Seasonal Adjustment in R

Table of contents

Part I

Welcome

This is the website for the work-in-progress edition of *Seasonal Adjustment in R*, an online Book by James Livsey and Christoph Sax.

About the book

This book will teach you how to do seasonal adjustment in R, using X13-ARIMA-SEATS.

Specifically, the audience will be both R users who want to learn about seasonal adjustment as well as seasonal adjustment practitioners, who are interested in using R. The book will be tailored to the practical applications of seasonal adjustment within R. It presents background material and references for the theoretically minded reader. The main focus, however, is on concrete problems and examples.

We will showcase methods through detailed examples with associated code. This presentation allows the academic level to be quite broad; understood by undergraduates all the way through advanced Ph.D. students.

Key features of the book

- Each chapter include a concrete practical problem and shows how X-13 can be used to address it
- Teach-by-example format
- Continuous connection of X-13ARIMA-SEATS input with R input and vice-versa
- Fundamental theoretical material is referenced throughout (mainly as an option)
- For each example given the book will give answers, code and provide data

1 Introduction

> ❗ Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

This book will teach you how to do seasonal adjustment in R, using X13-ARIMA-SEATS. The audience will be both R users who want to learn about seasonal adjustment as well as seasonal adjustment practitioners, who are interested in using R. The book will be tailored to the practical applications of seasonal adjustment within R.

Seasonal Adjustment

Many time series exhibit a regular seasonal pattern over the year. US unemployment, for example, is usually higher from January to March, and again in June and July. Similarly, retail sales tend to peak with the Christmas season. This seasonal behavior is regualar and predictable. The goal of seasonal adjustment is to estimate and remove the seasonal component from a time series.

Why do we want to do this? Seasonal data is usulally hard to interpret. For example, if we want to learn from the US unemployment rate if the economy is moving out of a recession during certain months, we want the labor market data be free from seasonal effects.

X13-ARIMA-SEATS

Fundamentally, seasonal adjustment decomposes a time series into a **trend**, a **seasonal** and an **irregular** component and removes the seasonal component from the data. There are many ways to perform this decomposition. This book focuses on a particular one, X13-ARIMA-SEATS (X13, for short), the seasonal adjustment software developed by the United States Census Bureau. X13 offers an elaborate toolkit to perform the most advanced seasonal adjustment.

R

This book will teach you how to use X13 in R, through the *seasonal* package, which offers access to all features of X13, with a ususally much simpler syntax. The required X-13 binaries are provided by the *x13binary* package, and automatically included in *seasonal*. The next chapter provides a minimal example to get you started in less than five minutes.

Target audience

We write this book for two primary audiences: The first focus is on current practitioners of seasonal adjustment who are interested in learning how to implement in R. This audience includes researchers from statistical agencies who want to include the scripting language features of R to evaluate properties of their seasonal adjustments.

The second focus is on current R users who want to learn seasonal adjustment. We are able to leverage the readers knowledge of R to make learning seasonal adjustment easier. We will feature interesting applications outside of official statistics, such as the seasonal adjustment of business data.

The book tries to be as practical as possilbe. It usually starts with a practical problems, and shows how to solve it in a cookbook style. Formal derviations are usually avoided. Each chapeter ends with a case study that discusses a real-live example of the topic.

History of X13

In official statistics, seasonal adjustment has a long tradition. The original X-11 software was developed by the US Census Bureau in the 1960s, and later improved by Statistics Canada (Dagum 1980). Subsequent software packages by the US Census Bureau were called X-12-ARIMA (Findley et al. 1998) and X-13ARIMA-SEATS (or X-13, for short) (Monsell 2007). Today, X-11 is still used as a name for filter-based seasonal adjustment methods within X-13. Meanwhile, TRAMO-SEATS, developed by the Bank of Spain (Caporello, Maravall, and Sánchez 2001), offers an alternative model-based approach to seasonal adjustment.

In its most recent version, X-13 offers these two seasonal adjustment methods in a single command-line tool, written in Fortran. The National Bank of Belgium has created an alternative Java-based implementation called JDemetra+ (National Bank of Belgium, Deutsche Bundesbank, Eurostat 2017) which is also widely deployed by statistical agencies. One of either the TRAMO-SEATS or X-11 method of seasonal adjustment is used by almost all (government) statistical offices throughout the world.

Acknowledgements

We are indebted to the the United States Census Bureau, for X-13ARIMA-SEATS and support for research around the software. Help and support by Brian Monsell is particularily acknowledged.

seasonal was originally developed for use at the Swiss State Secretariat of Economic Affairs. It has been greatly improved over time thanks to suggestions and support from Matthias Bannert, Freya Beamish, Vidur Dhanda, Alain Galli, Ronald Indergand, Preetha Kalambaden, Stefan Leist, James Livsey, Pinaki Mukherjee, Bruno Parnisari and many others. The related work on the x13binary package facilitated (automated) deployment thanks to the R package system, CRAN as well as GitHub for the x13prebuilt repository.

References

2 Getting started

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

2.1 Installation

If you use R installing X-13ARIMA-SEATS from CRAN is as easy as installing any other R package (Sax and Eddelbuettel 2018):

```
install.packages("seasonal")
```

2.2 A minimal example

Once the package is installed it can be loaded in the usual way:

```
library(seasonal)
```

The `seas()` function provides the core functionality of the package. By default, `seas` calls the automatic procedures of X-13 to perform a seasonal adjustment that works well in most circumstances:

```
seas(AirPassengers)
#>
#> Call:
#> seas(x = AirPassengers)
#>
#> Coefficients:
#>           Weekday          Easter[1]          AO1951.May  MA-Nonseasonal-01
```

