Time Series and Forecasting: A Project-based Approach with R

Arthur Small

Version of: 2022-02-21

Contents

This document is a compilation of class notes for SYS 5581 *Time Series and Forecasting*, University of Virginia, Spring, 2021.

Preface

This document contains class notes and other materials related to SYS 5581 *Time Series and Forecasting* at the University of Virginia.

Readings and references

Time series

FPP3 = Hyndman, R.J., & Athanasopoulos, G. (**?**) *Forecasting: principles and practice, 3rd edition*, OTexts: Melbourne, Australia. OTexts.com/fpp3

TFS = [these notes]

Statistics with R

Data science with R, general

R4DS = Wickham, Hadley, and Garrett Grolemund, *R for Data Science*

TSDS = Carrie Wright, Shannon E. Ellis, Stephanie C. Hicks and Roger D. Peng, *Tidyverse Skills for Data Science*

Tibshirani, Ryan, Statistics 36-350 *Statistical Computing*, Carnegie-Mellon University, Fall 2019

Other references on Zotero

A variety of other references to resources on time series and forecasting are gathered in the Zotero library for this course.

Acknowledgements

These notes are organized using the **bookdown** package (**?**), which was built on top of R Markdown and **knitr** (**?**).

Course Syllabus

title: “SYS 5581 Time Series and Forecasting” author: “Instructor: Arthur Small” date: “University of Virginia Engineering, Spring 2021”

Class meetings: MW 09:30-10:45 a.m. online via Zoom

Office Hours: MW 11:00 a.m.-12:30 p.m. online via Zoom (subject to change). Sign up in advance for a 45-minute session via the Collab “Sign Up” tool. If you cannot make any scheduled time, please contact the instructor via email to schedule an appointment. Meetings online via Zoom: https://virginia.zoom.us/my/arthursmalliii

Web Resources:

- Collab class site, for basic course information, assignments, office hours sign-up, links to online textbook and other resources.
- Github class site, for posting and sharing code.
- Zoom, for class sessions, recordings, and office hours.

Course Description

The course is designed to introduce graduate students and advanced undergraduates in engineering to time series and forecasting. The course will not include a deep exploration of theory. Rather, the goal is for students by the end to be able to analyze time series data competently, as part of their work designing and working with engineered systems.

In addition to learning theory, each student will undertake a semester-long research project. Ideally, this project will relate closely to the student’s own dissertation research, professional practice, or other domain application that interests them. My hope is that these projects could form the basis for subsequent research papers, dissertation chapters, or other professional work products, for interested students.

The course will, therefore, be structured primarily as a *workshop*: the ultimate goal is to help you to create a professionally presented report. Our workflow will, therefore, be subject to revision, according to my judgement of how best to use our time to help you produce a professional report.

The course outline is divided into two major sections. First, we will introduce the theory, with examples. In the later part of the semester, we will focus on workshopping your projects in progress.

Important: class readings are subject to change, contingent on mitigating circumstances and the progress we make as a class. Students are encouraged to attend lectures and check the course website for updates.

Prerequisites

Students should have taken at least one rigorous intermediate course in probability and statistics. They should be comfortable with the representation of uncertain information in the form of probability distributions, with conditional probabilities, and with other such foundational concepts.

In addition, to carry out the data analyis, the student should have at least ability be able to code, in some general-purpose language. For this course, we will work in R, focusing on specialized packages for working with time series data and generating forecasts.

Expectations

Each student will make a presentation on their data analysis project. Students will be evaluated based on their performance in these presentations and on their final project, on occasional short quizzes; and on their contributions inside and outside of class towards helping other students.

Readings

The primary text for the course will be *Forecasting: Principles and Practice*, 3rd ed. by Rob J. Hyndman and George Athanasopoulos. This resource is available for free online and is linked from the Collab site. The text includes example code in R, and covers several useful R packages related to time series and forecasting.

Additional readings including relevant articles will be provided as the course progresses. The choice of readings will depend in part on student interests, as conveyed through their choice of projects.

Course Objectives

1. Students will learn the foundations of time series and forecasting.

2. Students will gain the experience of building statistical models of time series, and models for forecasting, and will learn how to evaluate their performance.

3. Students will learn the concepts and practice of *reproducible research*, in the course of preparing a research paper.

