Seasonal Adjustment in R

Table of contents

Welcome **4**
- About the book . . . 4
- Key features of the book . . . 4
- Acknowledgements . . . 4

I Introduction **5**

1 A minimal example **7**
- 1.1 Installation . . . 7
- 1.2 A minimal example . . . 7

2 How to use the book **11**
- 2.1 Supplementary material . . . 12
- 2.2 For whom? . . . 12

II Basics **13**

3 Overview of the software **15**

4 Transform **16**
- 4.0.1 Case Study . . . 17

5 SEATS **21**

6 X11 **22**

7 regARIMA Model **24**

III Data Problems **28**

8 Irregular holidays **30**

9 Trading days **31**

10 Outliers 32

11 Seasonal breaks 33

IV Other Issues 34

12 Should a series be seasonally adjusted at all? 36
12.0.1 Avalabele Tests 36
12.0.2 ids test 36
12.0.3 Case Study 37

13 Annual constraining 38

14 Indirect vs direct adjustment 39

V Quality assessment 40

15 Quality measures 42

16 Revisions 43

VI The future of seasonal adjustment 44

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

Acknowledgements

Part I

Introduction

This chapter focuses on explaining the basics of seasonal adjustment and gets the reader involved with a minimal working example. It keeps the technical jargon to a minimum.

1 A minimal example

> **Note**
>
> You are reading an early draft of *Seasonal Adjustment in R*. This chapter should be readable but needs polishing.
> **It is part of the course materials intended for Dec 21, 2022**.

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.

To model the underlying structure of these series, any regular (seasonal) patterns are estimated and removed from the data. For example, to see if the economy is moving out of a recession during certain months, one wants the labor market data be free from such seasonal effects. Seasonal adjustment decomposes a time series into a trend, a seasonal and an irregular component and removes the seasonal component from the data.

