Time Series Decomposition

DATA 624
Predictive Analytics

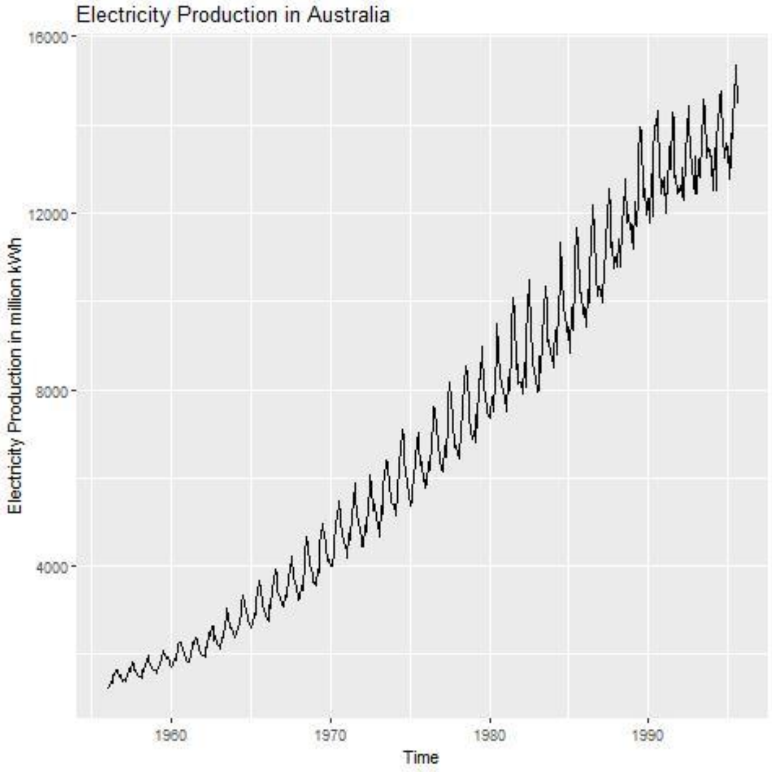
Dan Wigodsky
Sarah Wigodsky

Time Series Decomposition

Time series decomposition involves breaking a time series up into a seasonal component, trend and what remains after those two components are removed.

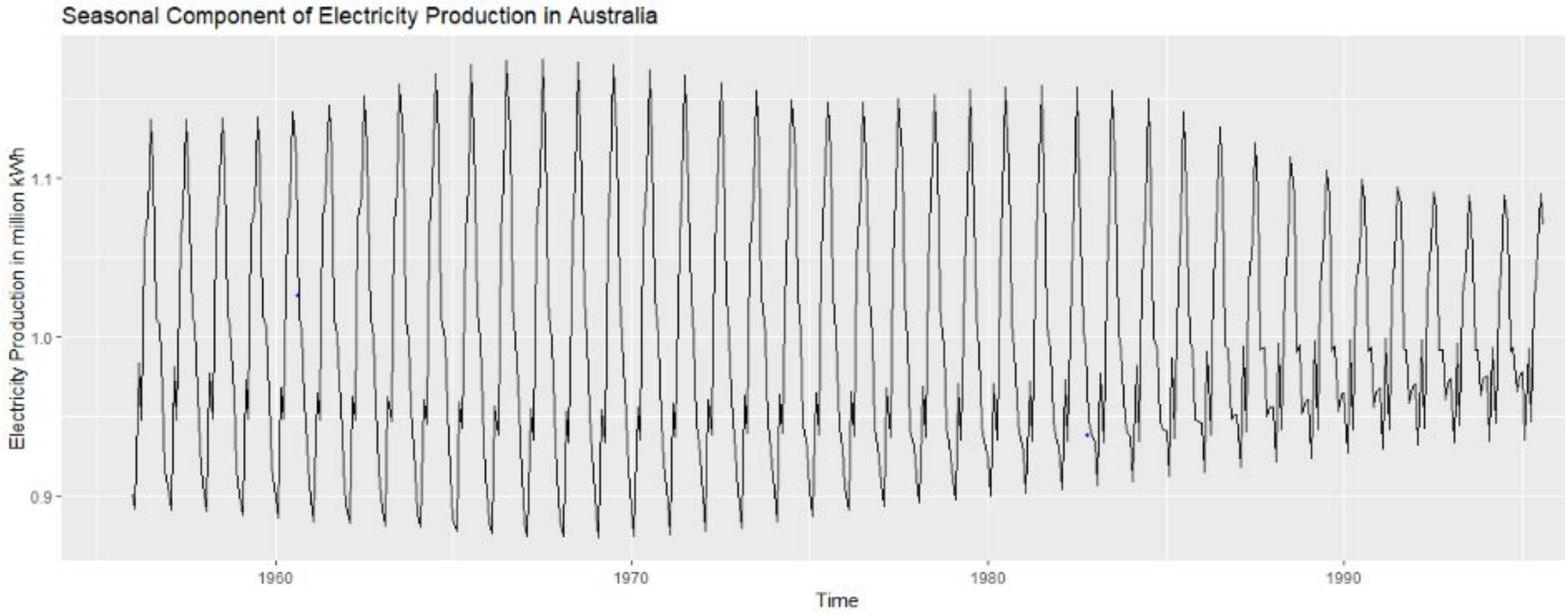
Time Series

A time series is composed of set of observations of a variable that are ordered in time and dependent through time.