4. Students will gain experience in making presentations and in preparing a polished research article.

Grading Policy

- **10%** of your grade will be determined by quizzes designed to test your understanding of the theoretical concepts introduced in class. This quiz will delivered at roughly the mid-point of the semester. It will be open-book and open-notes, outside of class. You will have multiple days to complete it. The quiz will not be designed to be especially challenging: the goal is to give you the opportunity to synthesize your understanding of core concepts, in preparation for developing your data analysis for your research project.

- **10%** of your grade will be determined by your performance in one in-class presentation based on your project. These presentations will be scheduled when you are, in the judgement of the instructor, far enough along to do so.

- **10%** of your grade will be determined by your contributions to assist other students. These contributions can come through class participation, by making useful contributions in online forums (Github), or through other means that add value to the group experience.

- **70%** of your grade will be determined by your performance on your final project. The development of the project will include multiple iterations, each with an associated deliverable:

  - An initial Concept Note.
  - A more developed Project Proposal.
  - A first working draft of your final project paper.
  - A final complete draft of your project paper.

Details of these staged intermediate deliverables will be forthcoming. The final product should be a polished professional paper that meets academic standards regarding format, quality, and integrity.

Attendance Policy

Regular attendance is very much in your pedagogic interest. However, it is up to you whether to attend in person or to view recorded class sessions afterwards.

Communications protocols, including emails and office hours

I prefer to avoid using email to communicate with students about class matters. For substantive questions about course materials and concepts, please use class time, office hours, or meetings by appointment. Please use email only for brief clarifying questions, or to set up appointments.

Academic Dishonesty Policy

Don’t cheat. Don’t plagiarize. Don’t present someone else’s work as your own.

Disabilities Policy

Together with the University of Virginia, I am committed to assuring that all students have the full opportunity to benefit from the course regardless of their disability status. If you have a disability that may require accommodations, please see the instructor early in the semester to work out appropriate arrangements.

Part I

Introduction and Overview

Chapter 1

Introduction

Readings: FPP3, Ch. 1

1.1 What is time series analysis?

1.2 Time series data

1.3 Time series patterns

Examples.

1.4 Types of problems that are amenable to time series analysis

1.5 Time series and forecasting

1.6 Overview of the course

Chapter 2

Project workflow

Readings:

- FPP3, Section 5.1
- TSDS, Sections 5.1-5.3

Case study examples:

- Open Case Studies: Exploring CO2 emissions across time
- Open Case Studies: Predicting Annual Air Pollution

Steps in a time series statistical analysis:

1. State your question
2. Acquire data and background information
3. Organize your data
4. Perform exploratory analysis of your data
5. Write down your model of the data generating process
6. If necessary: transform the model and data to make it ready for analysis
7. Choose an appropriate technique for estimating model parameters, consistent with your assumptions about your data generating process
8. Estimate model parameters
9. Confirm that your modeling assumptions are satisfied;
10. Compute measures of model quality; confirm that your model is good enough for your purpose
11. Use your calibrated model to address your original question.

Examples of using your model:

- Generate a forecast of future events
- Estimate the probability of a future event
- ...

Example problem: Estimating the probability of a weather event

A pub owner in Charlottesville plans to sell beer outside on St. Patrick's Day, March 17. The pub owner must decide whether to arrange to rent a supplemental refrigeration system for the day.

Supplemental refrigeration offers a form of insurance. If temperatures outside on March 17 are high and the pub owner has not arranged supplemental refrigeration, she will be left with warm beer that she will have difficulty selling, leading to financial losses. Conversely, if she pays for supplemental refrigeration when temperatures are low, she will have incurred an unnecessary expense.

To decide whether to insure herself against loss, she wishes to estimate the probability distribution of temperatures on March 17.

2.1 State your question

1. What is the probability that the high temperature on March 17 will exceed 23 degrees Celsius?

2.2 Acquire data and background information

Get historical temperature data.