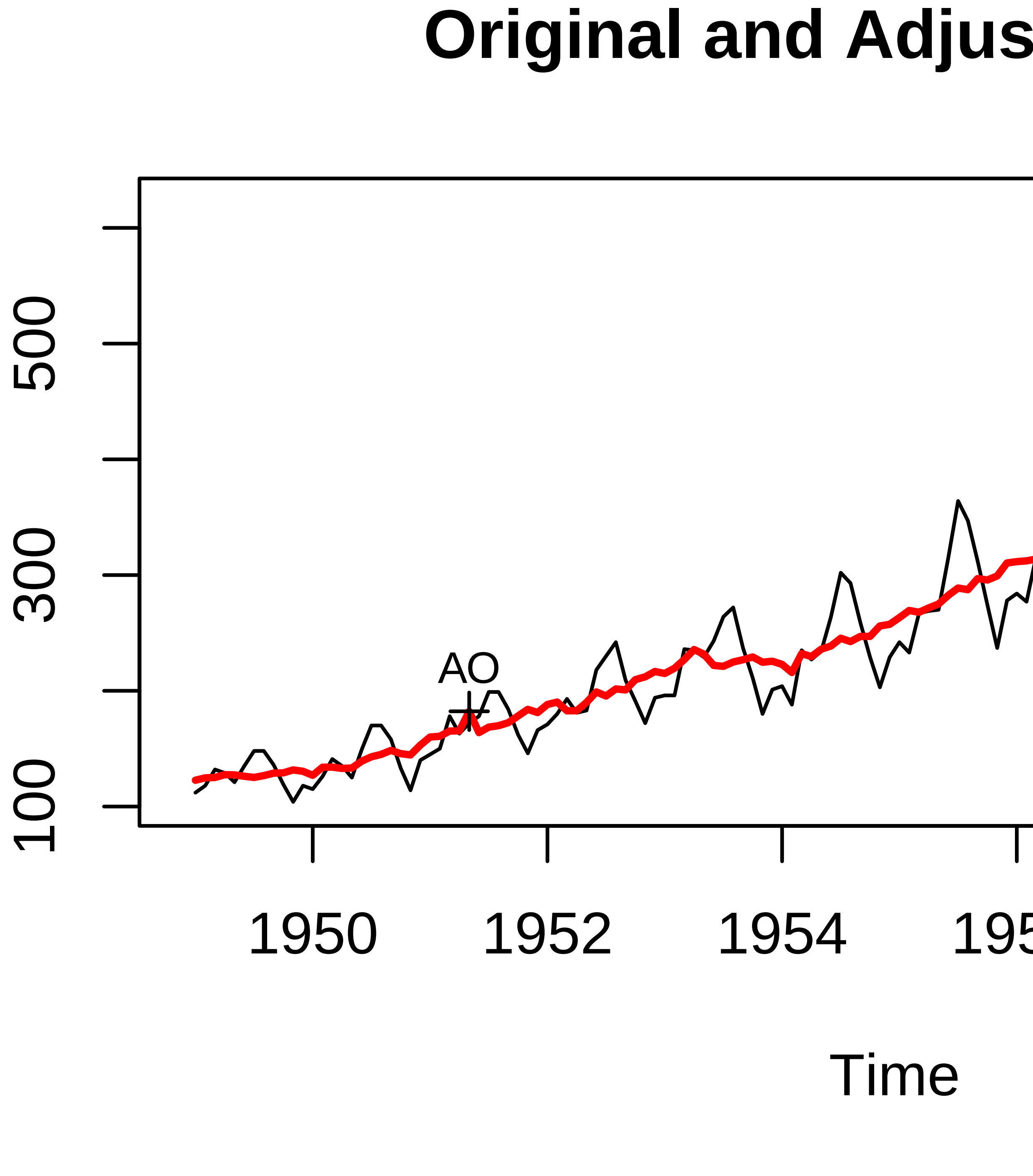
```
#>         -0.00295            0.01777            0.10016            0.11562
#>    MA-Seasonal-12
#>          0.49736
```

The first argument of seas is a time series of class `ts`. `ts` objects are frequently used in base R and are useful to store monthly, quarterly or annual data. The `AirPassengers` example series is included in base R, and shows monthly totals of international airline passengers from 1949 to 1960. `seas()` returns a `seas` object that contains the necessary information on the adjustment performed on this time series, we can assign it to a variable:

```
m <- seas(AirPassengers)
```

There are several functions and methods for `"seas"` objects. The final function returns the adjusted series. The plot method shows a plot with the unadjusted and the adjusted series.

```
plot(m)
```


Original and Adjus
AO
100
300
500
1950
1952
1954
195
Time

As you can see, the adjusted series is much less volatile than the original one. This is because the seasonal component was removed from the original series. But the adjusted series is not completely smooth, too. This is because it still contains the irregular component.

This is an important point about seasonal adjustment: It only removed regular, predictable movements, not irregular ones. In the adjusted series, we can see that there was a decrease in airline passengers both in 1953 an between 1957 and 1958. These decreases were impossible to discover in the original series.

The summary method displays an overview of the model, very similar to the one produced by R's lm function:

```
summary(m)
#>
#> Call:
#> seas(x = AirPassengers)
#>
#> Coefficients:
#>                     Estimate Std. Error z value Pr(>|z|)
#> Weekday           -0.0029497  0.0005232  -5.638 1.72e-08 ***
#> Easter[1]          0.0177674  0.0071580   2.482   0.0131 *
#> AO1951.May         0.1001558  0.0204387   4.900 9.57e-07 ***
#> MA-Nonseasonal-01  0.1156204  0.0858588   1.347   0.1781
#> MA-Seasonal-12     0.4973600  0.0774677   6.420 1.36e-10 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> SEATS adj.  ARIMA: (0 1 1)(0 1 1)  Obs.: 144  Transform: log
#> AICc: 947.3, BIC: 963.9  QS (no seasonality in final):    0
#> Box-Ljung (no autocorr.): 26.65   Shapiro (normality): 0.9908
```

The summary gives a short overview of adjustment model, and provides a few diagnostics. This book will help you do understand it more detail. The following section discusses some of the elements and relates them to the chapters in this book.

2.3 Where to go from here

`seas(AirPassengers)` produces a good seasonal adjustment of the airline passengers time series. If you are very new to seasonal adjustment, the automated routines of X13 and seasonal produce an adjustment that works well in most circumstances.

The command `seas(AirPassengers)` has invoked a large number of *specs* of X13. *spec* is X13 slang for a module within the software. X13 is build on top of 20 specs that are perform various subtasks of seasonal adjustment. Some specs are required most of the time (e.g., `regression`) while others are optional (e.g., `seats`) or purely technical (e.g., `spans` simply shortens the time series in use). Chapter 4 discusses the available specs in more detail.

This book teaches you how to use and fine-tune the individual specs, and deal with concrete data problems.

2.3.1 Fundamentals

Specifically, the command `seas(unemp)` has invoked the following fundamental specs – they are involved in most adjustments, and are covered in the **first part** of the book:

Transform A decision on initial transformation was made. The automated procedures concluded that a log-transformation was made and an multiplicative, rather than a additive seasonal adjustment model is estimated. Chapter 5 discusses the choices. Since `transform` is a relatively simple spec, it is a good starting point to get familiar with the spec idea.

Regression An automated model search concluded that `AirPassengers` is best modeled by an (0 1 1)(0 1 1) ARIMA model. Chapter 6 explains what that means an how such a model structure is determined and estimated.

SEATS / X11 Seasonal decomposition is performed by SEATS. SEATS is one of the two available options for decomposing a series, and discussed in more detail in Chapter 7. The alternative, X11, is discussed in Chapter 8.

2.3.2 Data issues

The command `seas(unemp)` has also dealt with a various of data issues, which are covered in the **second part** of the book:

Holiday Significant Easter effects have been found in `AirPassengers` and were removed from the adjusted series as well. Moving holidays like Easter or Chinese New Year are an important topic in seasonal adjustment, since they may have a large impact on the behavior of many time series. For `AirPassengers`, the number of passengers is higher in months with Easter. Moving holiday effects will be discussed in Chapter 9.

Weekday Not every month has the same number of weekdays. Since many activities (such as air travelling) are different bitween weekends and weekdays, this constitutes another predictable component. In `AirPassengers`, there are less passengers during a weekday than during a weekend, and the automated procedures decided to remove the effect. These effects are discussed in Chapter 10.

