Course Outline: Time Series Data in R

Harrison Brown

2022-05-06

# Welcome

Welcome to the course outline for *Time Series Data in R*! This course offers methods and workflows for analyzing and interpreting time series data, an overview of when, why, and how to use time series data, and various utilities and packages in R that are beneficial to analysts.

By the end of this course, students will have the skills to:

* Interpret and understand time series plots
* Import ts data to create and manipulate ts objects from the stats package
* Understand why time series data is fundamentally different than non-ts data.
* Analyze time series data with plots
* ?Intro to Wavelet analysis?

# Introduction to time series data

## What is Time Series Data

* Sampled at equi-spaced points in time

## Stationary vs Non-Stationary series

Non-stationary time series are defined by:

* Time-dependent Mean
* Time-dependent Variance
* Time-dependent Autocorrelation/Covariance

## Dickey-Fuller Test of Stationarity

# Creating and Manipulating Time Series

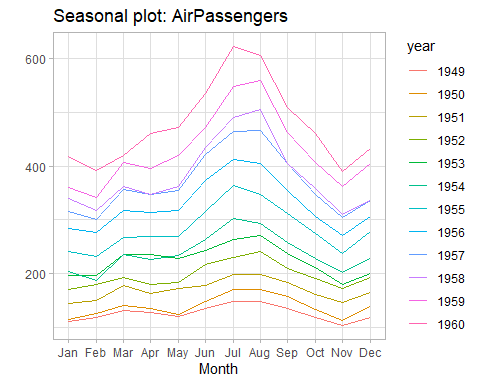
## ts Class

## Creating a ts.plot()

### Interpreting Plots

### Seasonality Plot

ggseasonplot(x = AirPassengers)



### Polar Seasonality Plot

## Trends and Seasons

### Decomposition

### De-trending Data

# Rolling and Expanding Windows

## Rolling Window

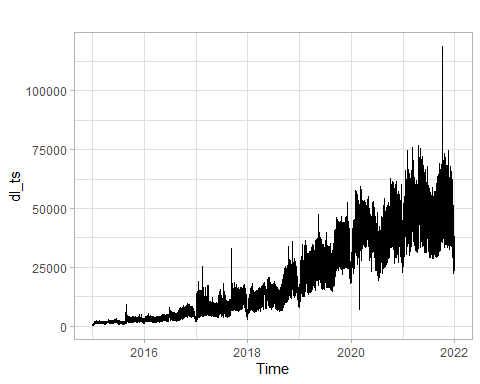
* Moving lower and upper bound

### Data

library(tsbox)  
  
dl\_dplyr <- cran\_data %>%  
 # filter(date >= as.Date("2018-01-01")) %>%   
 select(-package)  
  
# tsbox::ts\_ts() parses the Date column in a much easier way, making the  
# conversion process easy to interpret  
dl\_ts <- dl\_dplyr %>%  
 tsbox::ts\_ts()

## [time]: 'date' [value]: 'count'

autoplot(dl\_ts)



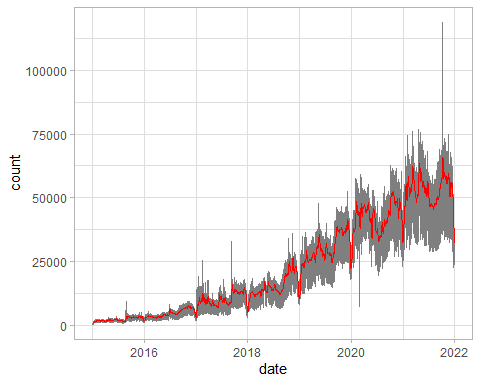
### Calculating a Rolling Window

dl\_dplyr\_rolling <- dl\_dplyr %>%  
 tq\_mutate(  
 select = count,  
 mutate\_fun = rollapplyr,  
 FUN = mean,  
 width = 7,  
 col\_rename = "weekly\_avg"  
 ) %>%  
 tq\_mutate(  
 select = count,  
 mutate\_fun = rollapplyr,  
 FUN = mean,  
 width = 30,  
 col\_rename = "monthly\_avg"  
 )  
  
weekly\_ts <- dl\_dplyr\_rolling %>%  
 select(date, weekly\_avg) %>%  
 ts\_ts()