(Here, will simulate the data.)

```
set.seed(1)       # Keeps random data from changing each time the code is run

theta <-  18      # True long-run average temp
sigma  <-   3     # True stnd dev of temp around this mean
n      <- 40      # n = number of simulated historical data points
                     # We don't use 'T' because in R language 'T' = logical 'TRUE'

y <- rnorm(n,theta,sigma)
```

2.3 Organize your data

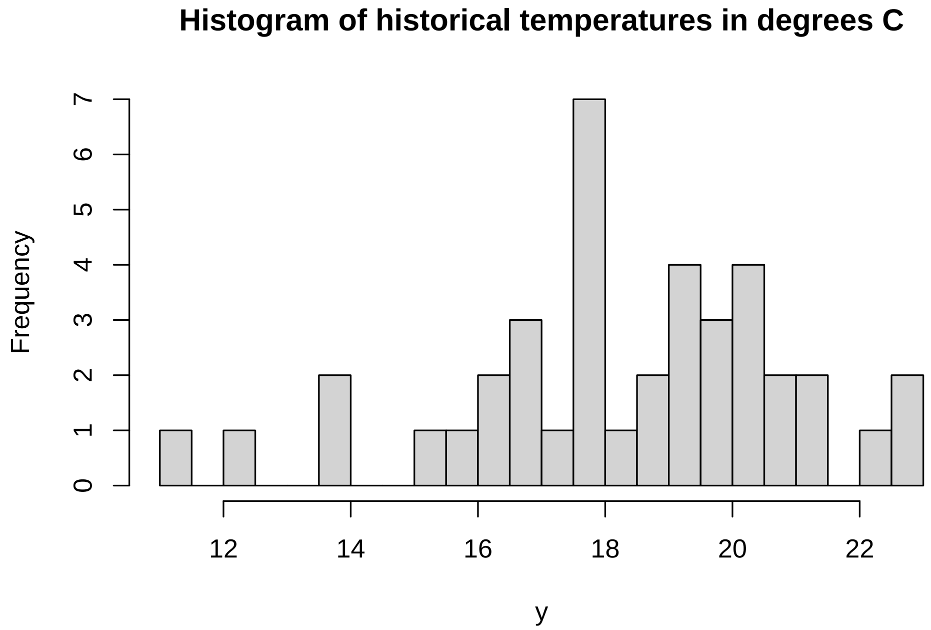
Let $y_1, \ldots, y_T$ denote the high temperature in Charlottesville on March 17 for each of the previous T years.

```
print(y)
```

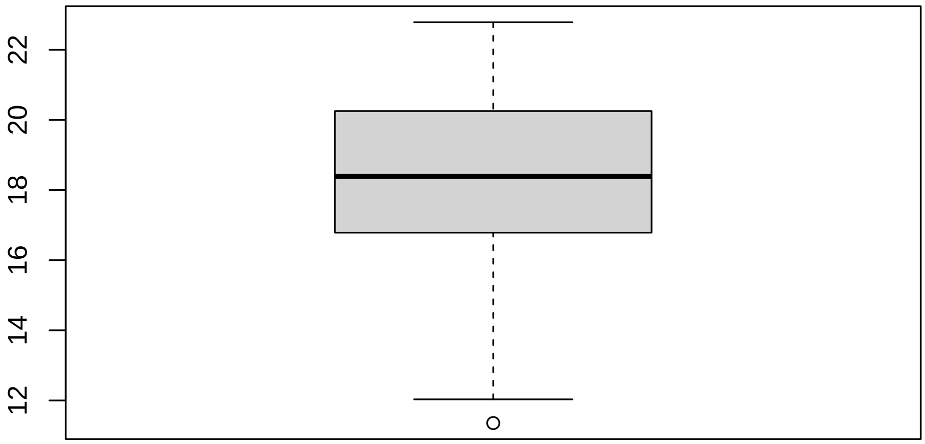
```
##  [1] 16.12064 18.55093 15.49311 22.78584 18.98852 15.53859 19.46229 20.21497
##  [9] 19.72734 17.08383 22.53534 19.16953 16.13628 11.35590 21.37479 17.86520
## [17] 17.95143 20.83151 20.46366 19.78170 20.75693 20.34641 18.22369 12.03194
## [25] 19.85948 17.83161 17.53261 13.58774 16.56555 19.25382 22.07604 17.69164
## [33] 19.16301 17.83858 13.86882 16.75502 16.81713 17.82206 21.30008 20.28953
```

2.4 Perform exploratory analysis of your data

```
hist(y, breaks = 20, main = "Histogram of historical temperatures in degrees C")
```



```
boxplot(y)
```



2.5 Write down your model of the data generating process

It is supposed that these data were generated as independent, identically distributed random draws from a normal distribution: for $t = 1, \dots, T$,

$$y_t = \theta + \varepsilon_t$$

where θ denotes the true but unobserved value of the long-run average temperature, and where $\varepsilon_t \sim N(0, \sigma^2)$.

