Unit 6 Live Session

Time Series Analysis Lecture 1

Figure 1: South Hall

Class Announcements

- Congratulations on finishing the first part of the course!
- HW 6 is out this week
- Lab-2 is due in 5 weeks

Roadmap

Rear view Mirror

- How to model different types of cross-sectional data
- How to conduct a thorough statistical analysis for binary, unordered multiclass, ordered multiclass, and count data

Today

- Introduction to time series analysis
- Basic terminology of time series analysis
- Notion and measure of dependency
- Notion of stationarity and Ergodic theorem
- Examples of simple time series models

Looking Ahead

- Linear time-trend regression
- Time-series smoothing techniques

Start-up Code

```
### Load a set of packages inclusing: broom, cli, crayon, dbplyr , dplyr, dtplyr, forcats,
## googledrive, googlesheets4, ggplot2, haven, hms, httr, jsonlite, lubridate , magrittr,
## modelr, pillar, purrr, readr, readxl, reprex, rlang, rstudioapi, rvest, stringr, tibble,
## tidyr, xml2
library(tidyverse)

# Insert the function to *tidy up* the code when they are printed out
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)

# to generate random walk plot
library(simts)
```

Introduction

Fundamental concepts in Time-series Analysis

Please define the following concepts:

a)- Stochastic process

- What is a deterministic process?

b)- Time series

c)- Realization of a random process

c)- Sample path

Fundamental Properties of Time Series

Expectation and Variance

Suppose we have a time series $Y_1, .., Y_T$

- Expectation of a time series is given by:

$$E(Y_t) = \int_{-\infty}^{\infty} y_t f_{Y_t}(y_t) dy_t$$

- The variance of a times series with a stationary mean is defined as:

$$\sigma^2 = E(Y_t - \mu)^2 = \int_{-\infty}^{\infty} (y_t - \mu)^2 f_{Y_t}(y_t) dy_t$$

a)- Compute the expectation of the following time series at (t-1) and (t).

$$X_t = 0.2 + W_t$$
$$Z_t = 0.5 \cdot t + W_t$$

- Where $\{W\}_{t=-\infty}^{\infty}$ is a white noise process with $E(W_t) = 0$ and $E(W_t^2) = \sigma^2$.

b)- Do the time series have a stationary mean?

c)- Compute the variance of X_t and Z_t at t.

The Autocorrelation and Autocovariance Functions

- The autocovariance function of a covariance stationary time series model Y_t is defined as:

$$\gamma_k = Cov(Y_t, Y_{t-k}) = E(Y_t - \mu)(Y_{t-k} - \mu)$$

- The autocorrelation function is a useful measure of how long effects persist in a time series and is defined as:

$$\rho_k = Corr(Y_{t-k}, Y_t) = \frac{\gamma_k}{\gamma_0}$$

a)- What does $\gamma(k) = 0$ imply?

b)- Compute the γ_1 of X_t and Z_t between t and $t-1$.

c)- Compute the ρ_1 of X_t and Z_t between t and $t-1$.

d)- What is the partial autocorrelation function? How is it different from the autocorrelation function?

Stationarity

- Recall there are two main types of stationarity that are commonly used:

  1- **Strict Stationarity**: The stochastic process is said to be strictly stationary if for every set of time indices $1 \leq t_1 \leq ..... \leq t_k$, the joint distribution of $Y_{t_1}, Y_{t_2}, ..., Y_{t_k}$ is the same as the joint distribution of $Y_{t_1+h}, Y_{t_1+h}, ..., Y_{t_k+h}$.

  2- **Covariance(weak) Stationarity**: A stochastic process is said to be weakly stationary if it has a constant mean and variance, and its covariance function $\gamma(Y_t, Y_{t+h})$ depends only on h (the time difference) and not on t (time itself).

a)- Are X_t and Z_t stationary? Why or why not?

Ergodic theorem and Estimation

- If Y_t is a stationary and ergodic process with $E(Y_t) = \mu$, then:

$$\bar{y} \equiv \frac{1}{T}\sum_{t=1}^{T} y_t \underset{p}{\rightarrow} \mu$$

a)- What's the intuition behind the Ergodic theorem?

b)- What is an example of an ergodic process?

c)- Is the Ergodic theorem related to any other important theorem you learned in w203?