Outliers Certain data points may me well out of the ordinary. These *outliers* are a problem for the modelling and adjustment process. An automated procedure scanned the series for outliers and found an additive outlier on May 1951. This outlier is shown in the plot above, too. Outliers are discussed in Chapter 11.

Seasonal Breaks The seasonal pattern in `AirPassengers` looks relatively stable. Some time series, however, show abrupt changes in the seasonal pattern. Chapter 12 discusses them an shows how to deal with seasonal breaks.

2.3.3 Additional issues

The **third part** of the book deals with additional issues:

Presence of seasonality While the presence of Seasonality in `AirPassengers` is obvious, this is not always the case. If a series has no seasonal pattern, there is no need for a seasonal adjustment. It it is adjusted anyway, the process simply adds noise to the series, and should be avoided. Chapter 13 shows how seasonality can be detected and how to decide whether an adjustment should be made or not.

Annual constraining Usually, seasonal adjustment may affect the annual values of a time series. In part, this is by design. The number of weekdays may differ between years, so the adjusted annual values may be different too. In part, this may an artifact of the adjustment process. X13 offers tools to enforce the annual values of the adjusted series to be the same as the original one. Chapter 14 shows how to constrain annual value and whether it is a good idea to do so.

Indirect vs direct adjustment Often, seasonal adjustment may be performed on individual series, or on an aggregate of these series. X13 offers tools that let you compare these two possibilities. Chapter 15 discusses the options and help you to decide which one is better.

2.3.4 Quality assessment

Adjusting a series with automated procedure is easy. But is the resulting series a good adjustment? The **fourth part** helps you to decide between competing seasonal adjustment models.

Quality measures In the lower part, the summary of the adjustment model shows various quality measures: The AICc and BIC information criterion and the QS, the Box-Ljung and the Shapiro statistic. Non of them shows any significance (indicated by one or several stars), which is a good sign. Various quality measures and their interpretation is shown in Chapter 16.

Revisions When comparing seasonal adjustment models, the stability of the model and the series is often an important consideration. One does not want to get a very different series with a new data point. X13 offers tools to analyse revisions. Chapter 17 discusses them and helps you to decide which model to pick.

2.4 References

3 How to use the book

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

- Overview of the book
- The Book wants to give concrete advise in case of a problem.
- Ideally we want to have a quick check list that gives readers a starting point where to look for further advice. That could be something like a Cheat Sheet (https://www.rstudio.com/resources/cheatsheets/) with quick advise and chapter references.
- For each section, we want to provide a concrete and informative case study. Some examples are provided in the outline.

This text will focus on seasonal adjustment and its implementation in R.

Specifically, the audience will be both R users who want to learn about seasonal adjustment as well as seasonal adjustment practitioners, who are interested in using R. The book will be tailored to the practical applications of seasonal adjustment within R. It presents background material and references for the theoretically minded reader. The main focus, however, is on concrete problems and examples.

We will showcase methods through detailed examples with associated code. This presentation allows the academic level to be quite broad; understood by undergraduates all the way through advanced Ph.D. students.

X-13ARIMA-SEATS is one of, if not the most, widely used seasonal adjustment software within federal and statistical agencies. Moreover, there is a movement in statistical agencies toward the use of R and open-source products. This text is motivated to unify these two positions. Additionally, it and also serve the following:

1. Guide to professional seasonal adjustment with R To make the entry to the world of seasonal adjustment more accessible for those with an understanding of R. We leverage

the users R knowledge to more easily understand the input/output of the X-13ARIMA-SEATS program. We also will give an overview of other possibilities of seasonal adjustment in R (e.g. stl, JDemetra).

2. Focus on practitioner's problem To bridge an important gap in the training for many seasonal adjustment practitioners. The book addresses practical problems and shows how they can be addressed in X-13ARIMA-SEATS. The use of R allows them to have reproducible examples at hand.

3.1 Overview of the book

3.2 The seasonalbook package

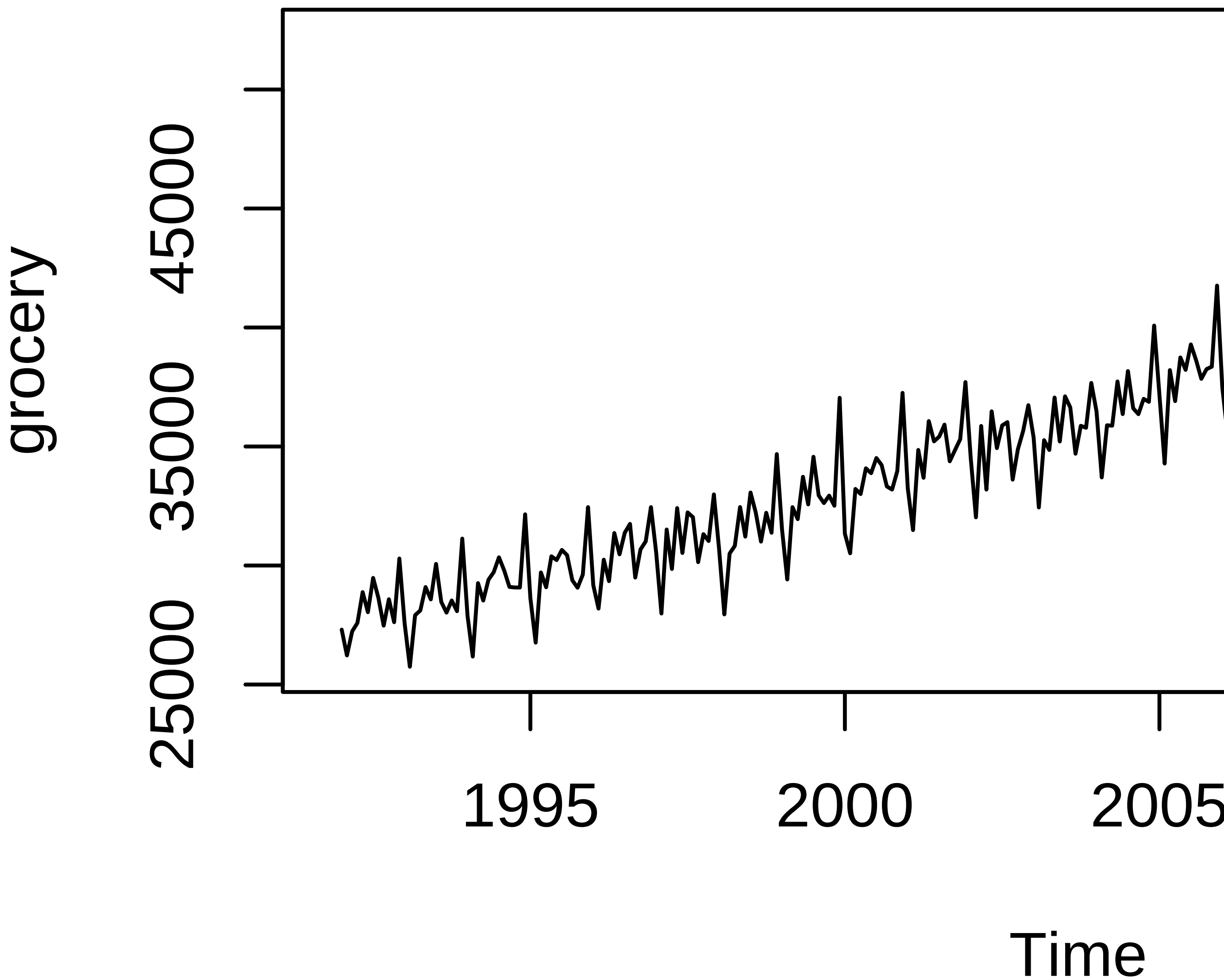
An R package that supplements "Seasonal Adjustment in R", and contains all data and examples.

