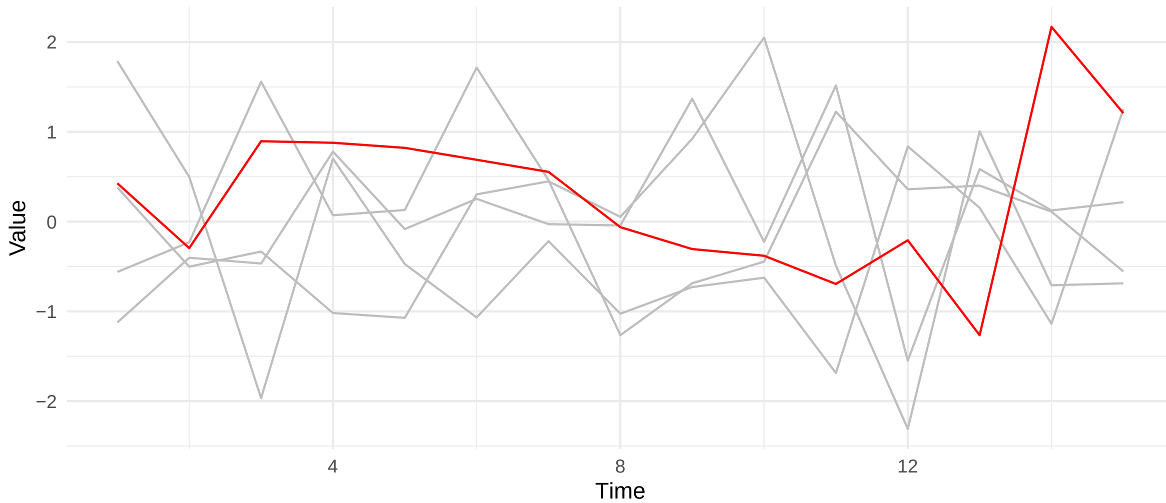


Figure 10

From which follows that the covariance between two arbitrary members y_t and y_{t+h} is a function of the lag h only. h is the difference between two time points within one series. The covariance is called the autocovariance and is denoted as γ_h.

$$Cov[y_{1+h}, y_1] = E[(y_{1+h} - \mu)(y_1 - \mu)] = Cov[y_{2+h}, y_2] = \cdots = Cov[y_T, y_{T-h}] = \gamma_h$$

For $h > 0$ and $h < P$. Under the assumption of stationarity, the ordinary descriptive statistics for iid sampled statistics are therefore appropriate to describe time series as well.

- Mean: $\hat{\mu} = \bar{y} = \frac{1}{T} \sum_{t=1}^{T} y_t$
- Variance: $\hat{\gamma_0} = \frac{1}{T-1} \sum_{t=1}^{T} (y_t - \bar{y})^2$
- Covariance, also called autocovariance: $\hat{\gamma_h} = \frac{1}{T} \sum_{t=1}^{T-h} (y_{t+h} - \bar{y})(y_t - \bar{y}), h = 1, 2, ...$

By convention, it is not common to do finite correction in time series analyses. The autocorrelation it the calculated as the relation between the covariance and variance.

- Autocorrelation (AC): $\hat{\rho_h} = \frac{\hat{\gamma_h}}{\hat{\gamma_0}}, h = 1, 2, ...$

A white noise process would lead to a stationary time series with common mean $\hat{\mu}$ for all observations and time-invariant variance $\hat{\gamma_0}$, as mentioned earlier, and also to an autocorrelation of 0. Consider for example the autocorrelation of the red time series (Figure 10). `R` comes with a native function `acf` that calculates the autocorrelation for h from 0 to `lag.max`. It can be seen that except for $h = 0$, which is of course always 1, the autocorrelation is close to 0 for all h. The set of ordered autocorrelations with increasing h, is also called autocorrelation function. The autocorrelation functions helps in identifying the time-dynamic component of a time series. The red series appears to be *stationary* indeed.

```
example_whitenoise_red_ts |> acf(lag.max = 10)
```

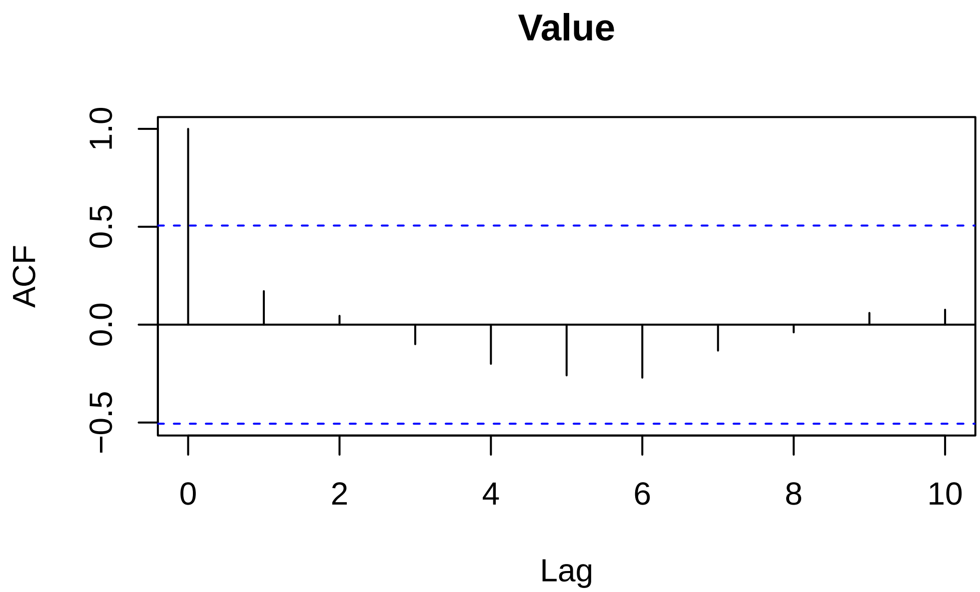


Figure 11: Autocorrelation function of a white noise time series. Note that the dashed lines cannot be interpreted as confidence intervals in the sense of significant correlation must be above s critical value even if this is sometimes proclaimed in scientific literature

Note that `urca` package (among others) provides a unit root test to perform a statistical test for stationarity. However, we will not capture this or any other testing procedure in this course.

> Time series exercise 4
>
> Join your group from Exercise 2 again. Load your workspace from Exercise 2.
>
> 1. Create an autocorrelation function (ACF) for your species with an appropriate lag order.
> 2. Considering both, the plot created in exercise 2 and the ACF. Which components do you expect to have in your series?
> 3. Save your workspace.

A white noise process with linear trend but without autocorrelation, is called *trend stationary* in time series statistics. Such a trend would be sufficiently described by means of an ordinary linear regression. Or in other words: A stationary process shows no autocorrelation after correcting for the linear trend. Followingly, the residuals of a linear regression would be stationary. The function `tslm` can be used to wrap an `lm` for `ts` objects. However, putting the `ts` object in `lm` directly would also work. You then need to define the years as the only covariate. Correcting for the linear trend of the stem wood prices (Figure 6), for example, leads to the following time series and autocorrelation functions.

```
# Calculate lm and save the residuals
detrended_stemwood_prices <- ts.union(                                    ①
  tslm(stemwood_prices[, "Oak"] ~ trend) |>                               ②
    residuals(),                                                          ③
  tslm(stemwood_prices[, "Spruce"] ~ trend) |> residuals(),
  tslm(stemwood_prices[, "Beech"] ~ trend) |> residuals())

# Keep the original names
colnames(detrended_stemwood_prices) <- colnames(stemwood_prices)

detrended_stemwood_prices |>
  autoplot(facets = TRUE, colour = TRUE) + facet_wrap(~ series) +
  ylab("Price Index") +
  guides(colour = "none") + # legend not necessary as the facets are annotated
  theme_minimal()
```

① `ts.union` is the `cbind`-pendant for `ts` objects.
② Calculate a regression model with trend = Detrending.
③ Store (only) the residuals of that model.