Sample Autocovariance Functions

- A natural estimator of the autocovariance function is given by its sample analogue:

$$c_k = \frac{1}{T}\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t-k} - \bar{y})$$

- And the estimator of the autocorrelation function is defined as:

$$r_k = \frac{c_k}{c_0}$$

For a time series, we can plot the sample autocorrelation and sample partial autocorrelation using the acf and pacf functions in R.

a)- Randomly draw 1000 observations from X_t and Z_t and plot their realizations.

b)- What do you notice about X_t and Z_t in their realization plots and also in the acf and pacf?

```
# Replace with your code
```

Stochastic Models

White noise

- A process W_t is white noise if:
  - $E(W_t) = 0$
  - $Var(W_t) = \sigma_2$
  - $cov(W_t, W_{t-k}) = 0$ for $k \neq 0$.
- If W_t is normally distributed, then it's a Gaussian white noise process:
- The white noise model is the building block for most time series models.

$$W_t \overset{iid}{\sim} N(0, \sigma_w^2)$$

a)- Write a function to take 100 draws from a Gaussian white noise with $\mu = 0$ and $\sigma^2 = 1$. Then use following code to plot 100 simulations of the white noise process.

```
w <- function(n= 100) {
   return(1) # Replace with your code
 }
### After you write the function uncomment and run the code to produce the plot
# data <- data.frame(t = seq(from = 1, to = 100, by = 1), replicate(w(), n = 100))
#
# data <- data %>%  pivot_longer(
#   cols = starts_with("x"),
#   names_to = "x",
#   values_to = "value"
# )
#
#
# ggplot(data, aes(x = t, y = value, col = x)) +
#   geom_line() +
#   ggtitle("100 Simulations of a White Noise") +
#   theme_bw() +
#   theme(legend.position = "none") +
#   xlab("t") + ylab("X_t") +
#   geom_hline(aes(yintercept = -1.96), slope = 0, color = "blue4") +
#   geom_hline(aes( yintercept = 1.96), color = "blue4")
```

b)- Do you think this is a stationary time series? Why or why not?

Random walk

- A random walk without drift can be defined as:

$$Y_t = Y_{t-1} + W_t$$

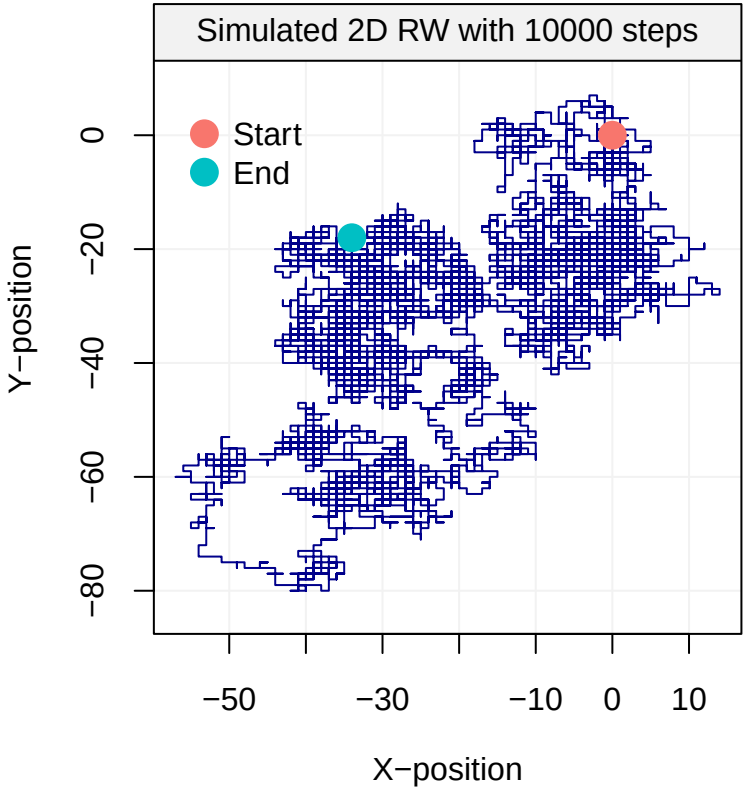
where W_t is a Gaussian white noise process with initial condition $Y_0 = c$ (usually c=0).