1.1 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")
```

1.2 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:

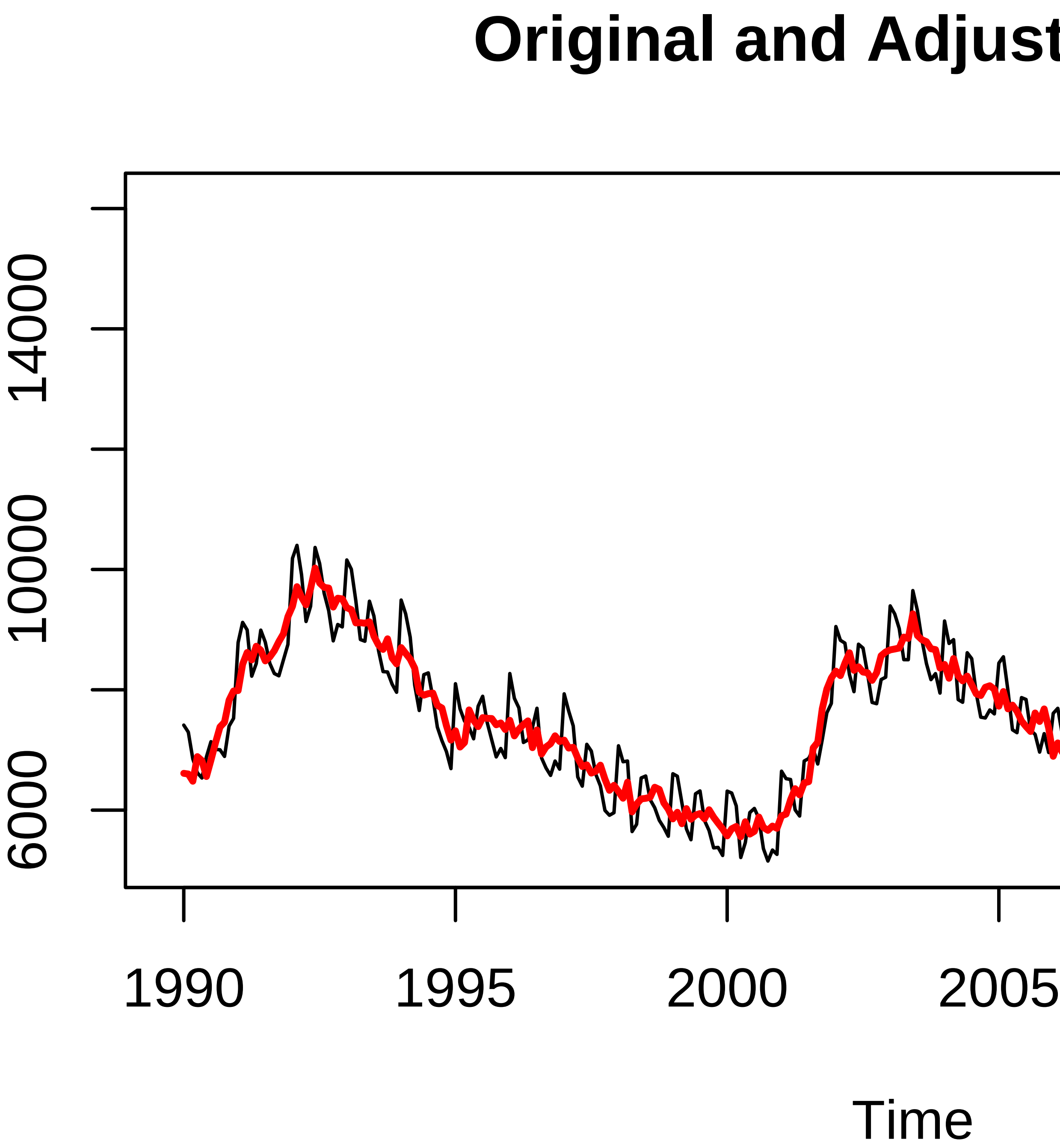
```r
seas(unemp)
#>
#> Call:
#> seas(x = unemp)
#>
#> Coefficients:
#> AR-Nonseasonal-01  MA-Nonseasonal-01     MA-Seasonal-12
#>            0.9436             0.8254             0.8507
```

The first argument of seas is a time series of class “ts”. The unemp example series measures US unemployment and is included in seasonal. The function returns an object of class “seas” that contains the necessary information on the adjustment performed on this time series.

```r
m <- seas(unemp)
```

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.

```r
plot(m)
```



Original and Adjust
6000
10000
14000
1990
1995
2000
2005
Time

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