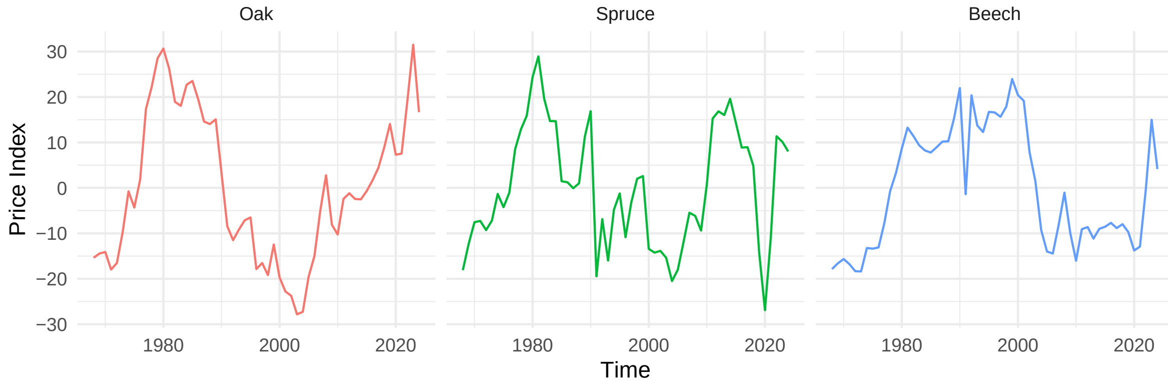


Figure 12: Detrended oak, spruce and beech stem wood price indices in Germany (the original series are shown in Figure 6).

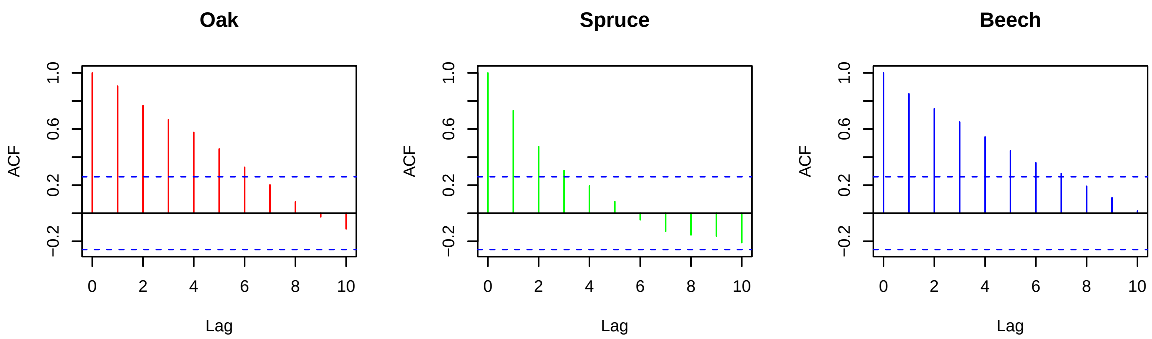
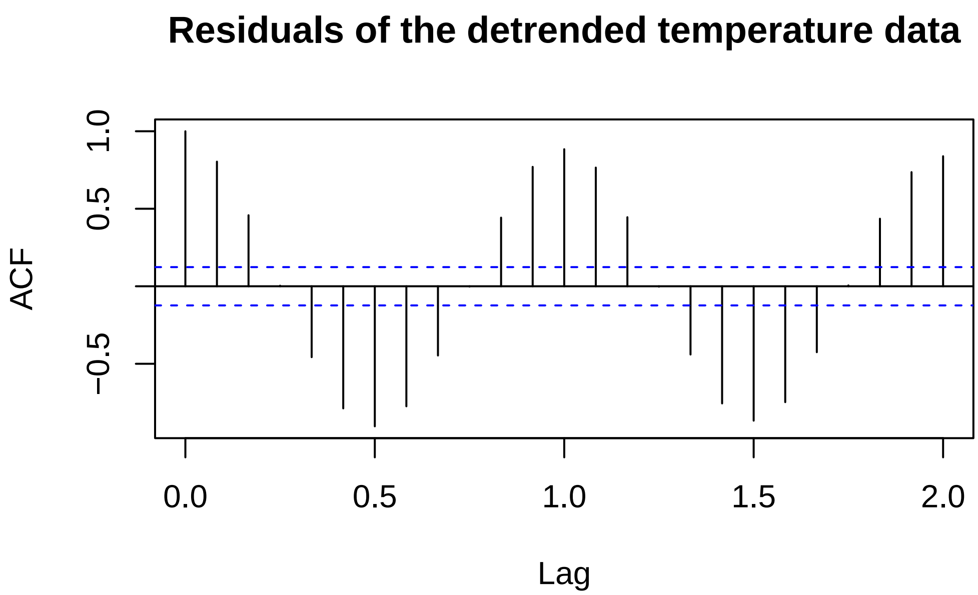


Figure 13: The respective autocorrelation functions of the detrended series from Figure 12 using the acf function.

Here we see that there is indeed still evidence of autocorrelation after detrending. All three species tend to have decreasing autocorrelations with increasing lag (h). Note that the dashed lines cannot be interpreted as confidence intervals in the sense of *significant correlation must be above s critical value* even if this is sometimes proclaimed in scientific literature. It is the autocorrelation under perfect noise. Nevertheless, the autocorrelation functions indicate that none of the time series are stationary or trend stationary. Doing the same (detrending and then calculating an autocorrelation function using `acf`) for the temperature data (Figure 9) leads to the following autocorrelation function.

```
tslm(temp_goe |> window(start = c(2000, 1), end = c(2020, 12)) ~ trend) |>
  residuals() |> acf(lag.max = 24, main =
                          "Residuals of the detrended temperature data")
```



Note that a `lag.max` of 24 means that h is set to 2 years (= 24 months). As expected, the autocorrelation is close to 1 after a full cycle (year) and highly negative correlated after the half cycle. This series is thus also neither stationary nor trend stationary. Additionally to the autocorrelation function of the timber wood prices (Figure 6), the autocorrelation function reveals a seasonal component. It can be followed that the time series is not trend stationary. Yet, it remains unclear whether this strong remaining autocorrelation after detrending is only due to the seasonal component (*the temperature in one month is autocorrelated with the same month of the previous year*) or whether the series is also autocorrelated in the sense of *the temperature in one month is autocorrelated with the temperature of the previous months*. The seasonal component can be removed in the same way as the trend component. A linear regression using only dummy variables, one dummy for each point in the cycle, 12 months in our example. Such kind of regressions thus require a very huge set of data as 13 parameters need to be estimated. In general, time series methods require a large number of data points. Especially if seasonal trends are to be estimated. The `tslm` function saves some programming effort here, as it automatically uses the frequency information of the `ts` object to create the number of dummy variables. A model containing this seasonal component and also the trend can be parameterised as follows:

```
tslm(temp_goe |>
       window(start = c(2000, 1), end = c(2020, 12)) ~ trend + season) |>
  residuals() |> acf(lag.max = 24, main =
                       "Resid. of the detrended & seasons-corrected temp.")
```