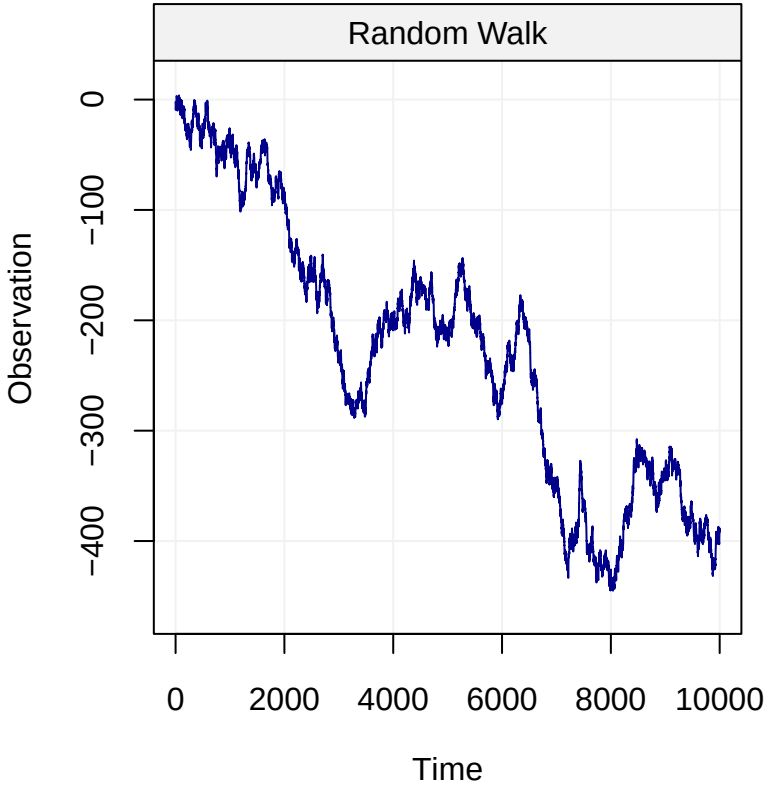
- By back substitution:

$$Y_t = Y_{t-1} + W_t = (Y_{t-2} + W_{t-1}) + W_t = \sum_{i=1}^{t} W_i + Y_0 = \sum_{i=1}^{t} W_i + c$$

- A random walk is defined as the cumulative sum of all the random white noise realizations that preceeded it.

- An example of a random walk is a drunk person on Saturday night who is walking on the street, and their next step can either be to their left, right, forward, or backward (each with equal probability). The plots below display one realization of such a random walk:





- A random walk with drift is given by:

$$Y_t = \delta + Y_{t-1} + W_t = \delta + (\delta + Y_{t-2} + W_{t-1}) + W_t = \delta \cdot t + \sum_{i=1}^{t} W_i + Y_0 = \delta \cdot t + \sum_{i=1}^{t} W_i + c$$

a)- What is $E(Y_t)$ and $Var(Y_t)$?

b)- Is a random walk with drift covariance stationary?

c)- Write a function to simulate random walks without a drift and $Y_0 = 0$, and plot 100 simulated random walks without drift using the following code.

```
rw_no_drift <- function(n = 100) {
  return(1) # Replace with your code
}

### After you write the function uncomment and run the code to produce the plot
# data <- data.frame(t = seq(from = 1, to = 100, by = 1), replicate(rw_no_drift(), n = 100))
#
# data <- data %>%  pivot_longer(
#    cols = starts_with("x"),
#    names_to = "x",
#    values_to = "value"
#  )
#
#
# ggplot(data, aes(x = t, y = value, col = x)) +
#    geom_line() +
#    ggtitle("100 Simulations of a Random Walk without drift") +
#    theme_bw() +
#    theme(legend.position = "none") +
#    xlab("t") + ylab("X_t")
```

d)- Repeat steps a-c for a random walk with drift $\delta = 0.2$.

```
rw_drift <- function(n = 100) {
  return(1) # Replace with your code
}

### After you write the function uncomment and run the code to produce the plot
```

```r
# data <- data.frame(t = seq(from = 1, to = 100, by = 1), replicate(rw_drift(), n = 100))
# #
# data <- data %>%  pivot_longer(
#     cols = starts_with("x"),
#     names_to = "x",
#     values_to = "value"
#   )
#
#
#  ggplot(data, aes(x = t, y = value, col = x)) +
#     geom_line() +
#     ggtitle("100 Simulations of a Random Walk with drift") +
#     theme_bw() +
#     theme(legend.position = "none") +
#     xlab("t") + ylab("X_t")
```

e)- What is the main difference between a random walk with and without drift?