To install:

```
remotes::install_github("christophsax/seasonalbook")
```

Example series:

```
library(seasonalbook)
plot(grocery)
```


grocery
250000
350000
450000
1995
2000
2005
Time

Some Examples: Chinese New Year, structural breaks, direct or indirect seasonal adjustment, SEATS or X-11.

Part II

Basics

Important

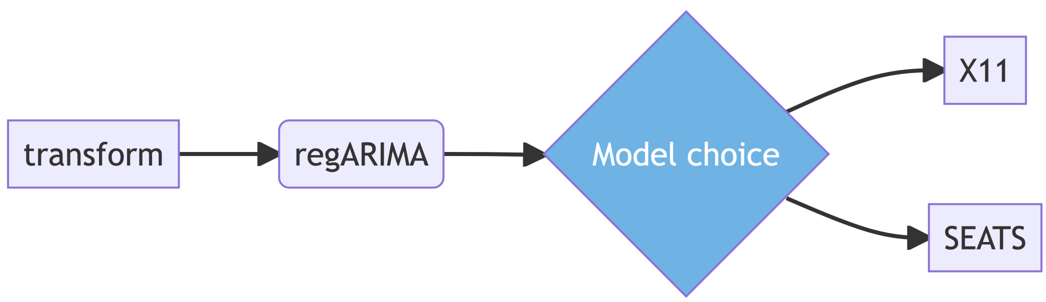
You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

This section gets readers familiar with X-13ARIMA-SEATS. It begins by explaining the history and pedagogy of the software (Chapter 4). This leads directly into discussing the principal elements of X-13ARIMA-SEATS.

4 Overview of the software

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

4.1 Elements of X13



4.2 Main user choices

4.2.1 Model

4.2.2 Transform

4.3 References

5 Transform

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

One of the first choices to make when modeling a time series is whether or not it needs to be transformed prior to modeling. There are two types of transformation types that typically occur within X13. The first is a prior modification. A prior modification scales each observation for known fixed effects. These effects can be well known and established such as length of a month/quarter and leap-year or more subjective such as a modification for a workers strike. We can think of prior modification factors of events or corrections made to your data that are fixed throughout the adjustment proeess. Additionally, these prior modification factors can be permanent (defualt) or temporary. The difference between permanent and temporary is wheither or not the effect is included back in the final seasonally adjusted series or not; permanent factors are excluded from the final seasonal adjustment while temporary are removed while calculating seasonal factors but then added back to the seasonally adjusted series. The second type is a nonlinear transformation applied to the observations. This is typically a choice between logarithmic transform and no transformation but for modeling can be any power of the Box-Cox transformation CITE.

5.1 Multiplicative or additive adjustment?

X-13A-S has a built in statisitcal test to decide between log and no transformatin. This is done with an information criteria based statistical test. The choice is made by comparing the AICC value CITE of an Airline model fit, or user specified ARIMA model, to the log transformed series and the original series. For all practical purposes this is an effective choice and can be left to the program to decide. Note, if your series has negative values it can not be log transformed and no transform is automatically selected by X-13 when automatic transformation is asked for. Other restrictions on the allowed transformtions exisit and can be found in CITE but these situations are rare. We can see the results of the transformation tests by looking at specific inputs to the UDG file.

```r
library(seasonal)
m <- seas(AirPassengers)
udg(m, c("aictest.trans.aicc.nolog", "aictest.trans.aicc.log"))
#> aictest.trans.aicc.nolog   aictest.trans.aicc.log
#>                1021.1919                 987.3845
```

We see the AICC for log transformation is lower and hence selected. The summary of the seasonal object `summary(m)` tells us this with `Transform: log` displayed. The automatically selected transformation can also be found in many other places such as the HTML output with `out(m)` or the udg with agruement name *aictrans* such as `udg(m, "aictrans")`.

The choice between log and none changes the type of seasonal decomposition that will occur and hence your interpretation of the seasonal factors. With no transformation, X13 will do an additive seasonal adjustment

$$X_t = T_t + S_t + I_t.$$

If a long transformation is selected, X13 will do a multiplicative adjustment

$$X_t = T_t \cdot S_t \cdot I_t.$$

This is major difference here is the way seasonal factors are interpreted and applied to the observed data to remove seasonality. For additive models the seasonal factor is subtracted from the original $X_t - S_t$. For example, an observed value of 100 with seasonal factor of 3.2 would result in a seasonally adjusted value of $100 - 3.2 = 96.8$. For multiplicative models the observed data is divided by the seasonal factor $\frac{X_t}{S_t}$. For example, an observed value of 100 with seasonal factor of 1.08 would result in a seasonally adjusted value of $100/1.08 = 92.59259$. Hence for multiplicative models values of $S_t > 1$, decrease the observed value and $S_t < 1$ increase it.