## [time]: 'date' [value]: 'weekly\_avg'

ggplot() +  
 geom\_line(data = dl\_dplyr, mapping = aes(x = date, y = count), color = "grey50") +  
 geom\_line(data = dl\_dplyr\_rolling, mapping = aes(x = date, y = weekly\_avg), color = "red")

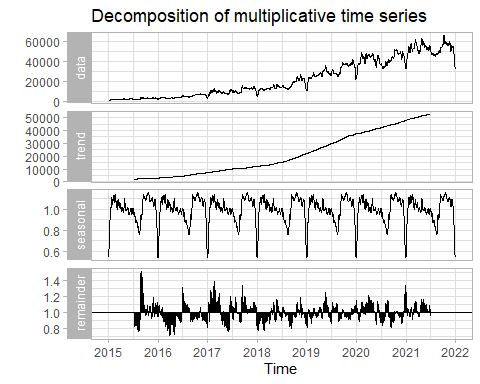
## Warning: Removed 6 row(s) containing missing values  
## (geom\_path).



dl\_rolling\_ts <- dl\_dplyr\_rolling %>%  
 select(  
 date, weekly\_avg  
 ) %>%  
 ts\_ts()

## [time]: 'date' [value]: 'weekly\_avg'

dl\_rolling\_ts %>%  
 decompose(type = "multiplicative") %>%  
 autoplot()



# Expanding Window

* Fixed lower bound
* Moving upper bound

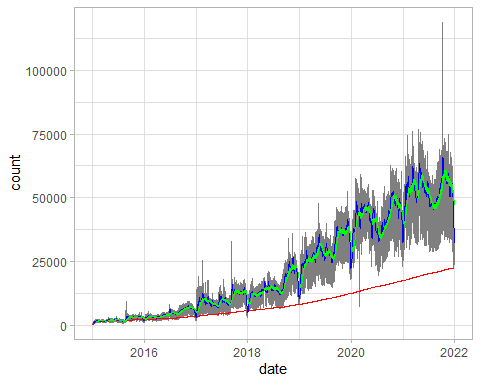
dl\_expand <- dl\_dplyr %>%   
 mutate(  
 mean\_expand = cummean(count),  
 cum\_sum = cumsum(count)  
 )  
  
expand\_ts <- dl\_expand %>%   
 select(date, mean\_expand) %>%   
 ts\_ts()

## [time]: 'date' [value]: 'mean\_expand'

ggplot() +  
 geom\_line(data = dl\_dplyr, aes(x = date, y = count), color = "gray50") +  
 geom\_line(data = dl\_expand, aes(x = date, y = mean\_expand), color = "red") +   
 geom\_line(data = dl\_dplyr\_rolling, aes(x = date, y = weekly\_avg), color = "blue") +   
 geom\_line(data = dl\_dplyr\_rolling, aes(x = date, y = monthly\_avg), color = "green") +   
 theme\_light()

## Warning: Removed 6 row(s) containing missing values  
## (geom\_path).

## Warning: Removed 29 row(s) containing missing values  
## (geom\_path).



# Introduction to Forecasting in R

## Methods for Forecasting

### Exponential Smoothing

# Final Exercise

The final exercise for this course involves performing a time series analysis on real-world sales data. You’ll go step-by-step from reading the data and checking attributes like stationarity, to normalizing, decomposing, adjusting, and interpreting the results.

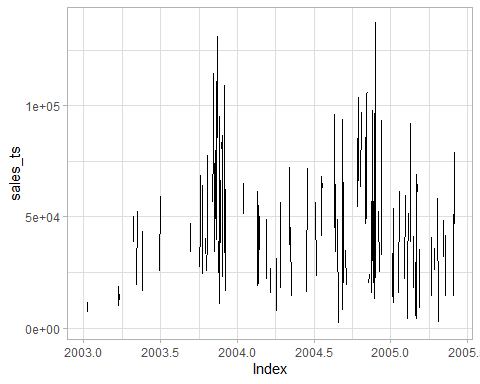
## Importing the Data

## Visual Checks

sales\_ts <- tsbox::ts\_ts(sales)

## [time]: 'date' [value]: 'sales'

autoplot.zoo(sales\_ts)



### Quarterly summary