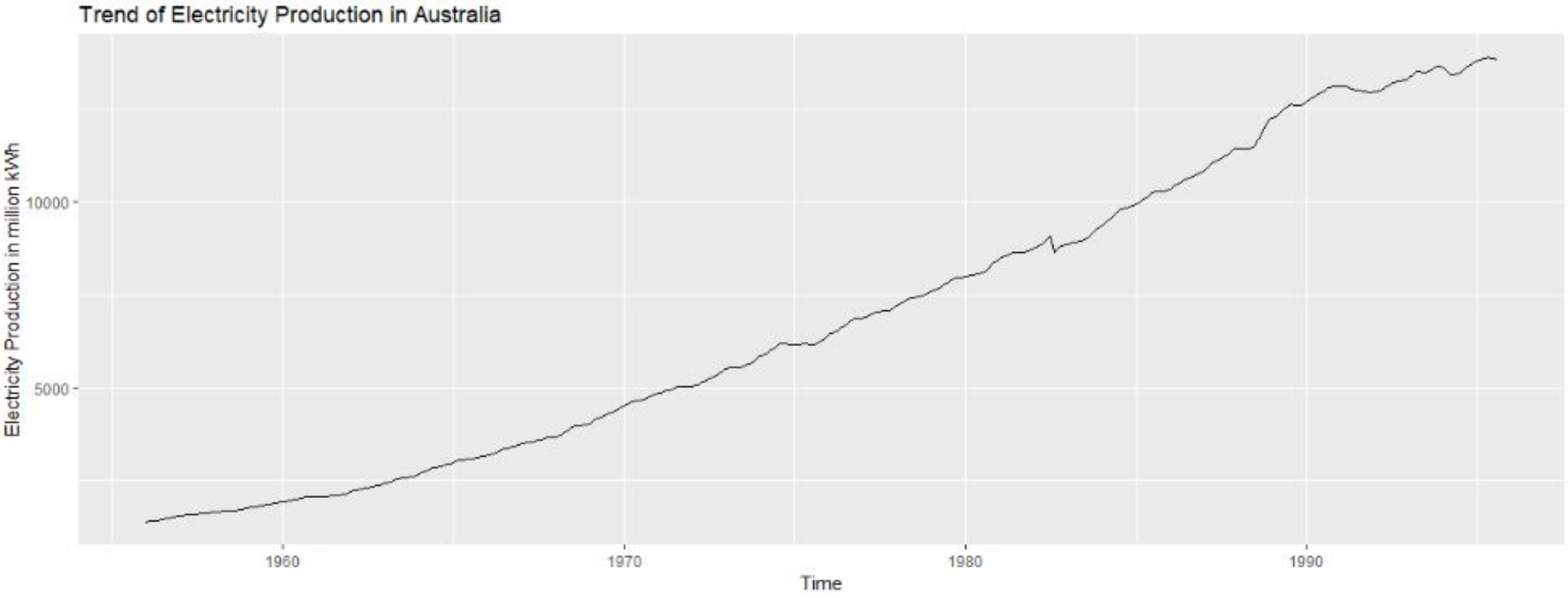
Seasonal Component

Data follows a pattern with a fixed frequency



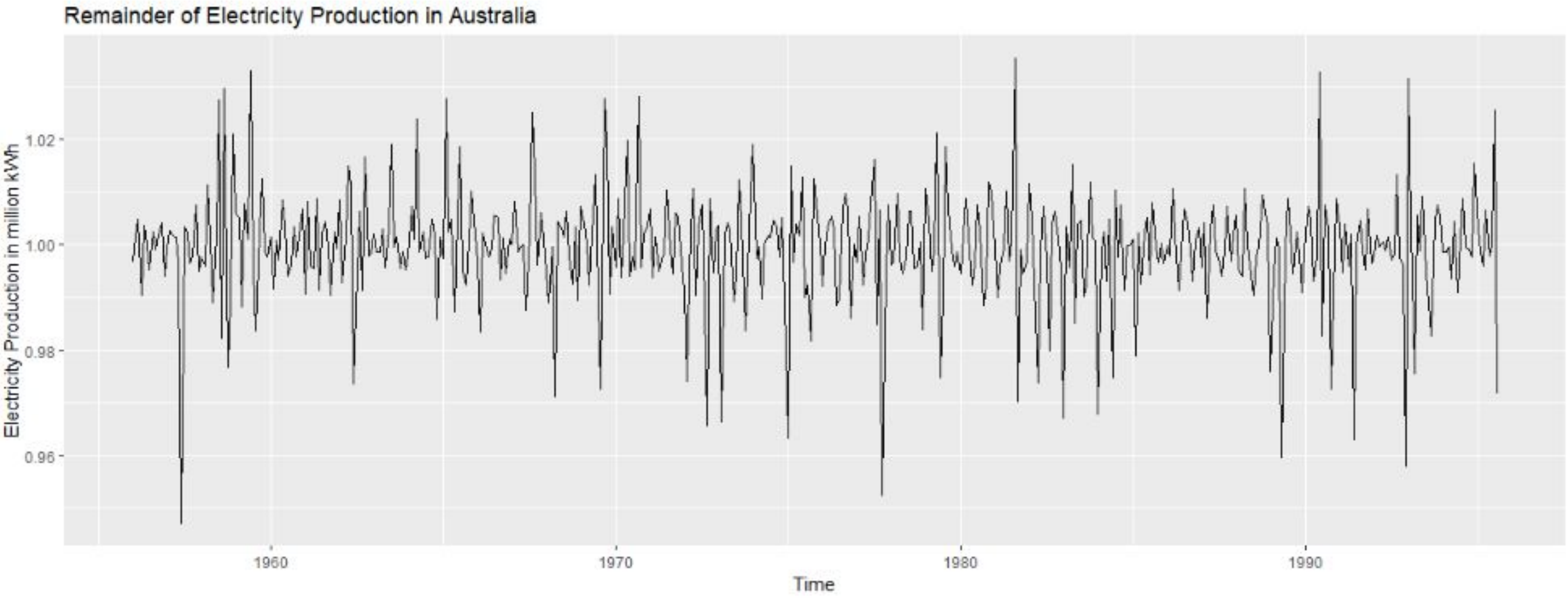
Trend-Cycle Component

Data follows a long term increasing or decreasing pattern



Remainder Component

Variations impacting individual variables