5.2 Transform options

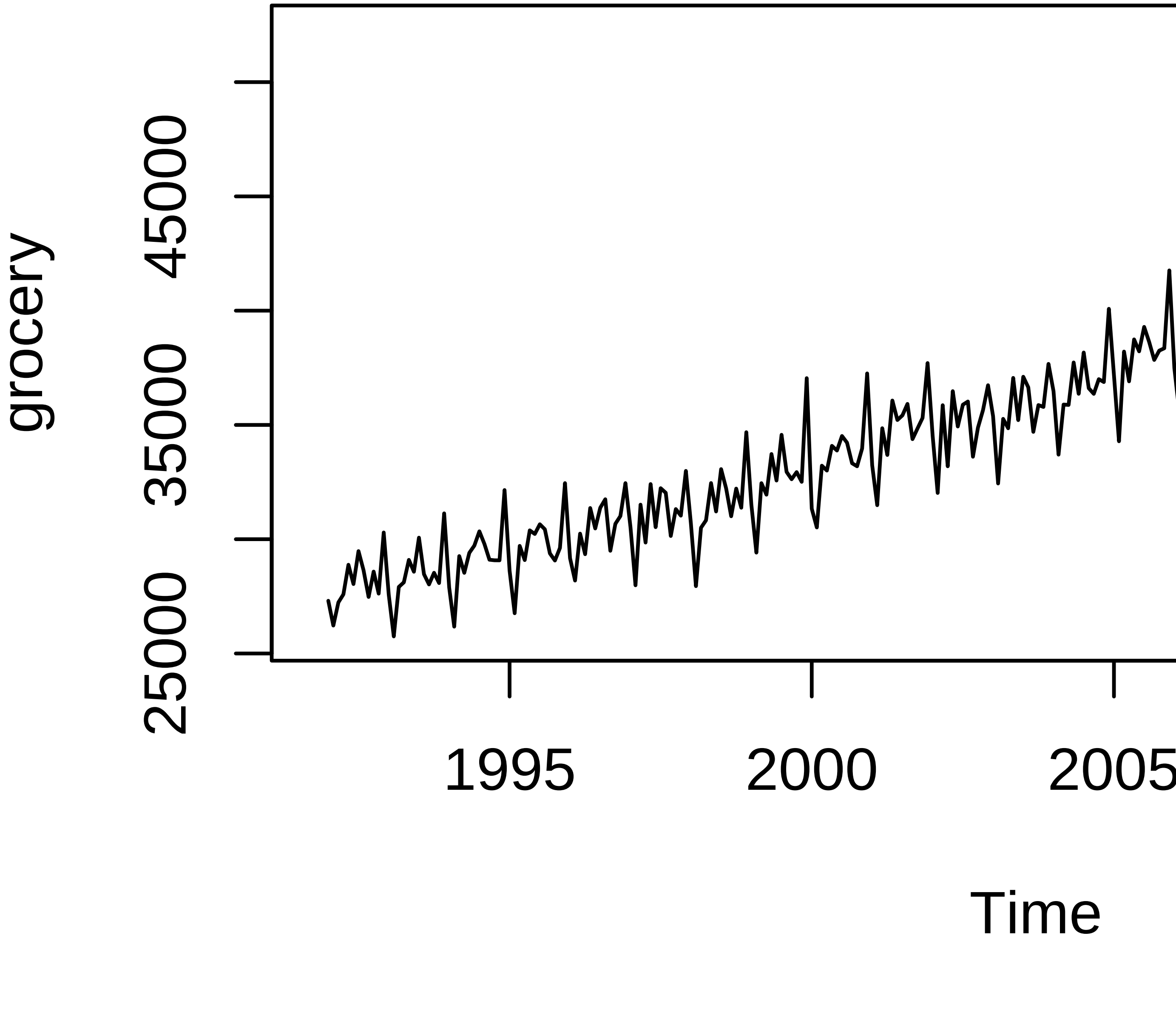
The `transform` spec controls these options. Some primary options within this spec are

Spec option	Use	Example values	default
function	specify transform	none, log, auto	none
data or file	specify prior adjustment factor	(1.2, 1.1, …, .99)	(1,1,…,1)
aicdiff	adjust tolerance of AIC test for log transform	0.0 3.0 -4.5	-2.0

5.3 Case Study

Consider the situaiton where you are trying to decide on transform choices for monthly retail grocery store data. The series `grocery` is part of the *seasonalbook* package.

```
library(seasonalbook)
plot(grocery)
```


grocery
250000
350000
450000
1995
2000
2005
Time

Visual inspection of the series shows no immediate reason to think we need to perform a log transform. There is possible seasonal hederoskadasity which could be mitigated by taking logs. Perform an X-11 adjustmnet with all of `seasonal` defaults.

```
m <- seas(grocery, x11 = "")
udg(m, c("aictest.trans.aicc.nolog", "aictest.trans.aicc.log"))
#> aictest.trans.aicc.nolog   aictest.trans.aicc.log
#>                 4202.960                 4201.042
```

This is interesting since the AICC for no transformation is lower than the AICC for log transform.

```
transformfunction(m)
#> [1] "log"
```

The default value for transform.aicdiff is -2 meaning the program slightly prefers log transform and the difference between the AICC values must exceed 2. In this situation we see the difference between the two AICC values is -1.917597. If you were to change this option to transform.aicdiff = 0 then the program selects no transform.

```
m2 <- seas(grocery, x11 = "", transform.aicdiff = 2)
transformfunction(m2)
#> [1] "none"
```

6 regARIMA Model

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

An important part of the X13 procedure is regARIMA modeling. As the name implies, there are two components that one needs to understand when fitting a regARIMA model; namely regression and ARIMA. In this chapter, we try to break these two components down to the most fundamental components without an overly technical exposition. Essentially, providing readers with enough information about each topic to understand the rest of this book or go off and perform satisfactory seasonal adjustment. The interested reader is encouraged to find material devoted to each of these components separately to more fully understand them.

ARIMA is an acronym describing the three parts of the modeling paridiam. AR = autoregressive, I = integrated (differenced), and MA = moving average. The prefix auto or "self", explains the AR portion perfectly. We model the current obervation with lagged values from the past. This is illustrated with the classic autoregressive model of order 1:

$$Y_t = \phi Y_{t-1} + a_t$$

where Y_t is the observed time series, ϕ is a coefficient to be estimated and $\{a_t\}$ is an uncorrelated sequence of errors similiar to that of standard linear regression. This model is notated AR(1). If instead of a single lag we used p lags, the model would be and AR(p) and have structure:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + a_t$$

where now we have p coefficients $\phi_1, \phi_2, \ldots, \phi_p$ to be estimated.

The moving average part of ARIMA model is similar in notation and reflects the number of lagged values of the error sequence should be included. For example, an MA(1) model with coefficient parameter θ is:

$$Y_t = a_t + \theta a_{t-1}$$

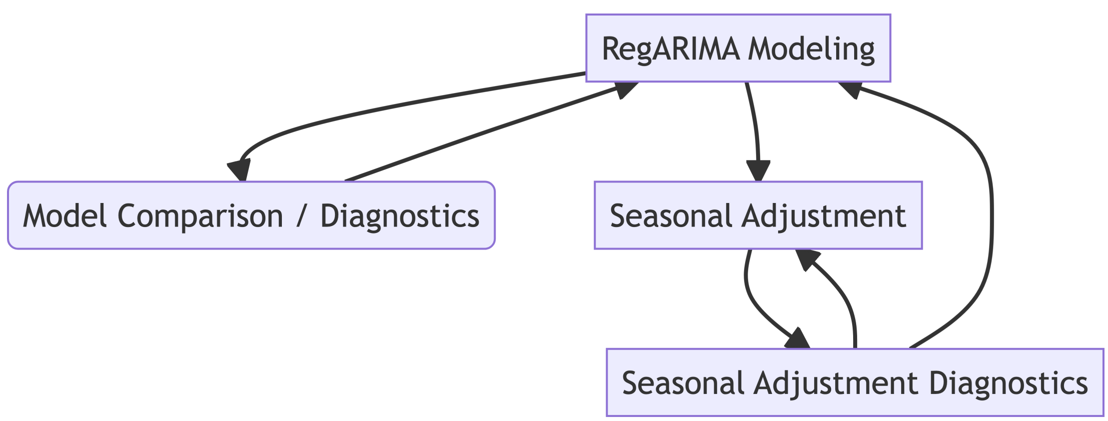
.

Note that instead of doing self-regression we include past values of the unobserved errors in the model at time t. If instead of a single lag we wanted q lags of the past error terms, we would have an MA(q) model:

$$Y_t = a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + \cdots + \theta_q a_{t-q}$$

.