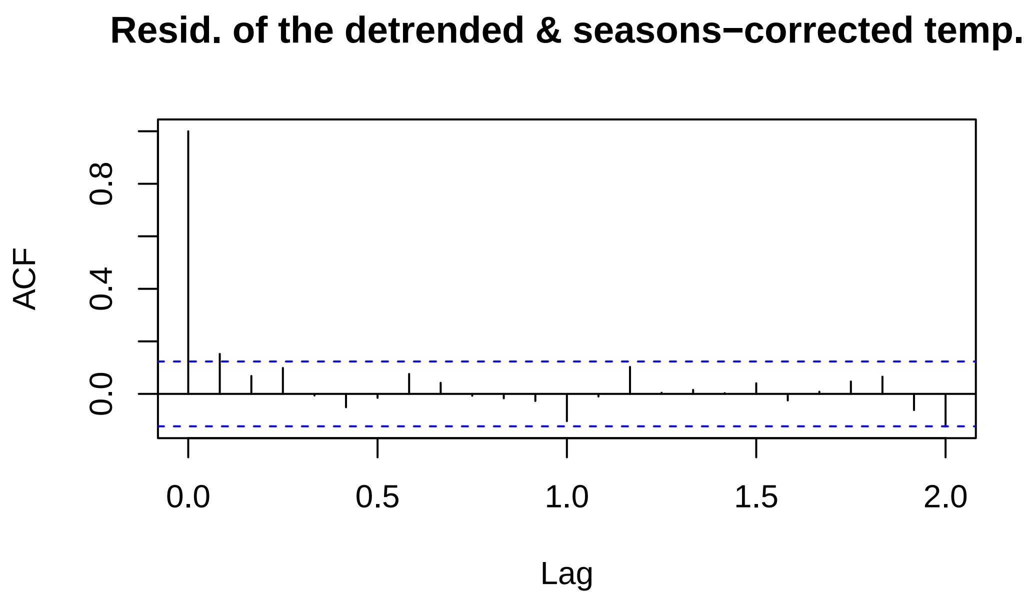


Figure 14: Autocorrelation function of the residuals of the detrended and seasons-corrected temperature data in Göttingen.

A look at the autocorrelation function of the residuals (Figure 14) reveals that the autocorrelation is now close to 0 for all h. There is evidence for stationarity of the residuals. The temperature data thus mainly consists of the season and the trend component. There is no further time dynamic in the data then the trend and the season. We can access the parameters of the trend and the season component just as we do it in `lms`. `summary` of the model gives us:

```
tslm(temp_goe ~ trend + season) |> summary()
```

```
Call:
tslm(formula = temp_goe ~ trend + season)

Residuals:
     Min       1Q   Median       3Q      Max
-11.7326  -1.0655   0.0466   1.1868   5.4121

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.878e-01  1.676e-01  -1.717 0.086152 .
```

```
trend        6.284e-04  7.489e-05   8.390  < 2e-16 ***
season2      7.927e-01  2.122e-01   3.735 0.000193 ***
season3      3.793e+00  2.122e-01  17.872  < 2e-16 ***
season4      7.847e+00  2.122e-01  36.972  < 2e-16 ***
season5      1.235e+01  2.122e-01  58.201  < 2e-16 ***
season6      1.553e+01  2.122e-01  73.191  < 2e-16 ***
season7      1.707e+01  2.126e-01  80.324  < 2e-16 ***
season8      1.649e+01  2.126e-01  77.571  < 2e-16 ***
season9      1.317e+01  2.122e-01  62.042  < 2e-16 ***
season10     8.715e+00  2.126e-01  40.999  < 2e-16 ***
season11     4.147e+00  2.126e-01  19.512  < 2e-16 ***
season12     1.147e+00  2.129e-01   5.387 8.02e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.939 on 1985 degrees of freedom
  (6 Beobachtungen als fehlend gelöscht)
Multiple R-squared:  0.9098,    Adjusted R-squared:  0.9092
F-statistic:  1668 on 12 and 1985 DF,  p-value: < 2.2e-16
```

Plotting these parameters and the residuals provides us graphical evidence of the relevance of the 3 components. In our example we see that there is a slight but significant trend component (climate change) and a strong and significant seasonal component. The remainder appears to be white noise only at first sight and by consideration of the autocorrelation function (Figure 14).

```r
tslm_example_temp_goe <- tslm(temp_goe ~ trend + season)

p0 <- temp_goe |>
  autoplot() +
  guides(colour = "none") + # legend not necessary as the facets are annotated
  theme_minimal() + ylab("Mean Temperature (°C)") + ggtitle("Observed data") +
  ylim(-10, 20)

p1 <- ggplot() + geom_abline(intercept = tslm_example_temp_goe$coefficients[1],
                             slope = tslm_example_temp_goe$coefficients[2]) +
  ylab("") +
  theme_minimal() + ylim(-10, 20) + xlim(0, 2004) +
  ggtitle("Trend component") +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) # To avoid several x-axes
```

```r
p2 <- tibble(Months = c(1 : 12), `Mean Temperature (°C)` =
               c(0, # Jan is the reference month
                 tslm_example_temp_goe$coefficients[3 : 13])) |>
  ggplot(aes(x = Months, y = `Mean Temperature (°C)`)) + geom_line() +
  ylab("") + #xlab("Month") +
  theme_minimal() + ylim(-10, 20) + xlim(1, 12) +
  ggtitle("Seasonal component") + scale_x_continuous(breaks = c(1 : 12))

p3 <- ggplot() + geom_line(aes(y = tslm_example_temp_goe$residuals,
                               x = 1 : 2004)) +
  ylab("") + xlab("") +
  theme_minimal() + ylim(-10, 20) + xlim(0, 2004) + ggtitle("Remainder") +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

patchwork::wrap_plots(p0, p1, p2, p3, ncol = 1)
```