Additive Decomposition

$$y_t = \text{Seasonal}_t + \text{Trend}_t + \text{Remainder}_t$$

Additive Decomposition

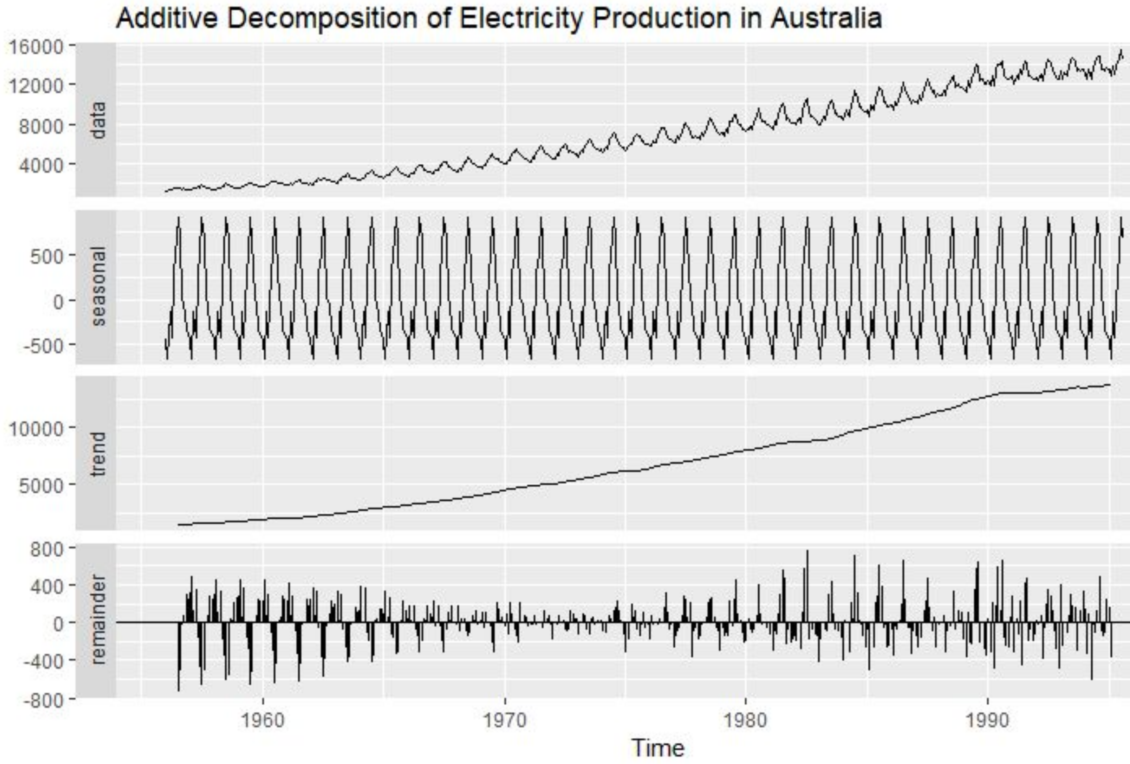
library(seasonal)

elec %>%

decompose(type="additive") %>%

autoplot() +

ggtitle("Remainder of Electricity Production in Australia")