```r
summary(m)
#>
#> Call:
#> seas(x = unemp)
#>
#> Coefficients:
#>                    Estimate Std. Error z value Pr(>|z|)
#> AR-Nonseasonal-01  0.94360    0.03441   27.43   <2e-16 ***
#> MA-Nonseasonal-01  0.82540    0.05654   14.60   <2e-16 ***
#> MA-Seasonal-12     0.85071    0.03362   25.30   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> SEATS adj.  ARIMA: (1 1 1)(0 1 1)  Obs.: 323  Transform: none
#> AICc:  4324, BIC:  4339  QS (no seasonality in final):    0
#> Box-Ljung (no autocorr.): 22.04   Shapiro (normality): 0.9946
```

2 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.

- 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.

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

2.1 Supplementary material

All material from the text will be made available to the reader. This includes but is not limited to:

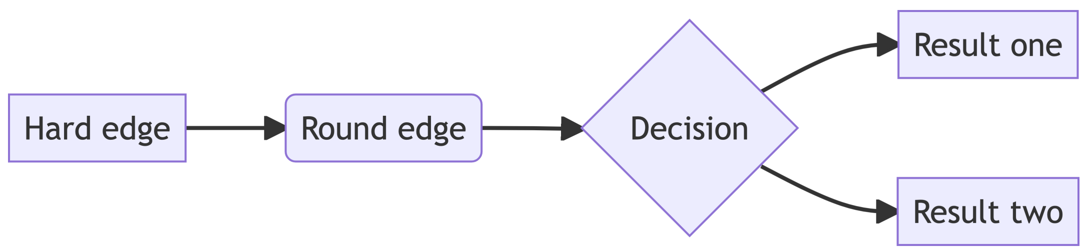
- R Package to accompany the book, containing all data and examples
- Interactive website, on which the examples can be run (similar to www.seasonal.website)

2.2 For whom?

There are two primary audiences:

1. 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.

2. 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.



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. This leads directly into discussing the principal elements of X-13ARIMA-SEATS.

3 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.

- History of the software
- Elements of the software
- Overview of main choices a user needs to make