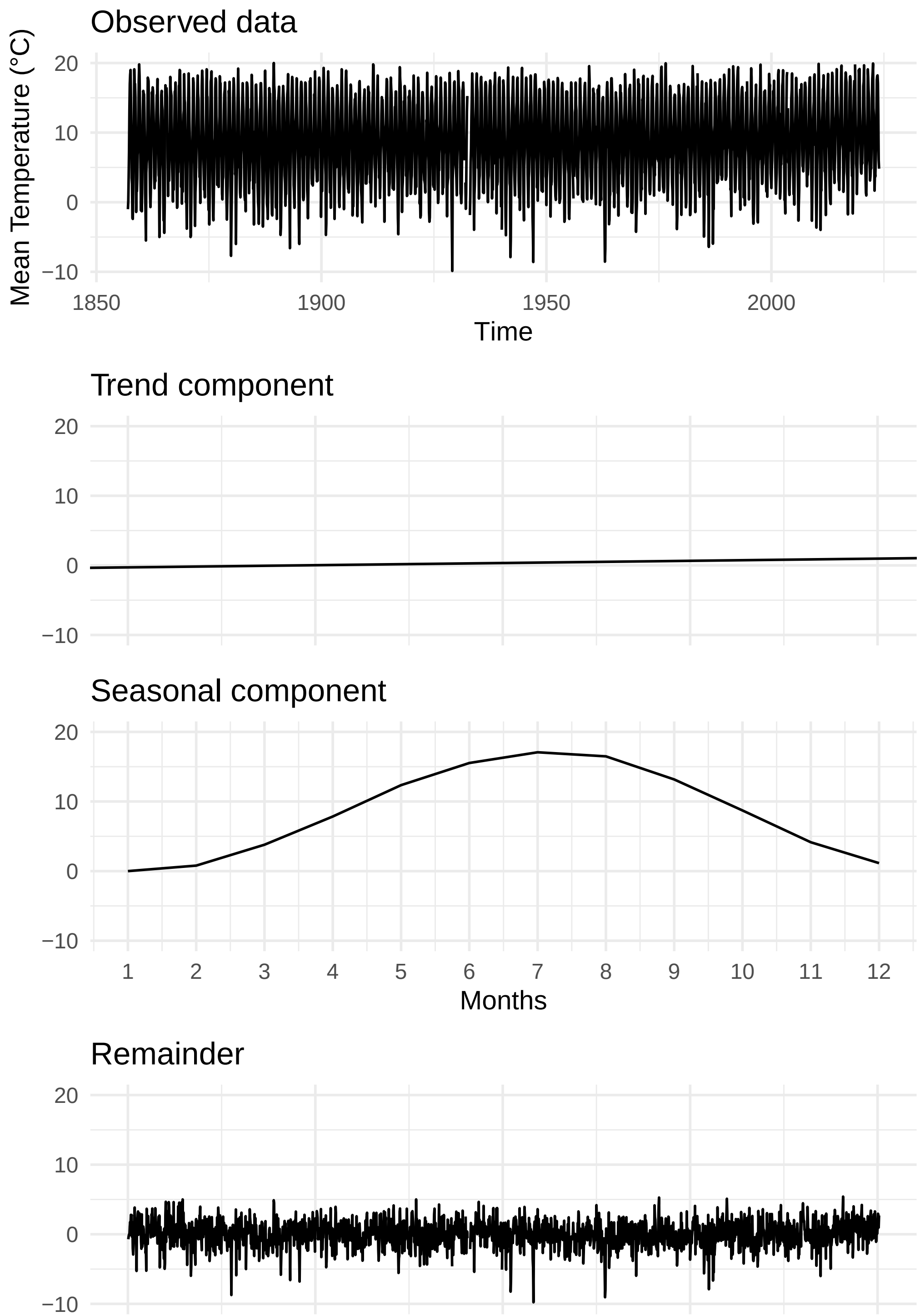


Figure 15: Raw data, trend component, season component and ramainder to visualise all components of time series data. Note that the seasonal component has an x-axis different to the other diagrams in order to better visualise the annual development.

Time series exercise 5

Join your group from Exercise 4 again. Load your workspace from Exercise 4.

1. Which of the 3 components (trend, season, autocorrelation) do you expect in your time series (see Exercise 4)?
2. Use `ts.lm` and `acf` to test your expectations.
3. Save your workspace.

Descriptive Statistics and statistical modeling

Classical Decomposition

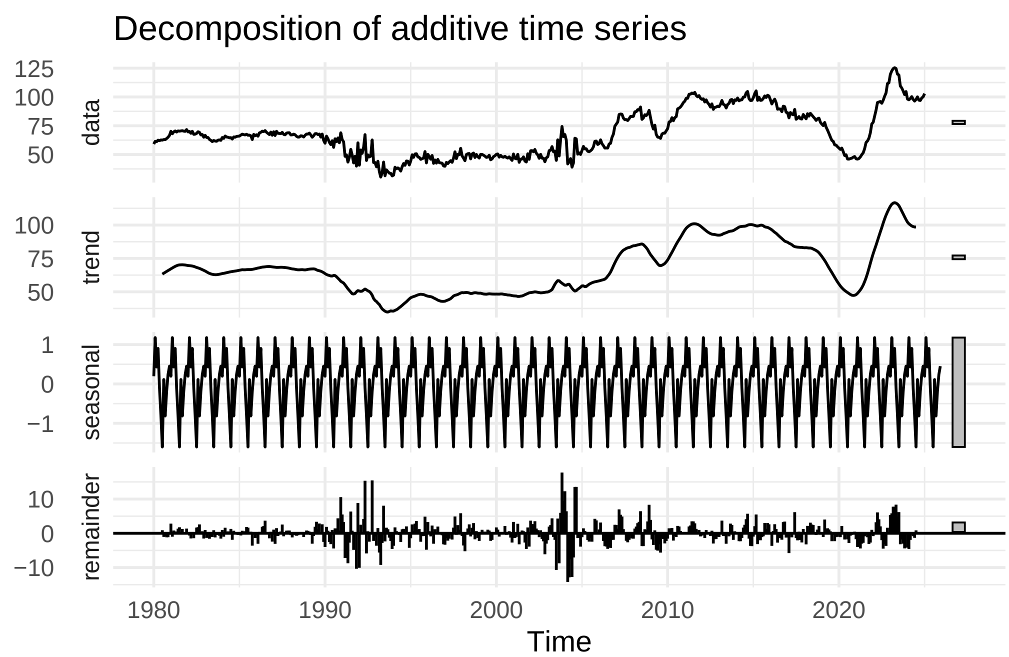
To decompose a time series into its components *trend*, *season* and *autocorrelation*, as we have done it in the last subchapter, is a commonly used technique to check whether time series statistics need to be applied to a series or whether ordinary models are sufficient. The aim is to determine whether there is a autoregressive time pattern or if the series has deterministic components (trend, season, and also constant) only. The so called classical decomposition is a set of descriptive statistics in time series statistics. In the previous subchapter, we developed a simple additive decomposition with a linear trend and a linear seasonal component. Others, such as polynomial trends or trigonometric seasonal components, are also commonly used. The native `R` functions `decompose` or `stl`, among others, provide numerous methods for decomposing a time series and for visualisation. The ordinary linear model that we parameterised above can be used if the following assumption of additivity holds:

$$y_t = m_t + s_t + u_t$$

where y_i is the observed time series, m_t is the trend component, s_t is the seasonal component, and u_t is the remainder. In general, any regression model can be used to decompose a time series into deterministic components and the possibly autocorrelated remainder (residuals). The most simple and straightforward model is a linear model with one parameter for the trend, as we have already performed. The native `R` function `decompose` used a symmetric moving average approach to estimate the trend. Advantage of the moving average approach is that autocorrelation (firstly in this course) is considered as the trend is calculated by means of last P observations. Per default, the last 6 observations are used with equal weights. Disadvantage is that we do not get a parameter for the trend component. `decompose` does not deliver any parameter information. The seasonal trend is then estimated by means of a linear model, just as we did in the last subchapter.

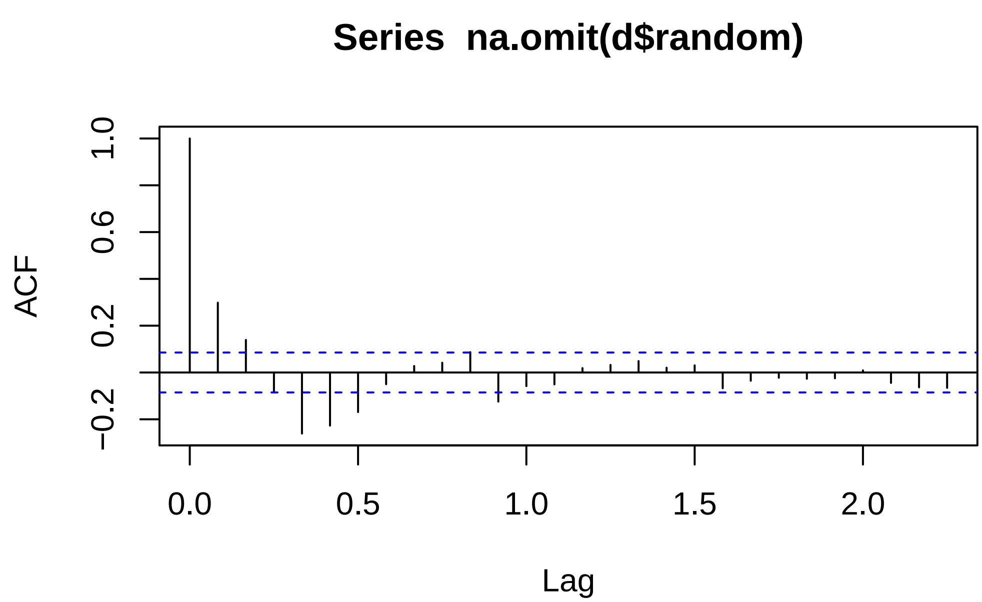
Decomposition of our industrial wood price of spruce (Figure 7) leads to the following picture:

```
decompose(ind_prices, type = "additive") |> autoplot() + theme_minimal()
```



and to the following autocorrelation function of the remainder:

```
d <- decompose(ind_prices, type = "additive")
d$random |> na.omit() |> acf()
```



Which is already pretty good in describing this relatively complex data set.