2.5.1 Comments on this statistical model: The risk of model mis-specification

This model asserts several substantive assumptions about the data generating process.

- The process is assumed to be *stationary*. There is no upward trend over time, no long-term climate change, etc.
- Temperatures are assumed to be *independent* from one year to the next. In particular, there is no *autocorrelation*. Knowing that one year's temperature was unusually high (say) provides no information about the likelihood that next year's temperature will also be unusually high. Inter-annual climate cycles (e.g., due to *El Niño*) are ruled out.

- Temperature variations around the long-run average are assumed to be *identically distributed*. This assumption rules out the possibility that variance is, say, greater when temperatures are higher than when they are lower.

And others.

In general, it is important to formulate a statistical model that accurately reflects the true characteristics of the underlying data generating process.

When your statistical model is mis-specified, your probabilistic forecast of future events are likely to stray from the true underlying probabilities. Model mis-specification can then lead to inaccurate estimates of the distribution of losses for each possible action. This error may in turn lead to selection of a sub-optimal action.

A particular problem to guard against is the possibility to underestimate the likelihood of extreme events that could cause catastrophic losses.

That said, your time on this Earth is limited. Depending on the decision problem and the stakes involved, refining your model to get sharper loss estimates may or may not be worth the bother.

A reasonable approach is to start by first writing down a simple forecasting model that appears to capture the essence of the process as you understand it. On the basis of this simple model, generate first-cut probabilistic forecasts of uncertain events. Use these to generate estimated distributions of losses for each possible action in your action set. On that basis, use the specified decision criterion to derive an initial optimal decision rule.

Then, go back and check things over more carefully. Review the realism of your statistical model, given your understanding of data generating process. Plot and examine the distribution of your prediction errors. Do your prediction errors appear to follow a pattern that matches what you would expect, given the assumptions you have made?

If you find evidence that your prediction model is mis-specified, it may be worth it to go back and refine your model, and see if you generate different results.

One very good idea is to perform a *sensitivity analysis*. How sensitive are your decision recommendations and outcomes to the assumptions you've built into your statistical model? If you are not highly confident in your statistical assumptions, and if those assumptions turn out to matter a lot for your recommendations and outcomes, then it could very well be worth the bother to revisit those assumptions, and investigate alternatives.

On the other hand, if your decision recommendations are not highly sensitive to your statistical assumptions, then keeping your initial model may be defensible. The point of this work is *not* to build the best possible prediction system, bullet-proof against any statistical criticism. The point is to help people make good decisions – or at least, decisiions better than they would have made otherwise. Your time and other resources are limited. A good-enough model may be good enough.

2.6 If necessary: transform the model and data to make it ready for analysis

Not needed here.

Will consider many cases where it is.

2.7 Choose an appropriate technique for estimating model parameters, consistent with your assumptions about your data generating process

For this case, ordinary least squares (OLS) estimation is just fine.

2.8 Estimate model parameters

```
theta_hat <- mean(y)         # Sample mean
epsilon_hat <- y-theta_hat   # Model residuals
ssr <- sum(epsilon_hat^2)    # Sum of squared residuals
sigma_hat <- ssr/(n-1)       # Estimated standard error
print(theta_hat)
```

```
## [1] 18.27608
```