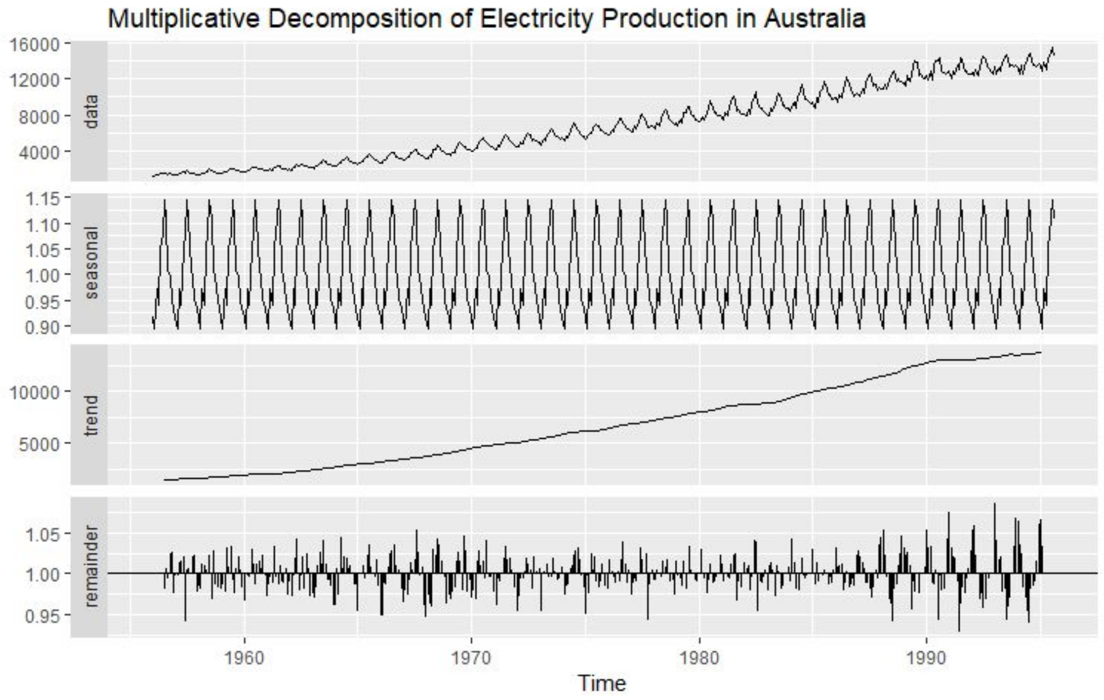
Multiplicative Decomposition

$$y_t = \text{Seasonal}_t \times \text{Trend}_t \times \text{Remainder}_t$$

$$\log y_t = \log S_t + \log T_t + \log R_t$$

Multiplicative Decomposition

library(seasonal)



elec %>%

decompose(type="multiplicative") %>%

autoplot() +

ggtitle("Remainder of Electricity Production in Australia")

Disadvantages of Classical Decomposition

- Relies on moving averages so there are no values for the trend at beginning and end of the time period
- Trend is over smoothed
- Considers the seasonal component to be constant

X-11 Decomposition

- Trend-cycle values exist for all observations
- Variations in trend are visible
- Model can accommodate variations within the day and the effects of holidays
- Used for additive and multiplicative decomposition

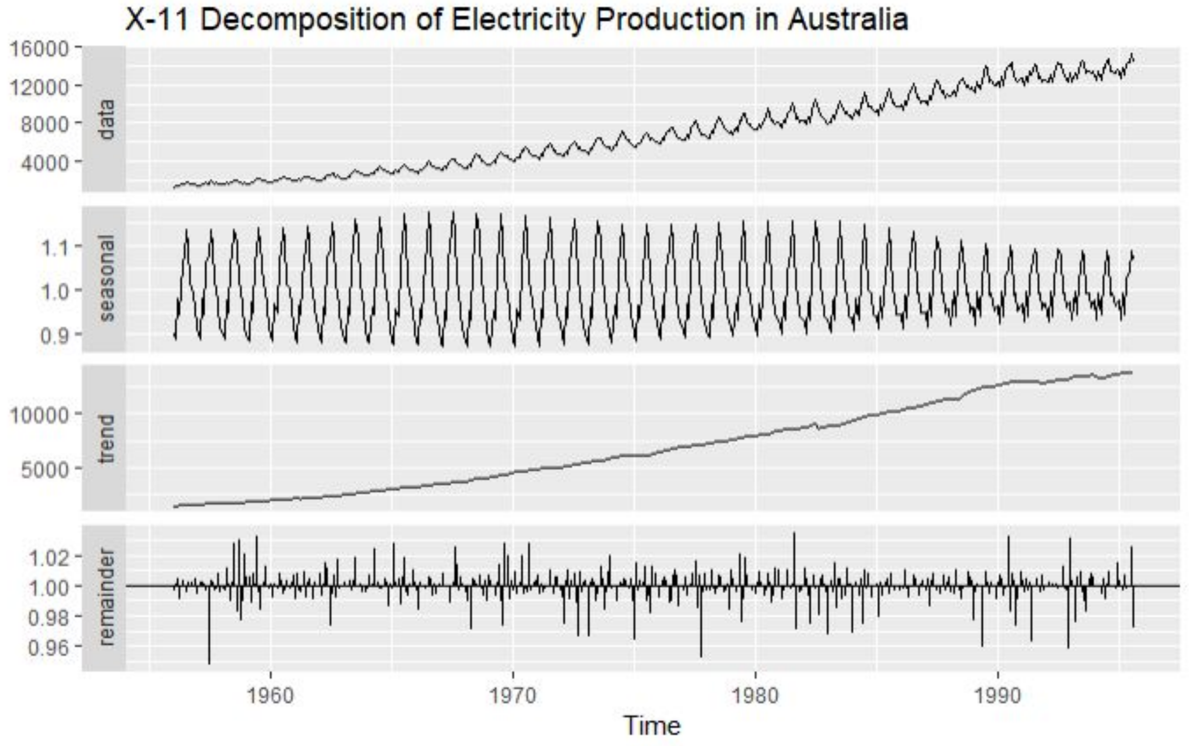
SARIMA time series

- autoregressive component compares to previous values
    - values are regressed on previous values
    - an AR- 1 model shows correlation with previous value, but not 2 before
    - reference is to x steps backward
- moving average component is headed in a direction
    - the accumulation of changes
    - mean is changing over time
- integrated series can be formed by differencing
    - new series is a series of changes from step to step
- S indicates a seasonal ARIMA model was used

X-11 Decomposition

library(seasonal)

```
elec %>%
seas(x11="") %>%
autoplot() +
ggtitle("X-11 Decomposition of Electricity Production in Australia")
```