> Time series exercise 6
>
> 1. Read in `month_mean_temp_goe.csv`
> 2. Filter a window from Jan 2000 till Dec 2020
> 3. Perform a classical decomposition using `decompose` (type = "additive").
> 4. Plot the results.
> 5. Compare the results to the Figure 15.
> 6. Why is classical decomposition usually referred to as **descriptive statistics**?

Autoregressive Models (AR)

In case there is autocorrelation in the data, even after correcting for the deterministic components, ARMA models are capable to estimate those autocorrelative terms via coefficients. Autoregressive (AR) and moving average (MA) models are commonly used to estimate coefficients for autocorrelative terms in univariate time series. According to Verbeek (2004, 279), a population that follows a ARMA process can be estimated by means of ordinary or nonlinear least squares, or by maximum likelihood. The `Arima` function from the package `forecast` uses the maximum likelihood approach as a default (`R` comes with the native function `arima`, which has the same functionality as `Arima` - however, `Arima` is streamlined with further functionalities of the `forecast` package, such as visualisation and further processing). AR models of order lag order p can be expressed as

$$y_t = \delta + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \cdots + \theta_p y_{t-p} + \varepsilon_t,$$

which is a consistent estimator for a population wit autocorrelation. ε_t are white the noise remainders, δ is the intercept, and are the coefficients to be estimated. Estimation of an autoregressive model is thus no different than that of a linear regression model with a lagged dependent variable. Consider e.g. Verbeek (2004, 280) for more details on the estimation. Even though MA models are among the most relevant time series regression models as well, we will concentrate on AR models only in this course. Take a look at Verbeek (2004, 106, 281) if you want to learn more about MA data structures in comparison to AR data and about MA modelling. We thus only use the `ar` part of the `Arima` function and followingly do not have to choose the type of model but only the deterministic components and the lag order for the autocorrelation. Both can straightforwardly be done visually by the time series plot and the ACF.

Let's come back to the oak stem wood prices. We have not yet found a satisfactory model. There was an obvious linear trend towards higher prices, but also a remainder that is not a seasonal trend (Figure 6, Figure 12, Figure 13). In order to describe the autocorrelation as independent as possible from the other components, we first exclude the linear trend (detrending). However, detrending is not mandatory to use AR or MA regression. The same could have been done with the seasonal component, if there were seasonality in the data or if saisonality would be expectable.

```
stemwood_prices_tslm <- tslm(stemwood_prices[, 1] ~ trend)

stemwood_prices_detrended <- stemwood_prices_tslm |>
  residuals()
```

Now let's define the lag length p for our AR model using `VARselect` from the `vars` package. We set the maximum lag to 5 years.

```
vars::VARselect(stemwood_prices_detrended, lag.max = 10)
```

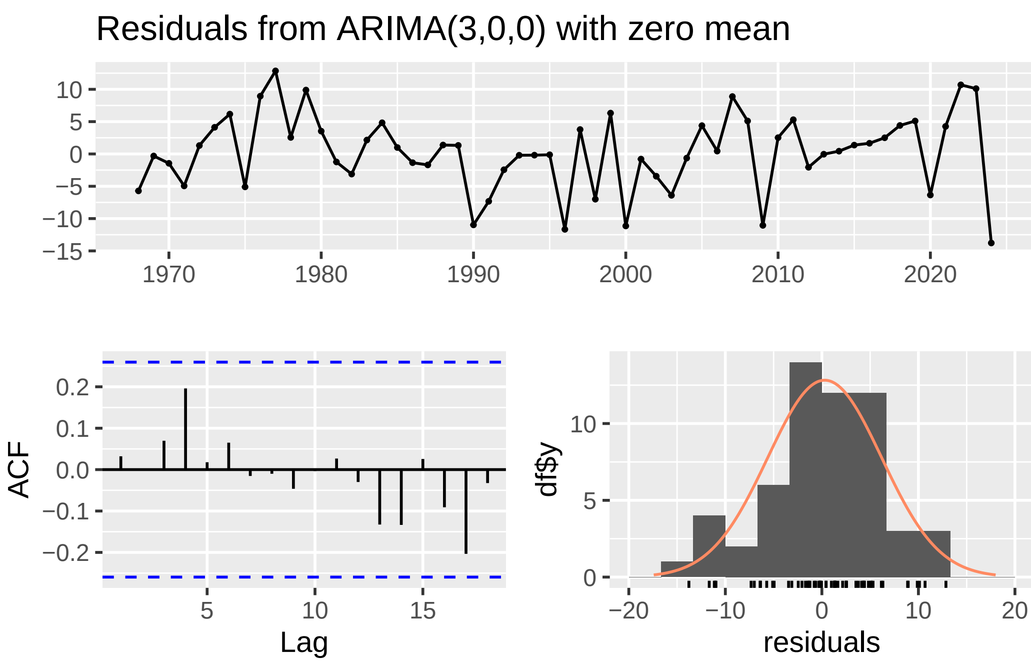
```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n) 
     3      2      1      3 

$criteria
               1         2         3         4         5         6         7
AIC(n)  3.699699  3.678604  3.677559  3.715726  3.712193  3.743193  3.732274
HQ(n)   3.729325  3.723044  3.736812  3.789792  3.801073  3.846886  3.850780
SC(n)   3.778428  3.796698  3.835018  3.912550  3.948383  4.018747  4.047193
FPE(n) 40.437202 39.597962 39.566059 41.121611 41.000906 42.326993 41.914011
               8         9        10
AIC(n)  3.774058  3.802477  3.840402
HQ(n)   3.907378  3.950610  4.003348
SC(n)   4.128342  4.196126  4.273416
FPE(n) 43.765469 45.108866 46.957429
```

Unfortunately, the indices indicate suggest a relatively large range of lag orders between 1 and 3 years. Let's additionally consider the ACF of the detrended residuals (Figure 13), which indicates that the lag length should be considered as long as possible. We can follow that there is a high an relatively long autocorrelation in the data. We thus parameterise an AR having a lag order of 3 years.

```
stemwood_prices_detrended_ar_3 <- stemwood_prices_detrended |>
  Arima(order = c(3, 0, 0), include.mean = FALSE)
```

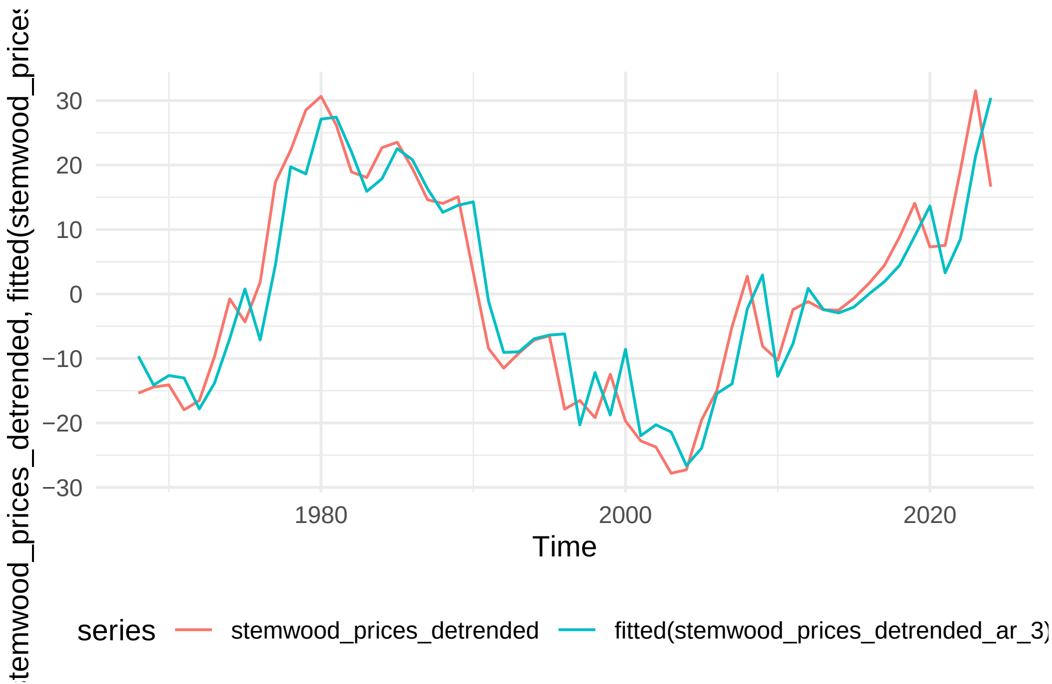
and then do first stability tests as

```r
checkresiduals(stemwood_prices_detrended_ar_3, test = FALSE)
```