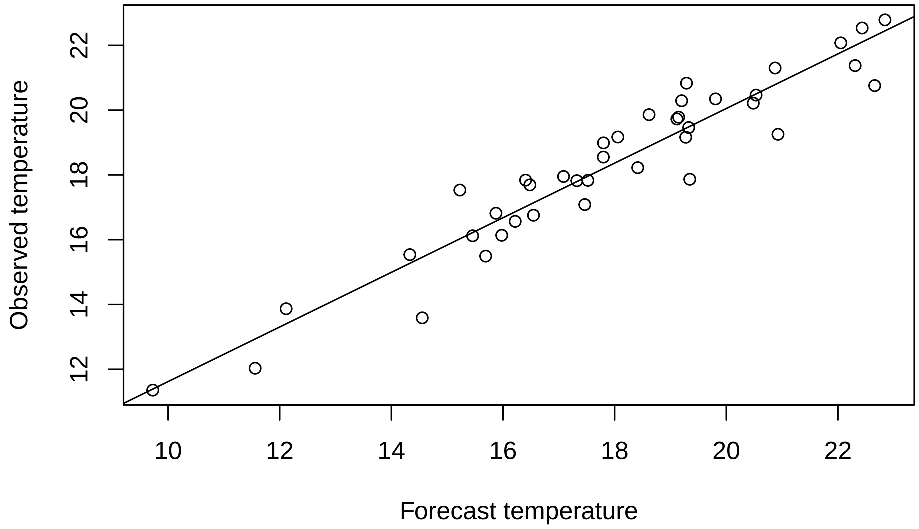
```
print(sigma_hat)
```

```
## [1] 7.075605
```

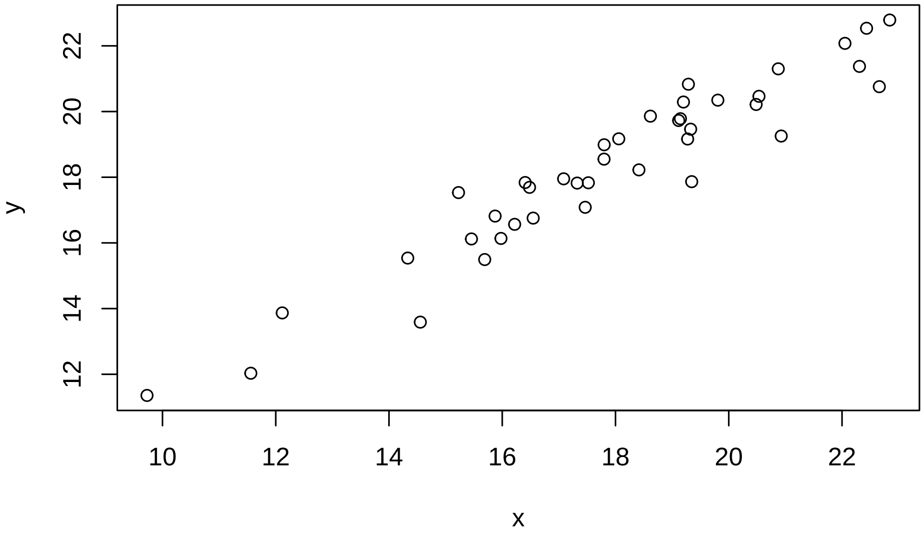
2.9 Confirm that your modeling assumptions are satisfied

2.10 Compute measures of model quality; confirm that your model is good enough for your purpose

```
forecast_bias <- -0.5
forecast_std_error <- 1

forecast_errors <- rnorm(n,mean = forecast_bias,sd = forecast_std_error)

x <- y + forecast_errors

linear_model <- lm(y~x)

plot(x,y, xlab = "Forecast temperature", ylab = "Observed temperature")
abline(linear_model)
```