Subseries Plot of X-11 Decomposition

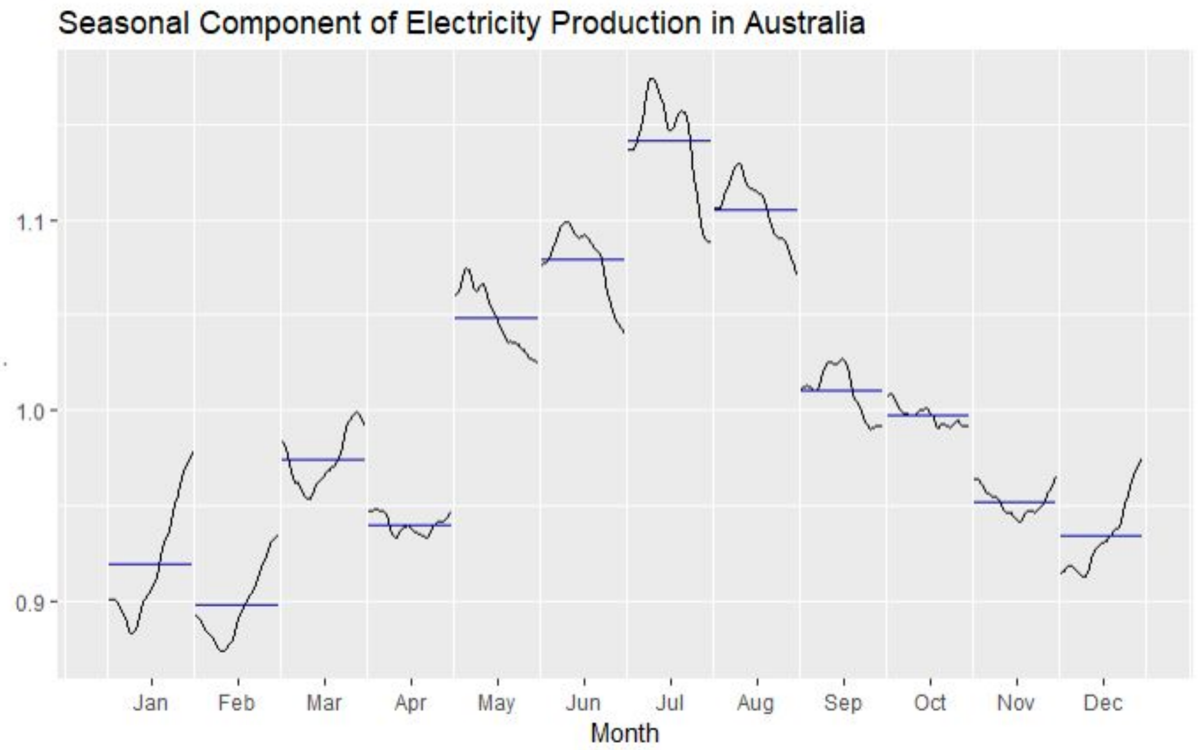
```
elec %>%

seas(x11="") %>%

seasonal() %>%

ggsubseriesplot() +

ggtitle("Seasonal Component of Electricity Production in Australia")
```



SEATS (Seasonal Extraction in ARIMA Time Series) Decomposition

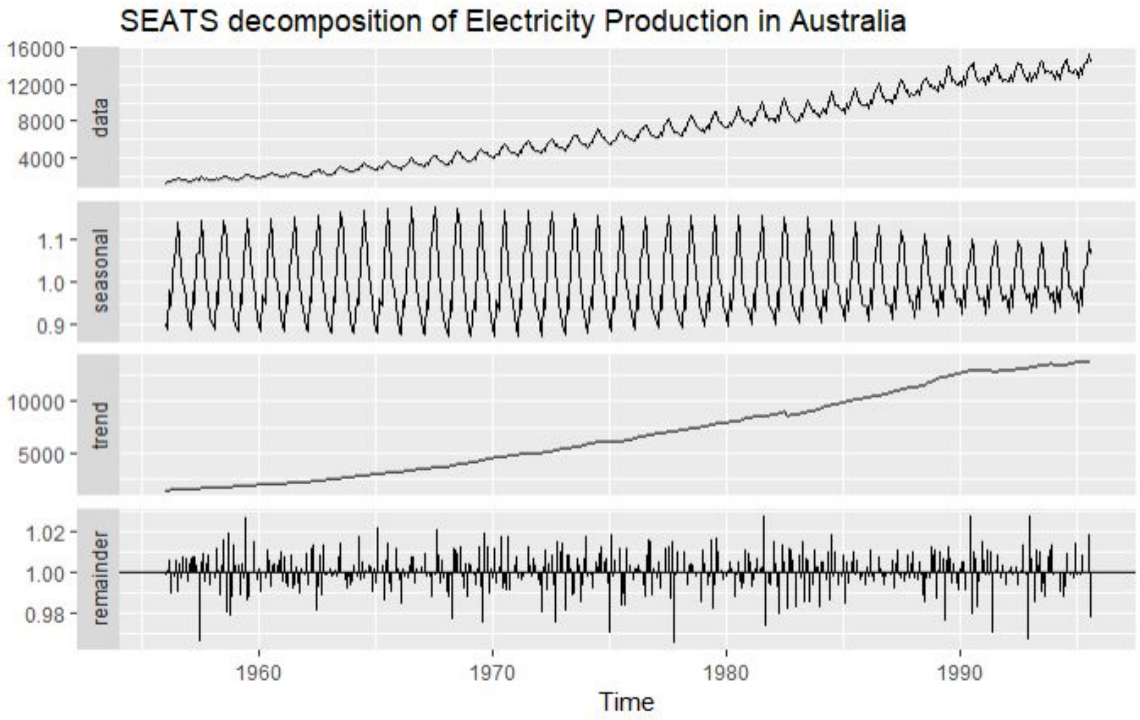
- SEATS decomposition can be used with quarterly or monthly data

$$y_t = Seasonal_t + Trend_t + Remainder_t$$

- ARIMA (AutoRegressive Integrated Moving Average) Model is used to build each of the components.