The model seems to be pretty stable. `checkresiduals` provides an ordinary residual histogram (bottomleft), which here does not reveal any advice for biasedness. The remainder (top) and the autocorrelation function (bottomleft) do not contain any indication of any kind of unexplained autocorrelation. We can further inquiry the model validity by applying it on the time series and visually compare the fitted values to the observed values as

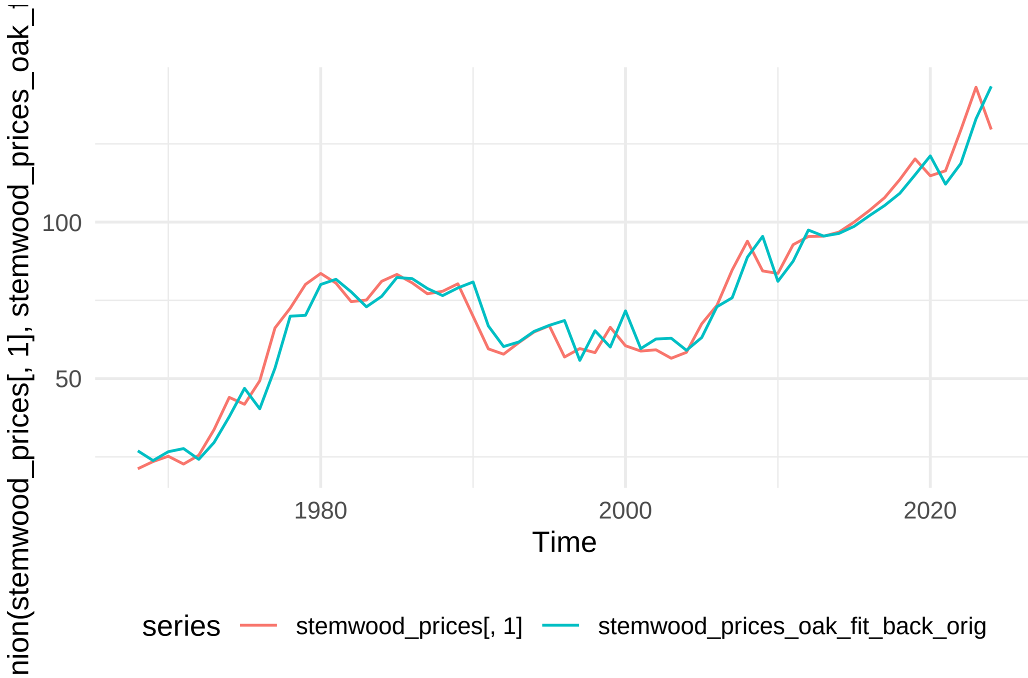
```r
ts.union(stemwood_prices_detrended,
         fitted(stemwood_prices_detrended_ar_3)) |> autoplot() +
  theme_minimal() + theme(legend.position = "bottom")
```



A lag length of 3 years thus appears to strike a good balance between accuracy and stability. Adding the linear trend from the `tslm` that we used for detrending brings the fitted value back the original scale. Note that detrending is not mandatory to perform time series regression. Alternatively, we could have fitted the AR model directly to the raw data. We decided to firstly do a linear detrending because we wanted to investigate the proposed general, long-term development towards higher wood prices separately to the short-term effect of autocorrelation.

```
stemwood_prices_oak_fit_back_orig <- fitted(stemwood_prices_detrended_ar_3) + ①
  stemwood_prices_tslm$coefficients["(Intercept)"] +                          ②
  stemwood_prices_tslm$coefficients["trend"] * c(1 : 57) |>
  ts(frequency = 1, start = c(1968))

ts.union(stemwood_prices[, 1],
         stemwood_prices_oak_fit_back_orig) |> autoplot() +
  theme_minimal() + theme(legend.position = "bottom")
```

① Forecasting the time series with regard to the autocorrelation `arima`.
② Adding the linear trend from the `tslm`.



Now that we have a valid model that captures all desired components (trend, autocorrelation) to our satisfaction, we can go ahead with this model. At first, we can interpret the model results. `summary` provides a table with the coefficients, their uncertainties, and some model quality measures.

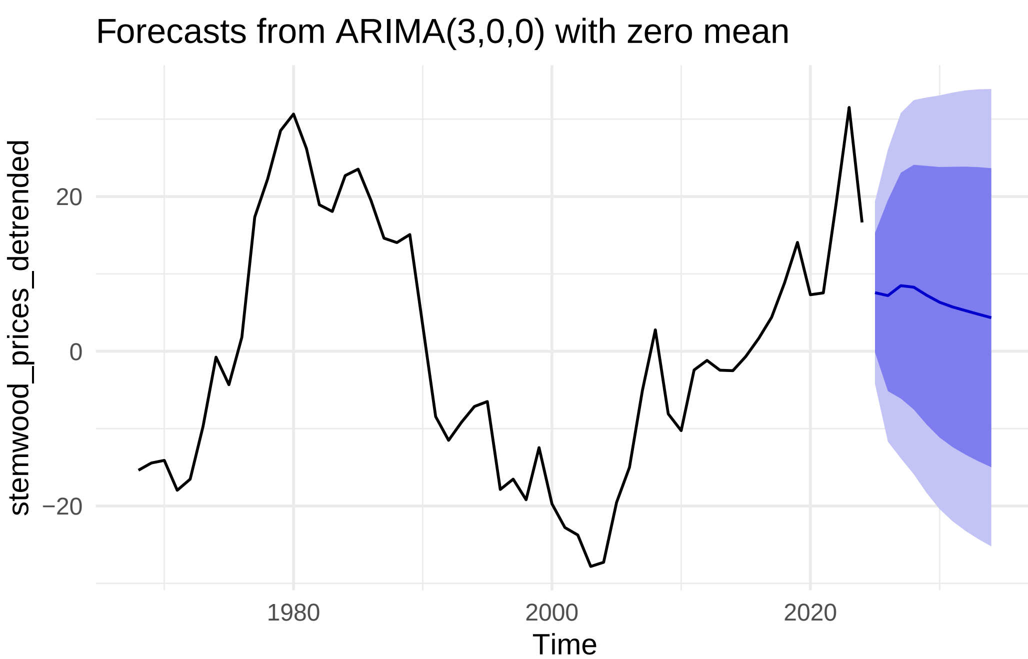
```
summary(stemwood_prices_detrended_ar_3)
```

```
Series: stemwood_prices_detrended
ARIMA(3,0,0) with zero mean

Coefficients:
         ar1      ar2     ar3
      1.2504  -0.5550  0.2212
s.e.  0.1376   0.2232  0.1439

sigma^2 = 36.23:  log likelihood = -182.73
AIC=373.47   AICc=374.24   BIC=381.64

Training set error measures:
                    ME    RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set 0.2925723 5.85842 4.523326 -1.122355 73.94896 0.9212414 0.03216206
```

The coefficients state that the last year's influence (`ar1`) is highest, followed by the negative influence of the year $t-2$. The influence of year $t-3$ is slight. The alternating signs of the parameters reflect the relatively erratic development of the timber price, but in combination, they are sufficient to fully describe the fluctuations of the market. At the same time, they have a stabilizing effect in the combination. Very high price spikes are thus equalised by these parameters relatively quickly. The low standard errors of `ar1` and `ar2` hint to a high significance of these parameters. The significance of `ar3` is on the borderline, which gives further advice that lag orders higher than 3 years might not be promising.