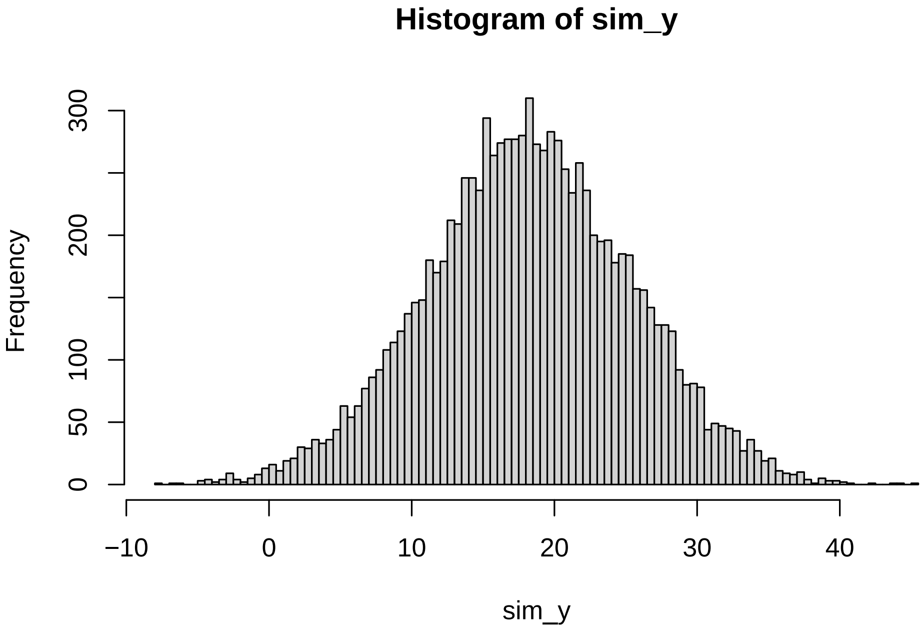


```
plot(x,y)
```



2.11 Use your calibrated model to address your original question

```
sim_y <- rnorm(10000, theta_hat, sigma_hat)

hist(sim_y, breaks = 100)
```


Histogram of sim_y
Frequency
0
50
100
200
300
−10
0
10
20
30
40
sim_y

Chapter 3

Assignment: Write a concept note

Write a concept note for a potential time series analysis project.

3.1 Assignment Instructions

In just a few paragraphs (1 page max), describe an application of time series analysis that you might undertake in connection with your own research, your professional practice, or for a class project.

3.2 Submission procedure

Prepare your note in R Markdown within R Studio. Use `knitr` to generate a PDF output.

Commit your saved .Rmd and .pdf files to your (local) repo, then push your changes to your repo on the course site on Github.

When all is ready, submit your assignment on Collab. But don't attach your pdf file. Instead, a link to your .pdf file on Github.

Generate your output as a PDF rather than as HTML: Github makes is complicated to download and view .html files.

3.3 Choosing a topic

Your topic might depend on which type of degree program you are in, and how far along you are in that program.

If you are well along in a research-oriented graduate degree program, then you likely have already a fairly clear notion of the research question(s) you are addressing or plan to address, and the methods you will use to approach your question. Ideally, you will have already secured access to a time series data set that you hope to analyze. In this case, please describe the data set, and the type of useful information you hope to learn about it. If applicable, describe how this analysis could inform your research program.

If you at the beginning stages of a research-oriented degree program, you likely have some idea of the topics and methods you will use, but may not have much idea of what data sets you will use or what insights you hope to extract from those data. In this case, please think about what kinds of time series data analysis might help advance your work. It is likely that a consultation with your research advisor will be helpful. Write about your research question(s), and how analysis of time series data might help your research program. Ideally, please try to identify a data set that you could use, and describe how you can access these data.

If you are not in a research-oriented degree program, think about how a time series analysis could be applied to some aspect of your work, professional practice, or personal or career interests.

If you don't have any ideas at all, and are looking for help finding one, let me know. We will identify together a project idea and data set you can work with, possibly related to energy. But try first to think of one on your own, one that you care about.

Part II

Set up

Chapter 4

Project set up: Good practices

4.1 Reproducible workflows

4.2 Setting up a new project

4.3 Folder structure

Readings:

TSDS Section 1.6

4.4 Naming things

Readings:

Jenny Bryan, naming things slide deck, Reproducible Science Workshop, 2015.

Hadley Wickham, The tidyverse style guide. Section 1: “Files”

Chapter 5

Assignment: Set up your computing environment

The course relies on computing resources. Please install the software as indicated on your local machine, and familiarize yourself with the associated documentation.

Topics: R, R Studio, git, Github, Markdown, R Markdown, Tidyverse and tidyverts packages for R

Assignment: Follow instructions in the course Computing setup guide.

5.1 The R programming language, and related resources

We will do our coding in R, a programming language especially well-suited to statistical computing.

- Download and install R, v. 3.0.1+.
    - Note: There is a later version, v. 4.0.2, in development, but you shouldn't need it.

R Studio is an integrated development environment (IDE) for R. It offers a variety of utilities to enhance the experience of coding and generating documents.

- Download and install R Studio, v. 1.4.1+.