First-Order Autoregressive Model

- A first-order autoregressive model or AR(1) is defined as:

$$Y_t = \phi Y_{t-1} + W_t$$

- by back substitution:

$$Y_t = \phi Y_{t-1} + W_t = \phi(\phi_t Y_{t-2} + W_{t-1}) + W_t = \phi^2 Y_{t-2} + \phi W_{t-1} + W_t = \phi^t \cdot Y_0 + \sum_{i=0}^{t-1} \phi^i W_{t-i}$$

- If $|\phi| < 1$, then $lim_{i\to\infty}\phi^i Y_{t-i} = 0$ and the $AR(1)$ process (Y_0=0) is:

$$Y_t = \sum_{i=0}^{t-1} \phi^i W_{t-i}$$

a)- How is an AR(1) process related to white noise and a random walk?

b)- Is this AR(1) process stationary?

c)- What happens if $\phi = 1$ or $\phi > 1$?

d)- Use the following code to simulate 100 realizations of an AR(1) process with $\phi = 0.5$.

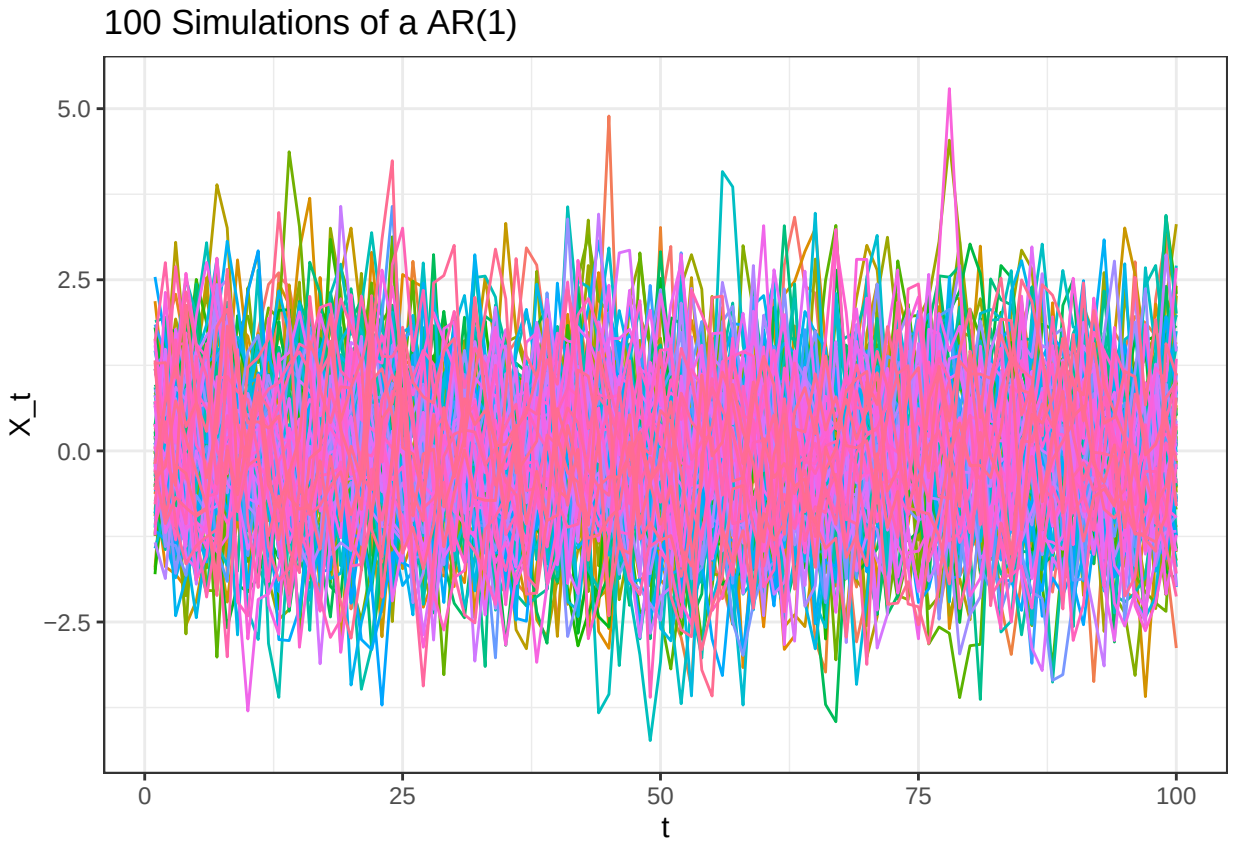
```
## ar_1()
phi = 0.5

ar_1 <- function(n = 100) {
  y <- w <- rnorm(n, mean = 0, sd = 1)
  for (t in 2:100) y[t] <- phi* y[t - 1] + w[t]
    return(y)
  }

data <- data.frame(t = seq(from = 1, to = 100, by = 1), replicate(ar_1(), n = 100))

data <- data %>%  pivot_longer(cols = starts_with("x"),
                               names_to = "x",values_to = "value")
ggplot(data, aes(x = t, y = value, col = x)) +
     geom_line() +
     ggtitle("100 Simulations of a AR(1)") +
```

```
    theme_bw() +
    theme(legend.position = "none") +
    xlab("t") + ylab("X_t")
```