Common use cases include, for example, predicting the future development of the time series (forecasting), identifying windows within the time series with different development or volatility, or quantifying the robustness of the time series against disturbances. The last point is an aspect of counterfactual statistics, which is currently widely used in the sciences. Regressions and, in particular, time series regressions are suitable for analysing the influence of counterfactual scenarios, for example to quantify resistance and resilience. Fuchs et al. (2022), for example, analyse hypothetical calamity events. Dalheimer, Herwartz, and Lange (2021) apply counterfactual prediction (prediction under different future scenarios). In this course, we cover forecasting only. The forecast of the autocorrelation is straightforwardly done by:

```
forecast(stemwood_prices_detrended_ar_3, h = 10) |>
  autoplot() + theme_minimal()
```

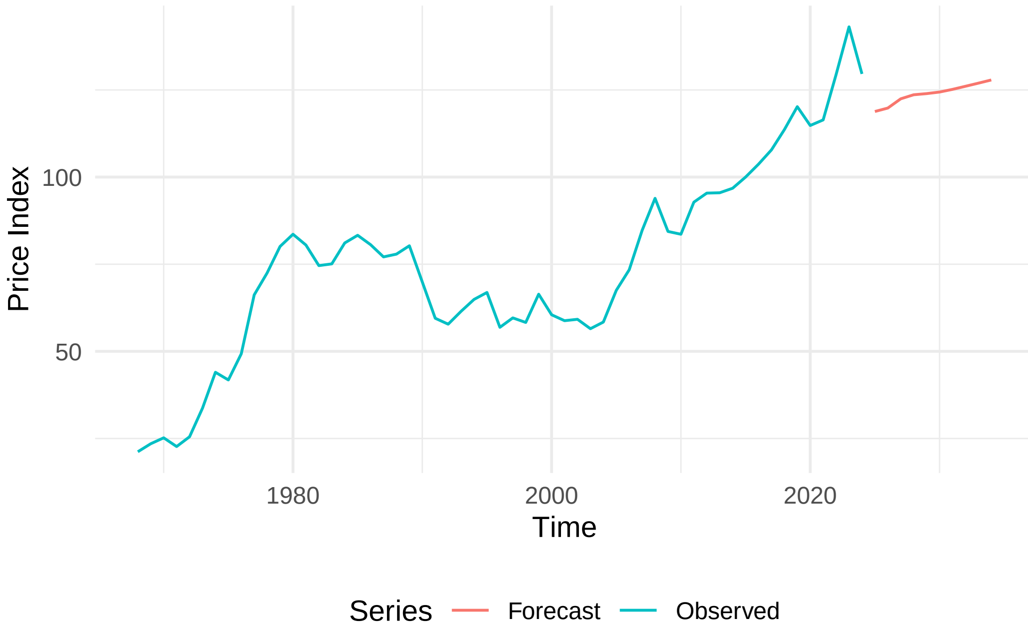


The forecast displays the afore mentioned stabilising effect. The very high price of 2023 in combination with the already declining signal of 2024 is leveled down in the first year of the forecast and then remains more or less stable.

Adding the linear trend to the forecast combines the findings of the two models. Adding the `thsm` result is quite challenging in terms of programming. One solution could be:

```
p1 <- tibble(`Price Index` =
               stemwood_prices[, 1] |> as.numeric(),
             Series = "Observed",
             Time = c(1968 : 2024))


p2 <- tibble(`Price Index` =
               forecast(stemwood_prices_detrended_ar_3, h = 10)$mean +      ①
               (coef(stemwood_prices_detrended_ar_3)["ar1"] *               ②
                  stemwood_prices[nrow(stemwood_prices), 1] +
                  coef(stemwood_prices_detrended_ar_3)["ar2"] *
                  stemwood_prices[nrow(stemwood_prices) - 1, 1] +
                  coef(stemwood_prices_detrended_ar_3)["ar3"] *
                  stemwood_prices[nrow(stemwood_prices) - 2, 1]) +
               stemwood_prices_tslm$coefficients["trend"] * c(0 : 9) |>     ③
               as.numeric(),
             Series = "Forecast",
             Time = c(2025 : 2034))

bind_rows(p1, p2) |> ggplot(aes(y = `Price Index`,
                                x = Time,
                                color = Series)) +
  geom_line() + theme_minimal() + theme(legend.position = "bottom")
```

① Forecast (point forecast only) of the autocorrelation.
② The intercept of the linear trend is the first out-of-sample forecast of the raw time series.
③ The linear trend.



To wrap up all results of the oak timber price: Our models reveal some general mechanisms of the historic oak stemwood marked, which might also allow some outlook on future developments. We hypothised a common trend towards higher prices, which seems to be revealed by the `tslm` (Figure 6). However, the remaining time series after correcting this linear trend indicates that further aspects are additionally influencing the oak stemwood price (Figure 12, Figure 13). Rather, the timber price also seems to be dependent on the prices of previous years. The AR(3) model suggests that the log price can be explained by the log prices of the last 3 years. In the long term, the timber price develops linearly. Short-term extreme prices tend to return to the value of the linear trend. This return to the linear trend occurs even faster if the price fluctuations are extreme.

Time series exercise 7

Join your group from Exercises 2, 4, and 5 again and load your last workspace.

1. Based on your analysis of the models so far, which of the following components (trend, seasonality, or autoregression) do you expect to be present in your series?
2. Create an AR model that accounts for the expected components. Find an appropriate lag order.
3. Check the model's validity.
4. Interpret the results of the model.

Regression with time dynamics - Temporal regression

> 🔥 Time series exercise 7
>
> 1. Let's take a look at the Forest Health dataset once again. You examined the autocorrelation of defoliation for your species in great detail. Now, please summarise your findings using all your plots and models and explain them to your colleagues. The following guiding questions may help you to conceptualise your explanations:
>
> - Regarding causality, what did we expect? Is autocorrelation biologically reasonable?
> - Do we find advice for this prospected autocorrelation in the raw data?
> - Can we find evidence for autocorrelation using the models applied?
> - How can the autocorrelation be interpreted?
>
> 2. How can we proceed in analysing the forest health? Which variables do you expect (causality) to influence the crown defoliation in addition to the autocorrelation?