Depending on if you ultimately choose to perform a SEATS or X-11 seasonal adjustment, your usage of a regARIMA can differ. Both seasonal adjustment methods will use the regARIMA model to forecast the series in order to apply symmetric filters and extract components. (but I thought SEATS was model-based? What do you mean SEATS uses filters? More on this in Chapter XXX). SEATS will use the ARIMA model to derive components for the trend, seasonal and irregular terms. The regARIMA model is also the place to include exogenous information or regressors into your seasonal adjustments. Regressors such as holiday effects, additive outliers, levels shifts, or any user-defined effect can be included in your regARIMA model. This can be thought of as solely a utility to forecast extend your series or a detection method to do inference. In practice it is useful to use regARIMA modeling to answer questions such as, "Does my series have trading day effects?" or "Does this outlier effect my results?". Another reason for regARIMA modeling is to include exogenous regression variables in our analysis. These include, but are not limited to, moving holiday effects, trading day, outliers and level-shifts. The regARIMA modeling takes place prior to seasonal and can be iterative until a final model is reached. This section will focus on the left self-loop of our seasonal adjustment flowchat:



The regARIMA take the form

$$f\left(\frac{Y_t}{D_t}\right) = \beta' \mathbf{X}_t + Z_t.$$

Here Y_t is the observed time series. The function f represents a transformation, most commonly used is the log transform ie $f(x) = \log(x)$. D_t is any intervention that has taken place prior to any transformation or modeling. This intervention is usually subjective and customized for individual series on an as-needed basis. For example, if a worker strike occurred in a specific industry that effected certain economic series. If no transformation or intervention is needed the model form is:

$$Y_t = \underbrace{\beta' \mathbf{X}_t}_{\text{Regression}} + \underbrace{Z_t}_{\text{ARIMA}}.$$

The regression variables appear in the columns of the design matrix $\mathbf{X}_t$ and Z_t is an ARIMA process. This last assumption on Z_t is what distinguished a regARIMA model from more classic linear models and multiple linear regression where error terms are assumed uncorrelated.

In order to achieve a suitable seasonal adjustment it is important to get the regARIMA model correct. For most dataset the built in automatic modeling features of the X13 program will be suitable to detect a reasonable model. This can be used as a starting point for more rigorous regARIMA model development or used as the final regARIMA modeling choice for your seasonal adjustment needs. We evoke automatic model identification through the XXX spec. The default behavior of the R seasonal package is XXX which includes automatic model identification.

Automatic and manual model choice

As an aside, the general rule is to not use automatic modeling in production. This mean, if you are going to include seasonal adjustment as part of a large scale data processing that occurs regularly (say monthly), then it is not advisable to have automatic model identification run every month. Instead, an alternative process, is to run automodel once and then fix the model choice in the XXX spec file. This does not need to be done manually since the `static()` function from the seasonal package can do this for you.

Outlier Type	Automatic Detection Available?
Additive outliers (AO)	Yes (default)
Level shifts (LS)	Yes (default)
Temporary level shifts (TL)	Yes
Temporary changes (TC)	No
Ramps (RP, QI, QD)	No
Seasonal outliers (SO)	No

Consider the default seasonal adjustment:

```r
library(seasonal)
m <- seasonal::seas(AirPassengers, x11 = "")
print(m$spc$automdl)
#> $print
#> [1] "bestfivemdl"
print(m$spc$arima)
#> NULL
```

Notice the value `NULL` indicates no ARIMA model is specified and the returned arguments for the automdl spec indicate it is active during the X13 run.

```r
seasonal::udg(m, "automdl")
#>           automdl
#> "(0 1 1)(0 1 1)"
```

Indicates that automatic modeling identified the (0 1 1)(0 1 1) model as the best choice. If we want to hardcode this model for subsequent runs, and turn off automatic model identification, this can be done via

```r
m_call <- seasonal::static(m)
#> seas(
#>   x = AirPassengers,
#>   x11 = "",
#>   regression.variables = c("td1coef", "easter[1]", "ao1951.May"),
#>   arima.model = "(0 1 1)(0 1 1)",
#>   regression.aictest = NULL,
#>   outlier = NULL,
#>   transform.function = "log"
#> )
m2 <- eval(m_call)
```

6.1 Outlier

There are many options you can modify when searching for outliers in your series. Some of the most practical options to start your exploration are the *type*, *critical value* and *span* that you would like to search.

Here is an example of using span to limit the outlier search to the last few years of a series:

```
m_span <- seas(AirPassengers,
  outlier.types = c("ao", "ls", "tc"),
  outlier.critical = 4.0,
  outlier.span = "1958.jan, ")
summary(m_span)
#>
#> Call:
#> seas(x = AirPassengers, outlier.types = c("ao", "ls", "tc"),
#>     outlier.critical = 4, outlier.span = "1958.jan, ")
#>
#> Coefficients:
#>                      Estimate Std. Error z value Pr(>|z|)
#> Weekday             -0.002644   0.000604  -4.377 1.20e-05 ***
#> Easter[1]            0.021321   0.008395   2.540  0.01110 *
#> MA-Nonseasonal-01    0.235404   0.083756   2.811  0.00495 **
#> MA-Seasonal-12       0.543743   0.074644   7.284 3.23e-13 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> SEATS adj.  ARIMA: (0 1 1)(0 1 1)  Obs.: 144  Transform: log
#> AICc: 965.3, BIC: 979.2  QS (no seasonality in final):    0
#> Box-Ljung (no autocorr.): 28.26   Shapiro (normality): 0.9829 .

m_nospan <- seas(AirPassengers,
  outlier.types = c("ao", "ls", "tc"),
  outlier.critical = 4.0)
summary(m_nospan)
#>
#> Call:
#> seas(x = AirPassengers, outlier.types = c("ao", "ls", "tc"),
#>     outlier.critical = 4)
#>
#> Coefficients:
#>                       Estimate Std. Error z value Pr(>|z|)
#> Weekday             -0.0029497  0.0005232  -5.638 1.72e-08 ***
#> Easter[1]            0.0177674  0.0071580   2.482   0.0131 *
#> AO1951.May           0.1001558  0.0204387   4.900 9.57e-07 ***
#> MA-Nonseasonal-01    0.1156204  0.0858588   1.347   0.1781
#> MA-Seasonal-12       0.4973600  0.0774677   6.420 1.36e-10 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#>
#> SEATS adj.  ARIMA: (0 1 1)(0 1 1)  Obs.: 144  Transform: log
#> AICc: 947.3, BIC: 963.9  QS (no seasonality in final):    0
#> Box-Ljung (no autocorr.): 26.65   Shapiro (normality): 0.9908
```

The default critical value is set based on the length of the outlier span. Notice the MA-Nonseasonal-01 value when comparing `m_span` with `m_nospan`. We see the choice of span, and ultimately the choise to include an outlier in your model can have a dramatic effect on the estimated regARIMA parameters.

6.2 How to use the regression spec

6.3 Table of all Regression effects available

6.4 Case study

Decide if you should include AO in May 2014. Construct a simple user defined regressor to handle specific issue.

7 SEATS

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Jan 21, 2023**.