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.

R (R Core Team 2017) offers several possibilities to perform seasonal adjustment in the stats package included with R. The decompose function uses filtering to split a time series into a trend, a seasonal and an irregular component. An alternative method is stl, which uses local regressions (and has some extensions offered by the stlplus package by (Hafen 2016). While both methods allow a multiplicative or an additive decomposition, these methods are somewhat limited in modeling actual data series.

The programs used by statistical offices provide an extensive toolbox to deal with many advanced (and more irregular) aspects of seasonality, such as trading day adjustments, moving holiday adjustments or automated outlier detection.

4 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.

- Discuss multiplicative vs additive adjustment
- How to use the transform spec
- This is our first chance to mention the idea of test first then hard-code in production
- Case Study idea: Decide between log vs non-log transformation

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.

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.

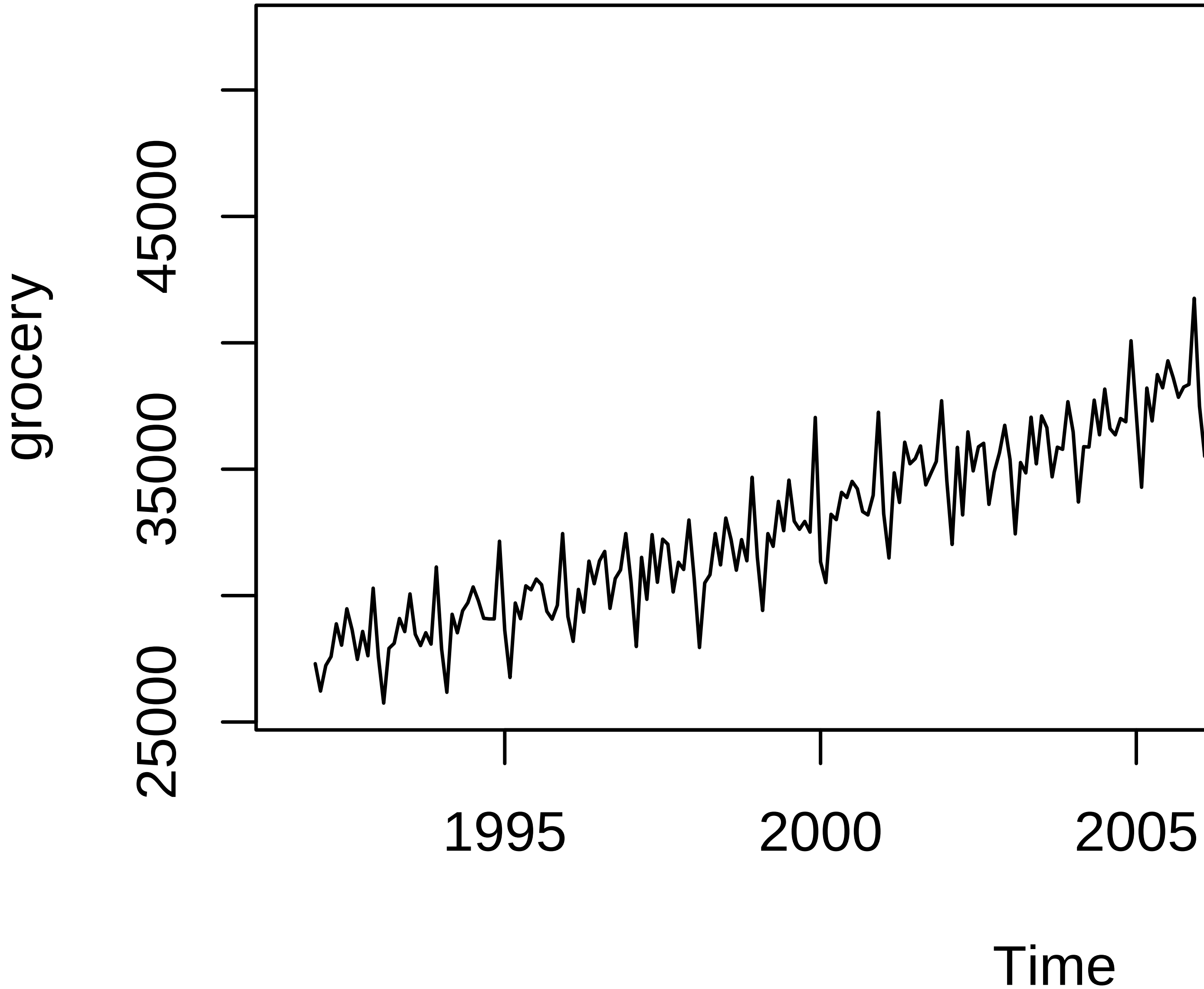
The `transform` spec controls these options. Some primary options within this spec are

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

4.0.1 Case Study

Consider the situaiton where you are trying to decide on transform choices for monthly retail grocery store data.