SEATS Decomposition

library(seasonal)




```
elec %>%
seas() %>%
autoplot() +
ggtitle("SEATS decomposition of Electricity Production in Australia")
```

STL (Seasonal and Trend decomposition using Loess) Decomposition

- STL can be used with any type of seasonality
- Can be used with a seasonal component that changes
- Smoothness of trend can be controlled

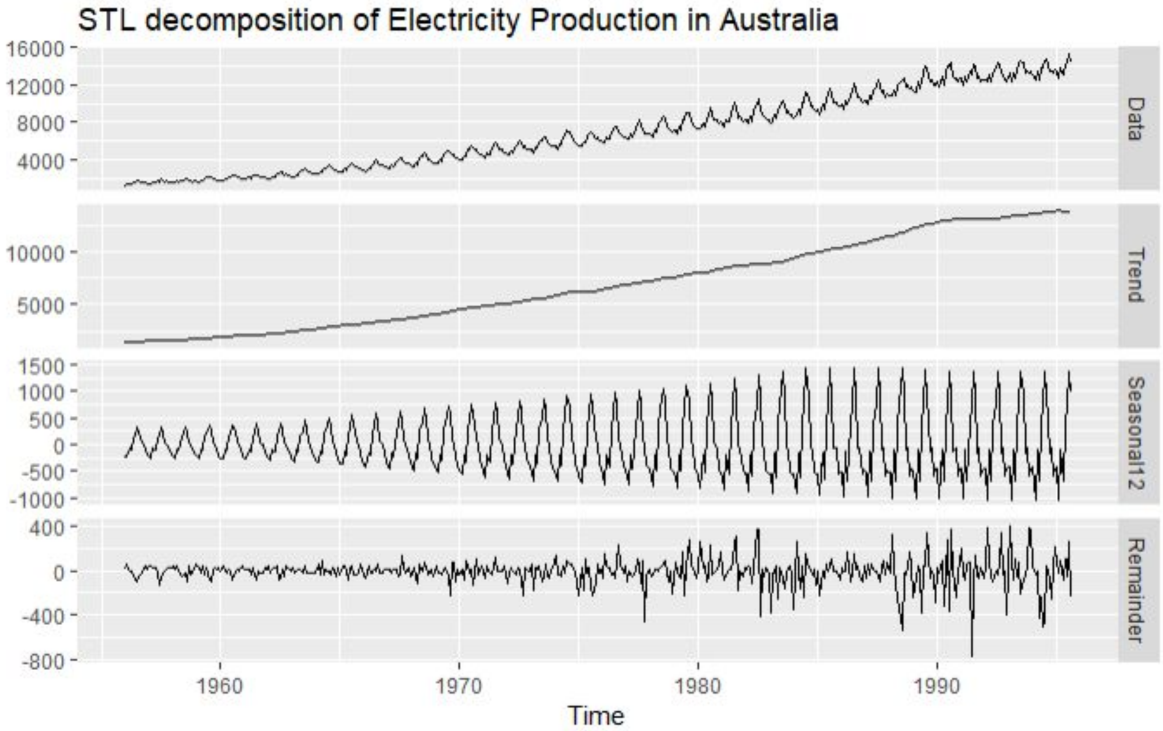
STL Decomposition

library(seasonal)

elec %>% stl(t.window=13, s.window=13, robust=TRUE) %>% autoplot()

s.window - seasonal window number of consecutive years to determine seasonal component

t.window - trend window number of consecutive observations to determine trend-cycle