- How to use the SEATS spec
- SEATS vs X11
- Case Study:

For SEATS, can be quite challenging since it relies heavily on seasonal ARIMA modeling.

8 X11

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Dec 21, 2022**.

One of the two main methods available in X-13ARIMA-SEATS to extract trend and seasonal components is the X-11 method. This is a nonparametric procedure that works by passing moving-average filters over the data to extract intended components.

In order to use symetric moving average filters at the end of the time series (current value), a regARIMA model is used to forecast extend the series. This RegARIMA model is where users can test for or specify outliers, trading day and moving holiday regressors in their adjustment. The forecast extended series is then used to filter.

Additionally, X-11 has a built in extreme value procedure included. This procedure identifies extremes and replaces. This results in a robust procedure that can automatically choose filters and identify extreme values without much user intervention. All that needs to be evoked beyond the default `seas()` call is to turn on the X11 spec option.

```
m <- seas(AirPassengers, x11 = "")
summary(m)
#>
#> Call:
#> seas(x = AirPassengers, x11 = "")
#>
#> Coefficients:
#>                      Estimate Std. Error z value Pr(>|z|)
#> Weekday            -0.0029497  0.0005232  -5.638 1.72e-08 ***
#> Easter[1]           0.0177674  0.0071580   2.482   0.0131 *
#> AO1951.May          0.1001558  0.0204387   4.900 9.57e-07 ***
#> MA-Nonseasonal-01   0.1156204  0.0858588   1.347   0.1781
#> MA-Seasonal-12      0.4973600  0.0774677   6.420 1.36e-10 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#>
#> X11 adj.  ARIMA: (0 1 1)(0 1 1)  Obs.: 144  Transform: log
#> AICc: 947.3, BIC: 963.9  QS (no seasonality in final):    0
#> Box-Ljung (no autocorr.): 26.65   Shapiro (normality): 0.9908
#> Messages generated by X-13:
#> Warnings:
#> - Visually significant seasonal and trading day peaks have
#>   been found in one or more of the estimated spectra.
```

Before further discussion about the details of the X-11 process, let us see what happened during this modeling run…

When using the x11 spec you can change the length of the filter used for the trend and seasonal components with the `trendma` and `seasonalma` arguments respectively. Additionally, `sigmalim` will control the amount of extreme value adjustment that is done during the seasonal adjustment.

8.1 Additive and multiplicative (again)

The X-13ARIMA-SEATS development was highly motivated to study economic time series. As such, the default seasonal adjustment mode is multiplicative due to most seasonal economic time series displaying seasonal fluctuations that increase and decrease along with the level of the time series.

If your series does not have this feature then additive adjustment might be more appropriate. This can be changed in the `mode` argument of the x11 spec. For example, `seas(x, x11.mode = 'add')` will perform an additive x11 run. There exist two other models for decomposition, pseudo-additive and log additive. These are less common than additive and multiplicative models and are not the focus of this text. If your series has some extremely small values in certain months (quarters) then pseudo-additive models could be worth further investigation. It has been observed that when multiplicative seasonal adjustment produces more extreme values in conjunction with small seasonal factors then pseudo-additive adjustment should be explored. NEED TO SHOW USERS HOW TO VIEW THEIR EXTREME VALUES - D8.B TABLE DESIGNATIONS NEXT TO OBSERVATIONS. A good reference on the subject is Baxter (1994).

8.2 Filter length

The X11 spec also allows users to control the length of the trend and seasonal moving average filters used during the adjustment. Generally speaking, longer filters imply a more stable seasonal component and shorter filters a more changing seasonal pattern. Of course, a longer filter will use more data for the caluclation of components at each time point. This is an important observation and understanding it might help a user decide on a short or long filter. Since longer filters use more data there tend to be smaller revisions when a new data point is added. However, there will be revisions to data values further back.

A shorter filter is just the opposite, they tend to produce smaller revisions but they do not extend as far back into the series. If a filter is not choosen by the user then automatic filter selection is used. To understand the length of a filter let's look at the (finite) number of choice available in during an x11 adjustment. Table 8.1 shows the different filters available for the seasonal component and the trend component.

Table 8.1: Filters available in X11

Value	Description
`s3x1`	3×1 moving average
`s3x3`	3×3 moving average
`s3x5`	3×5 moving average
`s3x9`	3×9 moving average
`s3x15`	3×15 moving average
`stable`	Stable seasonal filter. A single seasonal factor for each calendar month or quarter is generated by calculating the simple average of all the values for each month or quarter (taken after detrending and outlier adjustment).
`x11default`	A 3×3 moving average is used to calculate the initial seasonal factors in each iteration, and a 3×5 moving average to calculate the final seasonal factors.