Moving Average Process of Order 1

- An AR(1) can be written as a linear combination of all past white noise (Wt). This is known as invertibility of AR(1) processes.
- Similarly, an MA(1) can be written as a linear combination of the white noises, but it is a "truncated" version and only includes two white noise terms.

$$Y_t = \theta W_{t-1} + W_t$$

a)- Is the MA(1) process stationary?

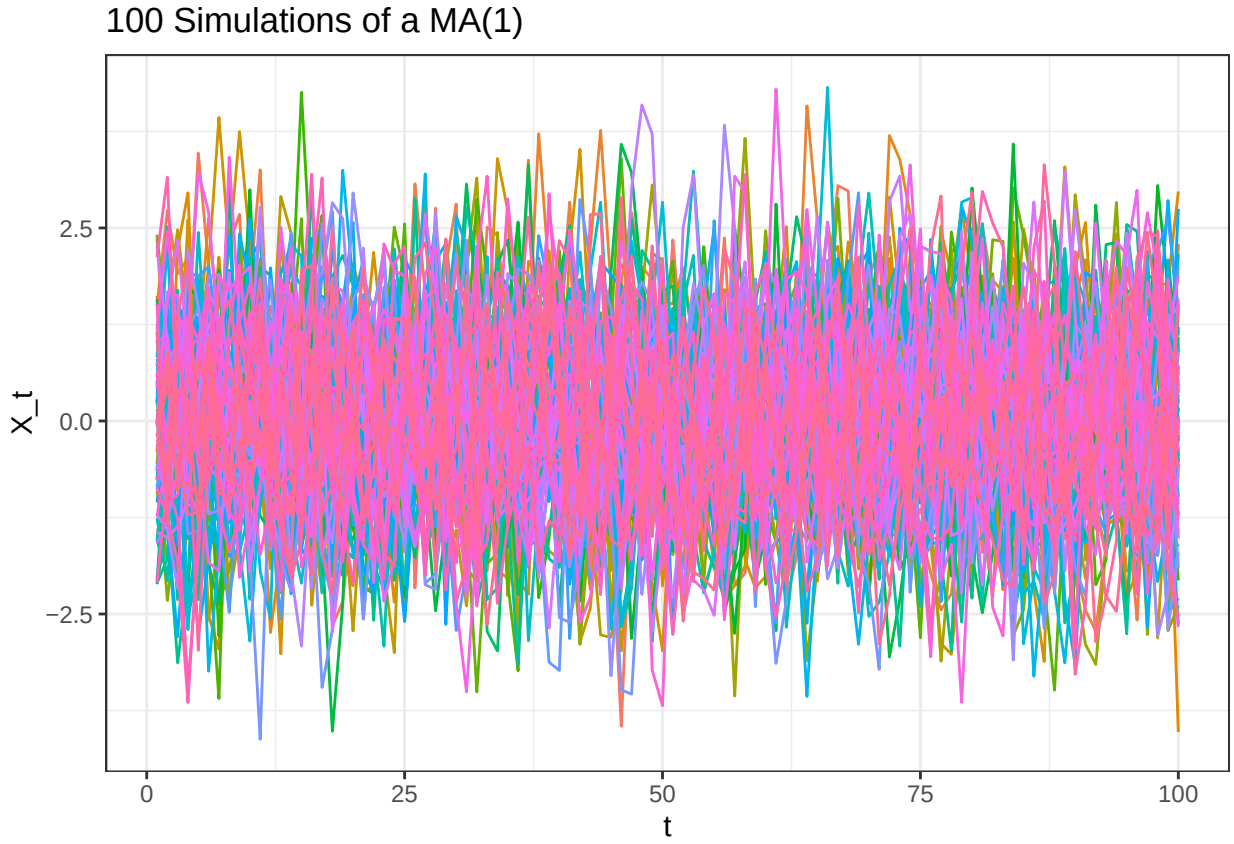
b)- Run following code to simulate 100 realizations of a MA(1) model with $\phi = 0.5$.

```
## ma_1()
theta = 0.5
ma_1 <- function(n = 100) {
      y <- w <- rnorm(n, mean = 0, sd = 1)
  for (t in 2:100) y[t] <- theta* w[t - 1] + w[t]
  return(y)
}

data <- data.frame(t = seq(from = 1, to = 100, by = 1), replicate(ma_1(), n = 100))

data <- data %>%  pivot_longer(
     cols = starts_with("x"),
     names_to = "x",
     values_to = "value"
   )


ggplot(data, aes(x = t, y = value, col = x)) +
     geom_line() +
     ggtitle("100 Simulations of a MA(1)") +
     theme_bw() +
     theme(legend.position = "none") +
     xlab("t") + ylab("X_t")
```