STL Decomposition

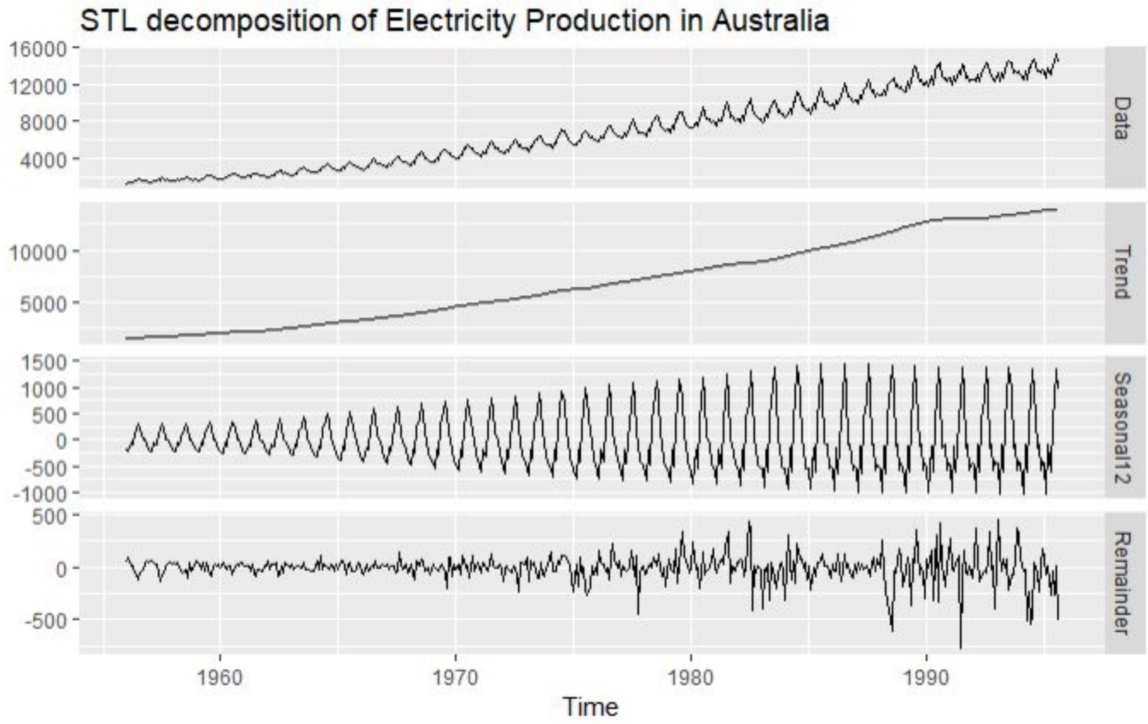
elec %>%

mstl(robust=TRUE) %>%

autoplot() +

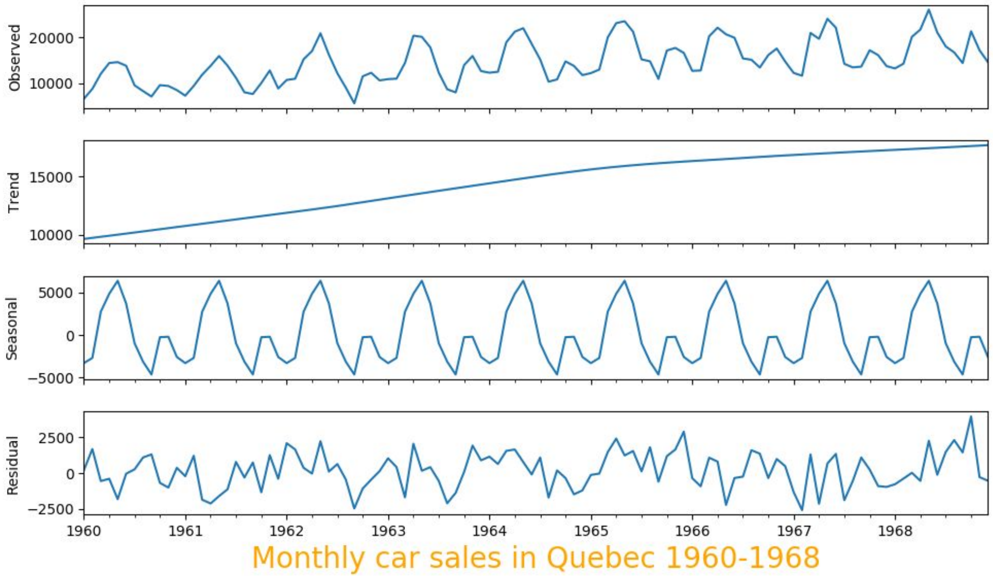
ggtitle("STL decomposition of Electricity Production in Australia")

mstl - chooses t.window and s.window



an STL Decomposition in python

```
import matplotlib.pyplot as plt
import pandas as pd
import csv
import datetime
import stldecompose
from stldecompose import decompose
a_list = []
with open('monthly-car-sales-in-quebec-1960.csv', newline='') as csv_file:
    csv_parsed = csv.reader(csv_file)
    next(csv_parsed, None)
    handler = ((pd.to_datetime(row[0], format="%Y-%m"),
              int(row[1]))
              for row in csv_parsed)
    for row in handler:
        row_df = a_list.append(row)
a_list = pd.DataFrame(a_list, columns=['Month', 'Monthly car sales in Quel
1960-1968'])
a_list = a_list.set_index('Month')   #index has to be promoted out of the data
set to create the right form
stl = decompose(a_list, period=12)
stl.plot()
plt.xlabel('Monthly car sales in Quebec 1960-1968',fontsize=20, color='orange')
plt.show()
```



local linear regression

Fit linear regression model

local lin regression

obs	1	2	3	4	5	6	7	8	9	…	20
Y_i				11.9	16.4	15.4	15.8				
X_i				9.2	10.2	10.8	12.0				

Using a span of .2

When fitting a local regression at $X=10$

you use the weighting function

$$K_i = \frac{2-|10-X_i|}{2}$$

You estimated $\hat{\beta}_1 = 2.489$

What is the estimate for $\hat{\beta}_0$?

span is .2 $\longrightarrow$ take 4 points closest to 10

We want to minimize the residual sum of squares

$$RSS = \sum K_i (Y_i - \beta_0 - \beta_1 X_i)^2$$

$$d/d\beta_0 \, RSS = 2 \sum K_i (Y_i - \beta_0 - \beta_1 X_i)(-1)$$

$$0 = \sum K_i (Y_i - \beta_0 - \beta_1 X_i)$$

$$= \sum K_i Y_i - \beta_0 \sum K_i - \beta_1 \sum K_i X_i$$

$$\hat{\beta}_0 = \frac{\sum K_i Y_i}{\sum K_i} - \hat{\beta}_1 \frac{\sum K_i X_i}{\sum K_i}$$

$$= \frac{.6(11.9) + .9(16.4) + .6(15.4)}{2.1} - \hat{\beta}_1 \frac{(.6)(9.2) + .9(10.2) + .6(10.8)}{2.1}$$

$$\hat{\beta}_0 = 14.83 - 10.09(\hat{\beta}_1)$$

$$= -10.28$$

loess smoothing

- kernel is fit in each local area
- the result is a less noisy series
- different from kernel smoothing
    - outliers are filtered before fitting

