Manipulating Time Series in R

This course will introduce learners to working with time series data in R. Learners will explore how to **store and format data in date and time objects** as well as how to manipulate time series datasets through subsetting, indexing, and extraction. Examples of time series data across a variety of fields in business and science should be discussed. The course will cover summarization, frequency, missing data, resampling, and comparison techniques as well as window functions for both rolling and expanding windows.

Course Outline:

## Chapter 1: Introduction to Time Series Data

* *Lesson 1.1: What is Time Series Data*
  + Learning Objective: Learner will be able to understand why TS-data is needed in real-world analyses, how TS-data differs from non-temporal data, as well as what can be gained by performing TS analysis
  + LO: Learners will be able to manipulate date-time data and difference between classes of date-data in R (numeric, Date, POSIXct, etc.)
    - e.g.: as.Date()
* *Lesson 1.2: How to Interpret Time Series Data*
  + LO: Learner will understand how to interpret a basic time series plot
  + – Learner will understand the difference between stationary and non-stationary data, and how the following differ between the two:
    - Mean,
    - Variance,
    - Covariance/Autocorrelation
* *Lesson 1.3: Components of Time Series Data*
  + – Learner will be able to understand the three primary components of TS-data: Trend, Seasonality, and Random components
  + – Learners will learn the difference between an “additive” and “multiplicative” model
  + – Learner will be able to interpret a “season plot”
* *Lesson 1.4: Creating a time series with* ***ts()***
  + LO: Learners will be able to read in data from a dataframe or tibble, and convert it into an object of class “ts” with the ts() function
  + – Learners will be able to understand what properties a *ts* object has, including:
    - Equi-spaced observations in time (i.e., equal, regular sampling intervals)
    - Attributes like frequency, start, and end.

## Chapter 2: Manipulating Time Series Data

* *Lesson 2.1: Subsetting, Indexing, and Extraction*
  + LO: Take subsets and windows of time series, both before and after creating the “ts” object:
    - Functions include dplyr::filter(), tsbox::ts\_ts(), and window()
* *Lesson 2.2: Quantifying a Time Series*
  + LO: Learners will be able to perform the Dickey-Fuller test to determine if a time series is stationary or non-stationary
    - tseries::adf.test()
  + LO: Learners will learn about the *print* and *plot* methods in R
    - forecast::autoplot
* *Lesson 2.3: Decomposing a Time Series*
  + LO: Learners will be able to decompose a time series and be able to plot and interpret the results.
    - stats::decompose()
  + – Learners will be able to adjust for the seasonality in the data
    - forecast::seasadj()
* *Lesson 2.4: Missing Values and Resampling*
  + LO: Why NA and missing values “mess up” the workflow
  + LO: Several imputation algorithms from the imputeTS package:
    - na\_mean, na\_interpolation, etc.

## Chapter 3: Rolling Windows

* *Lesson 3.1: What are Windows?*
  + LO: Understanding the importance of using windows, especially as the size of the data increases.
  + Why not just calculate the mean of the whole sample?
  + Several visualizations for difference b/w global mean and rolling mean
* *Lesson 3.2: Calculating a Rolling Window*
  + LO: Calculate a rolling window to find 7-day and 30-day (i.e., any arbitrary window) averages
    - zoo::rollapply()
* *Lesson 3.3: Interpreting the Results*
  + LO: Visualize smoothed data compared to original, looking at different levels of smoothing.
  + LO: Decompose rolled average data to see how smoothing affects (positively or negatively) the data
* *Lesson 3.4: Calculating an Expanding Window*
  + LO: What is an expanding window?
    - zoo::rollapply(), dplyr::cummean(), etc.
  + Visualize expanding mean compared to rolling mean & original data

## Chapter 4: Introduction to Forecasting

* *Lesson 4.1: What is Forecasting*
  + LO: Why businesses perform forecasting
  + Applications and Examples, e.g. predicting monthly sales based on previous years
  + LO: Recap of Confidence Intervals for non-stats learners
* *Lesson 4.2: Naïve Forecasting*
  + LO: Why naïve forecasting doesn’t always perform well on complex models
    - forecast::naive()
* *Lesson 4.3: Exponential Smoothing*
  + Perform simple exponential smoothing with:
    - forecast::ses()
* *Lesson 4.4: Introduction to Advanced Forecasting Models*
  + LO: Explanation of ARIMA
  + Perform basic ARIMA with
    - forecast::auto.arima()
  + Note that we shouldn’t always rely on automated models without understanding the foundational statistics of what we’re doing.