100 Simulations of a MA(1)
X_t
2.5
0.0
−2.5
0
25
50
75
100
t

Time Series classes in R

R has several libraries and classes for dealing with time series data. Of course it is possible to represent time series in a 'normal' data frame with one column for the time period and another column for the observations in the series. But the data will usually be better suited for analysis when it takes the form of one of R's specialized time series classes, which have different properties from regular data frames.

ts objects

- The `ts` object is the most basic type of time-series object in R, requiring only the base `stats` package which is automatically loaded when you start R.
- `ts` objects come with `frequency`, `start`, and `end` arguments. The `frequency` attribute specifies the number of (regularly-spaced) intervals per unit of time; a frequency of `7` might correspond to daily intervals of weekly units, `52` might correspond to weekly intervals of annual units, while `1` might correspond to annual data or any data where there is only one interval per time-unit.
- The `start` and `end` attributes consist of either a single number specifying a time unit, or a vector of two numbers specifying both a time unit and a particular number of intervals in that unit.
- To extract a subset of the time series, you can use the `window()` function with `start` and/or `end` arguments.
- The `time()` function returns the numeric time stamps for the observations.

xts and zoo objects

- `xts` stands for eXtensible Time Series. It is essentially matrix + (time-based) index (aka, observation + time), which allows irregular time intervals.
- `xts` is a constructor or a subclass that inherits behavior from parent `zoo` (Z's Ordered Observations). It extends the popular `zoo` class, and most `zoo` methods work for xts. These include methods for subsetting, merging, and interpolating time series data.
- `xts` are indexed by a formal time object. Therefore, the data is time-stamped. The two most important arguments are `x` for the data and `order.by` for the index. `x` must be a vector or matrix. `order.by` is a vector of the same length or number of rows of `x`; it must be a proper time or date object and be in an increasing order. The `coredata()` and `index()` functions retrieves the observations and their time stamps respectively.

tsibble objects

- `tsibble` objects are a variants of `tibble` objects, which are variants of data frames, and can be used with `dplyr` data-wrangling functions. They require specification of an index, which can be regular or irregular; in the case of regular intervals, the `yearquarter`, `yearmonth`, `yearweek`, `Date`, and `POSIXct` functions can convert time information to the appropriate class. Tsibble definitions can also include specification of a key variable allowing multiple time-series to be manipulated as a single object. The `tsibble` library has various functions for subsetting, merging, and interpolating time series data.

Reminders

1. Welcome to the Time Series part of the course!
2. Before the next live session:
    1. Complete the HW-6
    2. Complete all videos and reading for unit 7