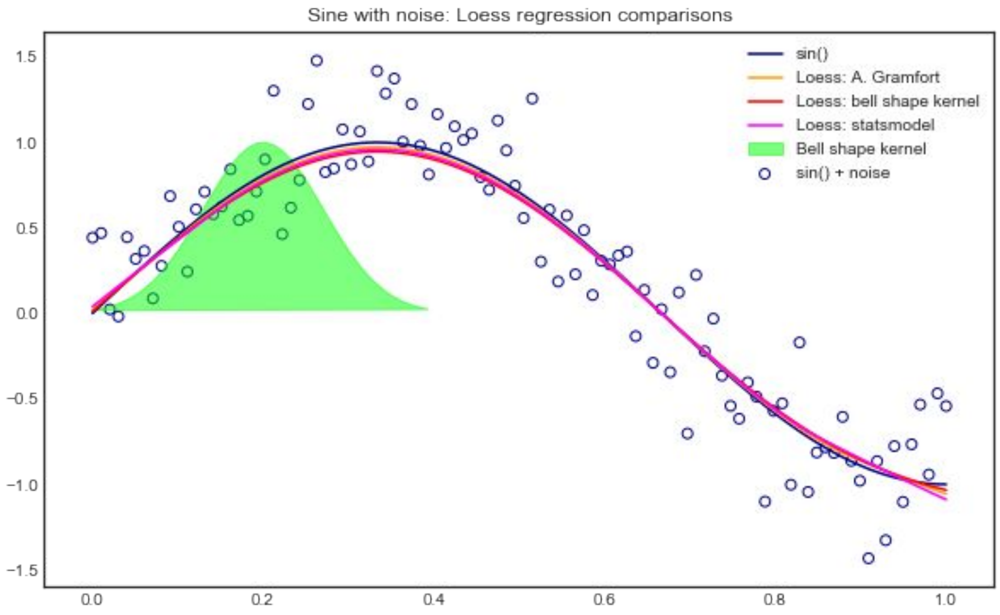


image: Sicotte

Berlin procedure – bv4

- reference intervals are chosen
- trend function and seasonal function are estimated in turn
- interval lengths are different for trend and seasonal
- asymmetric filters are used to get the right fit; trend uses:
    - weighted least squares
    - triangular function in middle
    - degree 3 polynomial used at ends to improve stability

The trend component is approximated by a polynomial of order 3: $\hat{T}_t = \sum_{j=0}^{3} a_j t^j$

The seasonal component is approximated by eleven trigonometric functions

$$\hat{s}_t = \sum_{i=1}^{5} (b_i \cos \lambda_i t + c_i \sin \lambda_i t) + b_6 \cos \lambda_6 t$$

Berlin procedure – bv4

- transfer function used as a filter differs
  - trend component uses a filter that puts a lot of weight where higher frequencies are
  - seasonal component uses a filter that puts the weight at a series of places and little in between

The trend component is approximated by a polynomial of order 3: $\hat{T}_t = \sum_{j=0}^{3} a_j t^j$

The seasonal component is approximated by eleven trigonometric functions

$$\hat{s}_t = \sum_{i=1}^{5} (b_i \cos \lambda_i t + c_i \sin \lambda_i t) + b_6 \cos \lambda_6 t$$

Fischer, p. 7

a seasonal adjustment procedure

Milk production for 2 yrs

2017	713	667	762	784	837	817	767	722	681	687	660	698
2018	734	690	785	805	871	845	801	764	725	723	690	734

the model fit is $X_t = S_t + Z_t$

X_t is the observed series, m_t is the trend, S_t is the seasonal effect and Z_t is the error term

The trend at t is estimated using the centered moving average

Find the seasonally adjusted data for Q2, 2018

By quarter:

Q	2017	2018
1	2142	2209
2	2438	2521
3	2170	2240
4	2045	2147

Centered moving averages

Q	2017	2018
	–	2251.25
	–	2279.0
	2207.125	–
	2225.875	–

for example Q3 2017 → $\hat{m}_3 = \frac{1}{8}(2142) + \frac{1}{4}(2438 + 2170 + 2045) + \frac{1}{8}(2209)$

the quarterly additive effects are:

Q	2017	2018
1	–	−42.25
2	–	242
3	−37.125	–
4	−180.875	–

2209 − 2251.25

actual is this much less than the moving average

ave Q effect

ave Q effect		
−42.25		−37.6875
242	adjust so mean = 0 →	246.5625
−37.125		−32.5625
−180.875		−176.3125
		ave = 0

Q2 (2018), seasonally adjusted is 2521 − 246.5625 = 2274.4375

adjust by taking away the effect

Citations

Coaching Actuaries. ADAPT for MAS-1

Dagum, E. (2010). Time series modeling and decomposition. *Statistica, 70*(4), 433-457.

Deppa, Brant. http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Time_Series_Decomposition.html#stl-decomposition

Fischer, Bjorn. *Decomposition of Time Series: Comparing Different Methods in Theory and Practice*

Hyndman, Rob J., and George Athanasopoulos. *Forecasting: Principles and Practice ;* Otexts, 2018.

Hyndman, R.J. “Time Series Data Library”, https://datamarket.com/data/list/?q=provider:tsdl - monthly car cales in Quebec, accessed on Feb. 2, 2019

Mélard, Guy. “On Some Remarks about SEATS Signal Extraction.” *SERIEs*, vol. 7, no. 1, 2015, pp. 53–98., doi:10.1007/s13209-015-0137-y.

Sicotte, Xavier Bourret. https://xavierbourretsicotte.github.io/loess.html