Tidyverse is a collection of packages that extend the capabilities of R for doing data science.

- Install the Tidyverse packages for R: From the Console tab in R Studio (or from R running in a Terminal window), enter:

```
install.packages("tidyverse")
```

- Alternatively, you may install packages via the `Packages` tab in R Studio.
- Optional: To learn how to wrangle and visualize data using the Tidyverse packages, you may find it useful to go through the Tidyverse Fundamentals with R modules on Datacamp.
  - Datacamp also offers a range of other learning modules for developing data science skills in R.

Tidyverts is a collection of R packages for time series analysis designed to work well with the Tidyverse packages. Each package in the tidyverts suite needs to be installed individually:

- From the Console tab in R Studio (or from R running in a Terminal window), enter:

```
install.packages(c("tsibble", "tsibbledata", "feasts", "fable"))
```

- You don't need to install the `tsibbletalk` and `fable.prophet` packages; we probably won't use them in this course.

5.2 Git and Github

Reference: Happy Git and GitHub for the useR

Git is software for version control. Github is a web service that provides remote storage and access to files via git. This setup greatly facilitates collaboration between multiple individuals working on the same code base.

First watch this short YouTube video to get an orientation to git and Github: Git and GitHub for an Organized Project (STAT 545 Episode 2-A) from the University of British Columbia.

Then install git on your machine and link it to your R Studio instance and your file repository on Github:

- Follow these instructions to download and install git and to link git with R Studio.

A collection of files associated with a single project is in git-speak called a "repository" or *"repo"*. You should already have a basic repo set up for you on the course site on Github. The next step is to copy ("clone") this remote repo to your local machine.

- Clone your course repo on Github to a new R Studio project on your local machine.
  - Navigate to the course website on Github. Select your repo.
  - Click on the green button labeled "Code". Copy the URL.
  - In the R Studio window, from the pull-down menu in the upper-right corner, select `New Project...`, `Version Control`, `Git`. Paste the URL into the dialog box labeled `Repository URL`.
  - Optional: Change the name of the project folder, and the location of this folder on your local directory tree.
  - Click on `Create Project`. The files from your remote repo should be copied to your local machine in a new folder with the name you chose.
- Optional: Download and install the Github desktop client, or an alternative GUI client.
  - The git operations you need for this course can be managed within R Studio, from the `Git` tab. Some more advanced operations require using either a Terminal window, or a Git desktop client.

As you get going, you will likely want to learn more about how to work with git and Github. Review the documentation for git and this Github Guide. Learn the basics.

5.2.1 Using personal tokens to access Github

Github is phasing out the use of passwords for authorizations.

```
---- Forwarded Message -----
From: GitHub <noreply@github.com>
To: Arthur Small <asmall@virginia.edu>
Sent: Sunday, February 21, 2021, 6:20:58 AM EST
Subject: [GitHub] Deprecation Notice

Hi @arthursmalliii,
You recently used a password to access the repository at uva-eng-time-series-sp21/coron

Basic authentication using a password to Git is deprecated and will soon no longer work

Thanks,
The GitHub Team
```

Instead, you must create a personal access token. See the Github documentation.

5.3 Markdown and R Markdown

Markdown is a markup language: a set of formatting instructions for rendering documents. R Markdown is an extension of Markdown that allows for embedding chunks of R code into a Markdown document. In this course, we will write our work in R Markdown within the R Studio environment, then use the `knitr` package to generate HTML and PDF output files.

For a nice introduction to Markdown and R Markdown, watch the short YouTube video Reproducible Reports with R Markdown (STAT 545 Episode 3-A) from the University of British Columbia.

As you proceed in creating your documents, you will probably want to access additional resources:

- From within R Studio, you can access an R Markdown Cheat Sheet via `Help/Cheatsheets`.
- Markdown reference: https://www.markdownguide.org/
- R Markdown reference: https://rmarkdown.rstudio.com/

5.4 Bibliographic resources: Zotero and Bibtex

[Coming soon…]

5.5 General course web resources

- Collab class site, for basic course information, assignments, office hours sign-up, links to online textbook and other resources.
- Github class site, for posting and sharing code.
- Zoom, for class sessions, recordings, and office hours.

Part III

Data Acquisition and Preparation