An important insight from the entire section on time series and Chapter 9 (mixed models) is that the correlation within the data must be taken into account when creating a regression model in order to obtain unbiased estimates. This does not necessarily refer to the correlation actually observed in the data, but the expected correlation can also be included in the model. You should ask yourself: What is causally expected? In the National Forest Health dataset, for example, we can assume that defoliation in previous year(s) influences defoliation today. We need to take this into account if we want to analyse the effects of other variables. Temporal correlation (autocorrelation) can be taken into account in regression models like any other type of correlation. In this regard, the integration of an AR process provides another way to account for correlation in estimation, in addition to the mixed groups or repeated measures discussed in Chapter 9.

```
# Read the data
crown_defoliation <- read_rds("data/trees.rds")                    (1)

crown_defoliation <- crown_defoliation |>
  select(year, sp, mean_loss, bio1 : bio19) |>
```

```
  filter(sp == "spruce") |>                                                    ②
  group_by(year) |> summarise(across(c(mean_loss, bio1 : bio19), mean))       ③
```

① Data import
② Spruce only
③ Simplification of the data set to annually means of all variables

`gls` (generalised least squares) from the `nlme` package enables to consider different types of correlation when fitting linear models including autocorrelation of the response variable. We won't go into the detail of the estimator. An intercept only model of the crown defoliation using `lm`

```
lm(mean_loss ~ 1, data = crown_defoliation)
```

```
Call:
lm(formula = mean_loss ~ 1, data = crown_defoliation)

Coefficients:
(Intercept)
      22.16
```

provides a (prospectively biased) mean estimate of the crown defoliation of 22.2. The following code shows how to take a 2 years lagged autoregressive process of the response into account.

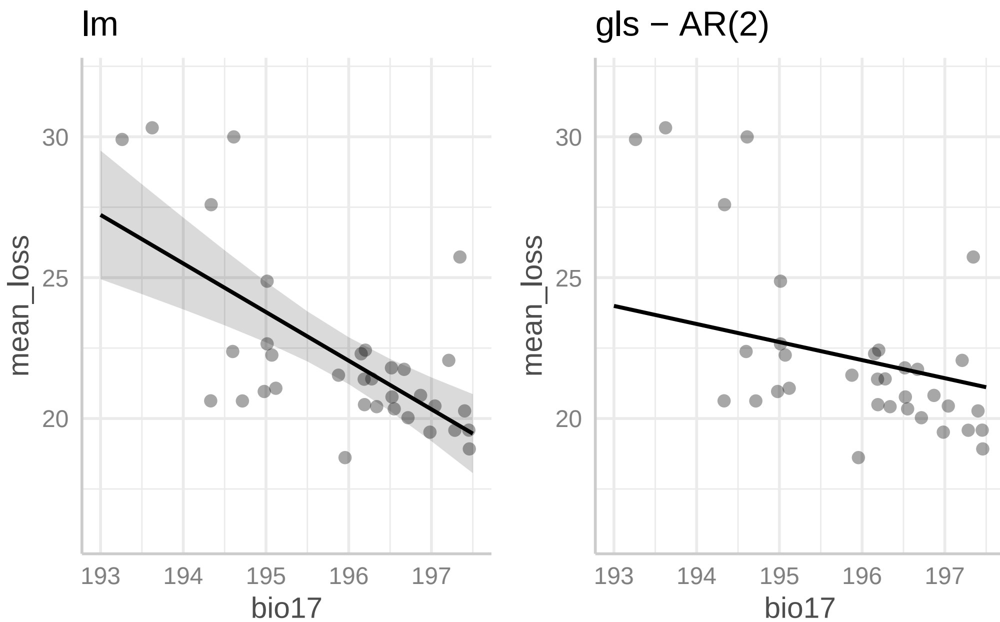
```
library(nlme)
mean_loss_intercept_model_ar2 <-  gls(mean_loss ~ 1,
                                       correlation = corARMA(p = 2),
                                       data = crown_defoliation)
summary(mean_loss_intercept_model_ar2)
```

```
Generalized least squares fit by REML
  Model: mean_loss ~ 1
  Data: crown_defoliation
       AIC      BIC    logLik
  132.3115 138.2975 -62.15576

Correlation Structure: ARMA(2,0)
 Formula: ~1
 Parameter estimate(s):
      Phi1       Phi2
 1.1486715 -0.2735285
```

```
Coefficients:
                 Value Std.Error  t-value p-value
(Intercept) 22.0742  1.887472 11.69511       0

Standardized residuals:
        Min          Q1         Med          Q3         Max
-0.92330036 -0.43942343 -0.22375817  0.07599469  2.19952767

Residual standard error: 3.748037
Degrees of freedom: 34 total; 33 residual
```

We can mainly read two aspects from the summary. First is that the coefficients `Phi` estimated to correct for the autocorrelation are pretty close to the AR coefficients of the `Arima` estimation of the group responsible for spruce. The second finding is, that the mean estimates of the crown defoliation do not differ across the 2 models. In this case, there is thus no evidence for a bias when analysing the response only. However, this might change if we consider further variables. The variable `bio17` (https://www.worldclim.org/data/bioclim.html) contains the precipitation of the warmest quarter of the year.

```r
mean_loss_bio17_lm <- lm(mean_loss ~ bio17, data = crown_defoliation)
mean_loss_bio17_ar2 <- gls(mean_loss ~ bio17, correlation = corARMA(p = 2),
                           data = crown_defoliation)


patchwork::wrap_plots(
  ggeffects::ggemmeans(mean_loss_bio17_lm, terms = "bio17") |>
    plot(show_data = TRUE) + ylim(c(16, 32)) + ggtitle("lm"),
  ggeffects::ggemmeans(mean_loss_bio17_ar2, terms = "bio17") |>
    plot(show_data = TRUE)+ ylim(c(16, 32)) + ggtitle("gls - AR(2)")
  , nrow = 1)
```



The pictures reveal the prospective bias. The `lm` is highly leveraged by the extreme mean loss values observed at a very low level of `Bio17`. The Cook's distance reinforces this statement. However, these points are expected to be highly affected by autocorrelation as well, which is not accounted for in the `lm`.

```r
broom::augment(mean_loss_bio17_lm) |> arrange(desc(mean_loss)) |> head(10)
```

```
# A tibble: 10 x 8
   mean_loss bio17 .fitted .resid   .hat .sigma  .cooksd .std.resid
       <dbl> <dbl>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>      <dbl>
 1      30.3  194.    26.2  4.17  0.149    2.28 0.315         1.90
 2      30.0  195.    24.4  5.54  0.0687   2.18 0.215         2.42
 3      29.9  193.    26.8  3.13  0.189    2.33 0.249         1.46
 4      27.6  194.    24.9  2.67  0.0865   2.36 0.0653        1.17
 5      25.7  197.    19.7  6.00  0.0736   2.14 0.273         2.62
 6      24.9  195.    23.8  1.12  0.0485   2.41 0.00593       0.483
 7      22.6  195.    23.8 -1.10  0.0485   2.41 0.00577      -0.476
 8      22.4  196.    21.7  0.721 0.0310   2.41 0.00152       0.308
 9      22.4  195.    24.5 -2.09  0.0695   2.38 0.0310       -0.912
10      22.3  196.    21.8  0.506 0.0304   2.41 0.000732      0.216
```

References

Dalheimer, Bernhard, Helmut Herwartz, and Alexander Lange. 2021. "The Threat of Oil Market Turmoils to Food Price Stability in Sub-Saharan Africa." *Energy Economics* 93: 105029.

Fieberg, John R, Kelsey Vitense, and Douglas H Johnson. 2020. "Resampling-Based Methods for Biologists." *PeerJ* 8: e9089.

Fuchs, Jasper M, Hilmar v Bodelschwingh, Alexander Lange, Carola Paul, and Kai Husmann. 2022. "Quantifying the Consequences of Disturbances on Wood Revenues with Impulse Response Functions." *Forest Policy and Economics* 140: 102738.

Hamilton, James D. 2020. *Time Series Analysis*. Princeton university press.

Lemoine, Nathan P. 2021. "Unifying Ecosystem Responses to Disturbance into a Single Statistical Framework." *Oikos* 130 (3): 408–21.

Lütkepohl, Helmut, and Markus Krätzig. 2004. *Applied Time Series Econometrics*. Cambridge university press.

Verbeek, Marno. 2004. "A Guide to Modern Econometrics. Erasmus University Rotterdam." *John Wiley & Sons Ltd., Hoboken, in: Organizational Behavior and Human Decision Processes* 67: 326–44.