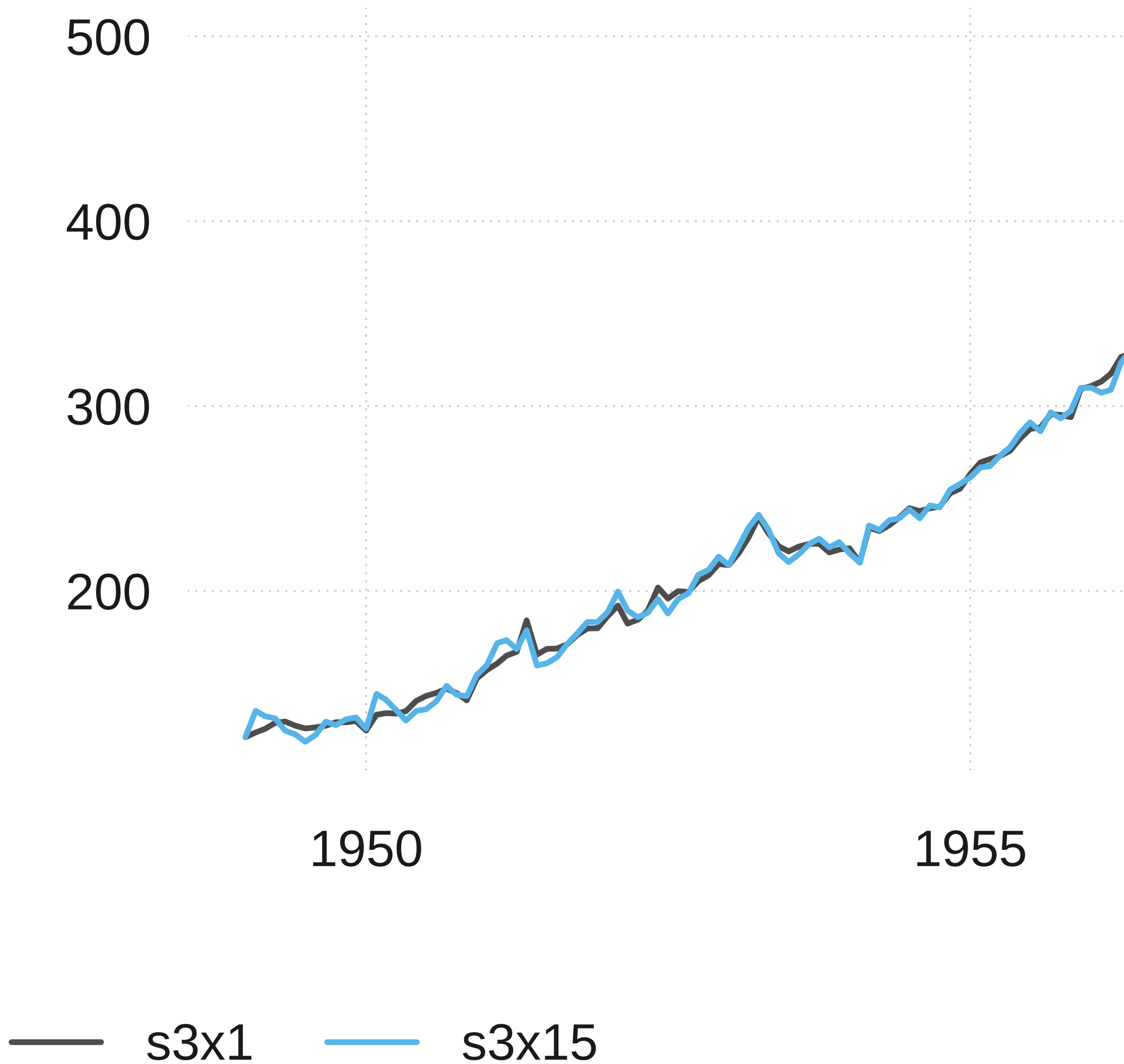
8.3 Extrem value

⚠ Related concept: Outlier

FIXME: explain how this is related to transform.

8.4 Case study

```r
tsbox::ts_plot(
  s3x1 = predict(seas(AirPassengers, x11.seasonalma = "s3x1")),
  s3x15 = predict(seas(AirPassengers, x11.seasonalma = "s3x15"))
)
```


500
400
300
200
1950
1955
s3x1
s3x15

Part III

Data Problems

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

In part III we look at more in-depth at practical issues with seasonal adjustment. The focus is on concrete solutions to each situation presented. Each subsection will prominently feature a case study dedicated to each problem.

9 Irregular holidays

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.
> **It is part of the course materials intended for Jan 21, 2023**.

- Why should we adjust for holiday effects
- Easter adjustment
- User defined adjustments (Chinese New Year, Diwali)
- Case Study: How to adjust for Ramadan (which is connected with some additional challenges)

10 Trading days

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- Why should we adjust for trading day effects
- Seven or two coefficient trading day
- Using country specific calendars
- Case Study: Movie tickets (or another series with very clear trading day effects)

11 Outliers

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- Why care about outliers?
- Additive outliers, level shifts, temporary changes

12 Seasonal breaks

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- Why to care about seasonal breaks?
- Detection of seasonal breaks
- Correction for seasonal breaks

Part IV

Other Issues

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

Part IV investigates more holistic issues that practitioners face. The main focus is to give classical methodology to answer their problems. Since these types of issues can be highly specialized, we concentrate on known solutions to the topics.

13 Presence of seasonality

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

Should a series be seasonally adjusted at all?

X-13AS removes seasonality from series, even if a series is not seasonal from the beginneing. If a series is not seasonal, the resulting series may be bad.

Fortunatly, X-13 contains a few tests that help users to decide if a series is is seasnal or not.

Before appying X-13AS ist may be necessary to descide if the series is seasonal.

13.0.1 Avalabele Tests

X13 offers has several formal checks:

- qs test
- ids
- m7

The *ids* test is closely connected to *m7*, but the *QS* test is quite different. Which tests are preferrable, and how should a user descide if the tests are not aligned?

13.0.2 ids test

http://www.ons.gov.uk/ons/guide-method/method-quality/general-methodology/time-series-analysis/guide-to-seasonal-adjustment.pdf

> From ONS 18.2 A general criterion for existence of seasonality
>
> Empirical research showed that the most appropriate test for seasonality is the "Combined test for the presence of identifiable seasonality", given after table D8 of the output. In particular, one of the following statements will always appear:

1. IDENTIFIABLE SEASONALITY PRESENT
2. IDENTIFIABLE SEASONALITY PROBABLY NOT PRESEN
3. IDENTIFIABLE SEASONALITY NOT PRESENT

It is recommended that a series is adjusted in the first two cases and not adjusted in the last one. However there are two cases where one might need to deviate from this practice:

This is the ids test shown below

13.0.3 Case Study

14 Annual constraining

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- Should the annual values be restrained?
- How to use the force spec

15 Indirect vs direct adjustment

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- Should the subcomponents of a series be adjusted separately?

Part V

Quality assessment

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

This section focuses on diagnostic tools for seasonal adjustment. This will be written as a stand-alone section as well as a continuance of prior sections. The idea here is that many readers may be interested in checking the quality of their adjustments but not need help performing it.

16 Quality measures

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- What is a good seasonal adjustment?
- M statistics
- Other statistics available in X13

17 Revisions

> **!** Important
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

- How to measure revisions?
- Should a model be re-estimated each period?
- How to use the slidingspan and history spec

Part VI

The future of seasonal adjustment

> **Important**
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter is currently a dumping ground for ideas, and we don't recommend reading it.

This short section outlines the future projects in the seasonal adjustment field. Daily or multiple seasonal adjustment plays a major role here. Ideally, examples of how to solve these problems are given.

- Daily adjustment
- Multivariate seasonal adjustment
- Other methods

Status of the book

Table 17.1: Current status of sections

section name	status	due date
A minimal example	drafting	2022-12-21
How to use the book	drafting	2022-12-21
Overview of the software	drafting	2022-12-21
Transform	drafting	2022-12-21
X11	drafting	2022-12-21
regARIMA Model	drafting	2022-12-21
Holidays	drafting	2023-01-21
SEATS	drafting	2023-01-21

Caporello, Gianluca, Agustin Maravall, and Fernando J Sánchez. 2001. "Program TSW Reference Manual." 0112. Banco de España Madrid. https://ideas.repec.org/p/bde/wpaper/0112.html.

Dagum, Estela Bee. 1980. *The x-11-ARIMA Seasonal Adjustment Method*. Statistics Canada, Seasonal Adjustment; Time Series Staff.

Findley, David F, Brian C Monsell, William R Bell, Mark C Otto, and Bor-Chung Chen. 1998. "New Capabilities and Methods of the x-12-ARIMA Seasonal-Adjustment Program." *Journal of Business & Economic Statistics* 16 (2): 127–52.

Monsell, B. 2007. "The x-13A-s Seasonal Adjustment Program." In *Proceedings of the 2007 Federal Committee on Statistical Methodology Research Conference*. http://www.fcsm.gov/07papers/Monsell.II-B.pdf.

National Bank of Belgium, Deutsche Bundesbank, Eurostat. 2017. *JDemetra+: Econometric Software for Seasonal Adjustment and Other Time Series Methods*. Eurostat. https://ec.europa.eu/eurostat/cros/content/download.

Sax, Christoph, and Dirk Eddelbuettel. 2018. "Seasonal Adjustment by X-13ARIMA-SEATS in R." *Journal of Statistical Software* 87 (11): 1–17. https://doi.org/10.18637/jss.v087.i11.