```r
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"
```

5 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.

- 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.

6 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.

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.

```
library(seasonal)
m <- seas(AirPassengers, x11 = "")
```

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.

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).

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. There are different filters used for the seasonal component and the trend component.

Seasonal Filter	Description
s3x1	3x1 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.

- How to use the X11 spec
- Case Study: Changing the length of trend and/or seasonal filter

7 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.

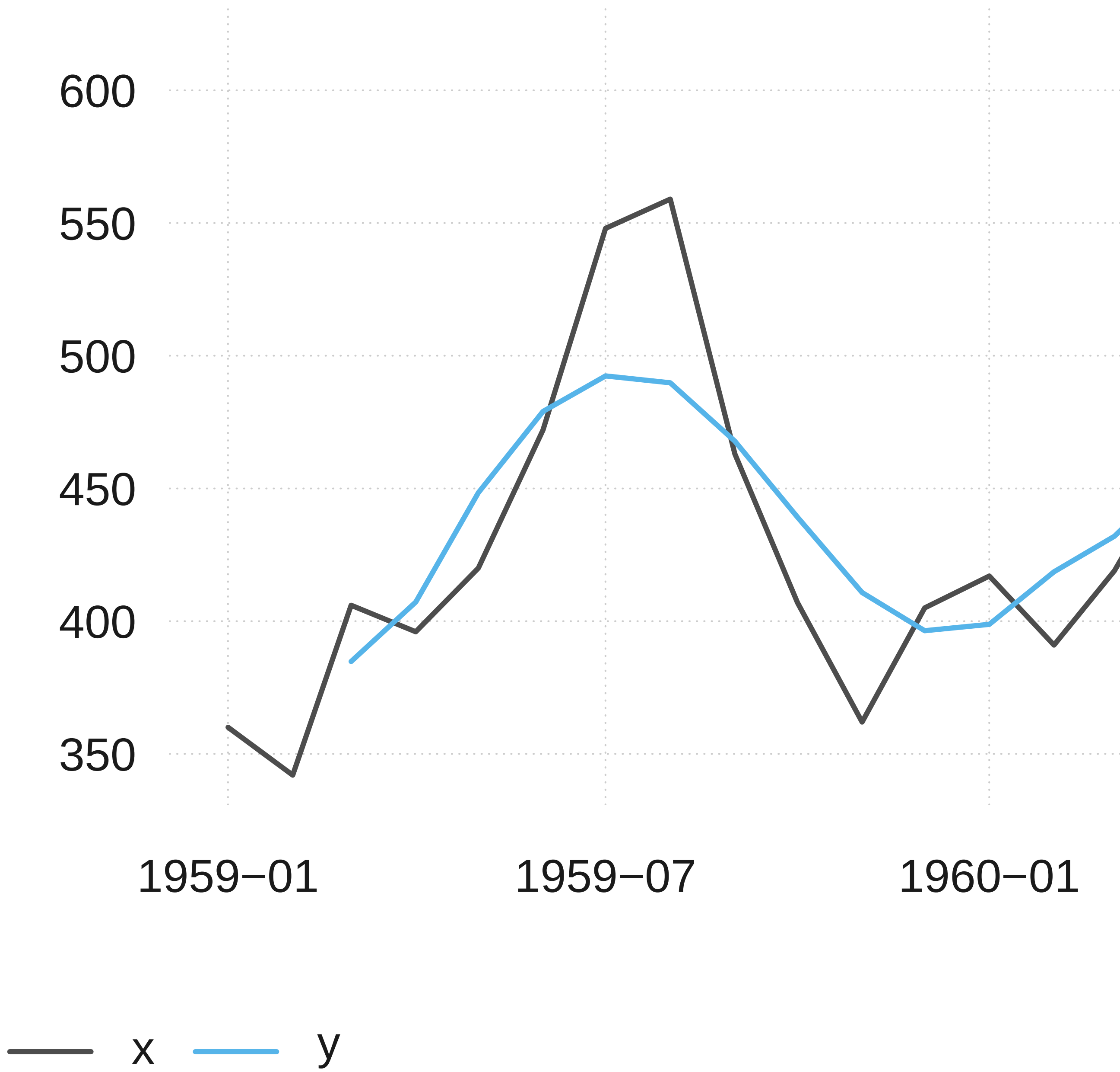
- Idea of regARIMA

The most fundamental procedure involved in X-13 is applying moving average filters. When symmetric filters are used we "lose" observations at the end of the series. Consider the simple moving average filter

$$Y_t = \frac{1}{5}X_{t-2} + \frac{1}{5}X_{t-1} + \frac{1}{5}X_t + \frac{1}{5}X_{t+1} + \frac{1}{5}X_{t+2}.$$

```r
library(seasonal)

x <- window(AirPassengers, start = c(1959, 1), end = c(1960, 12))
y = filter(x, filter = rep(1/5, 5), sides = 2)
tsbox::ts_plot(cbind(x, y))
```


600
550
500
450
400
350
1959−01
1959−07
1960−01
x
y

The output of the filter Y_t is shorter than X_t by 4 total observations. Two at the beginning and two at the end.

This is one of the primary reasons we consider regARIMA models when discussing seasonal adjustment; the observed time series is forecast extended to allow for symmetric moving average filters to be used all the way to the end of the series. 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 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 $f(x) = \log(x)$. 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. 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. For example, consider the default seasonal adjustment

```
m <- 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
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 <- 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)
```

- How to use the regression spec
- Table of all Regression effects available in X13 (steal from CR slides)
- Case studies: Decide if you should include AO in May 2014. Construct a simple user defined regressor to handle specific issue.

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.

8 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.

- 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)

9 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)

10 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

11 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.

12 Should a series be seasonally adjusted at all?

> ! 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.

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.

12.0.1 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?

12.0.2 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

12.0.3 Case Study

13 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

14 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.

15 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

16 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

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.

Hafen, Ryan. 2016. *stlplus: Enhanced Seasonal Decomposition of Time Series by Loess*. https://CRAN.R-project.org/package=stlplus.

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.