MATH 8090: Review and Overview of the Course

Whitney Huang, Clemson University

8/21/2025

Contents

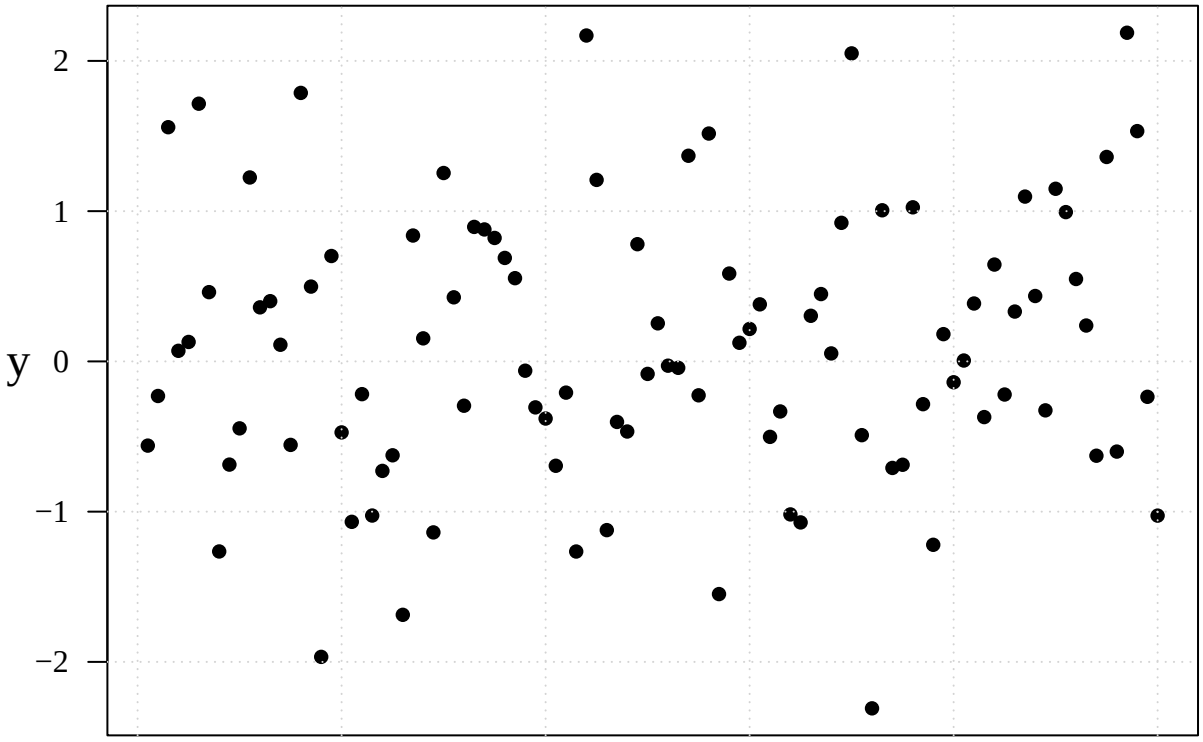
Review of Statistical Inference

Inference for a Mean (One-Sample t-Interval)

For an i.i.d. sample $X_1, \ldots, X_n$ from a population with unknown mean μ and variance σ^2, we want to infer μ.

```
n <- 100
set.seed(123)
y <- rnorm(n)

par(mar = c(3, 3.5, 0.5, 0.6), mgp = c(2, 1, 0), las = 1,
    family = "serif")
plot(y, pch = 16, ylab = "", xaxt = "n", cex.lab = 1.5)
mtext(expression(y), 2, line = 2, cex = 1.5)
grid()
```

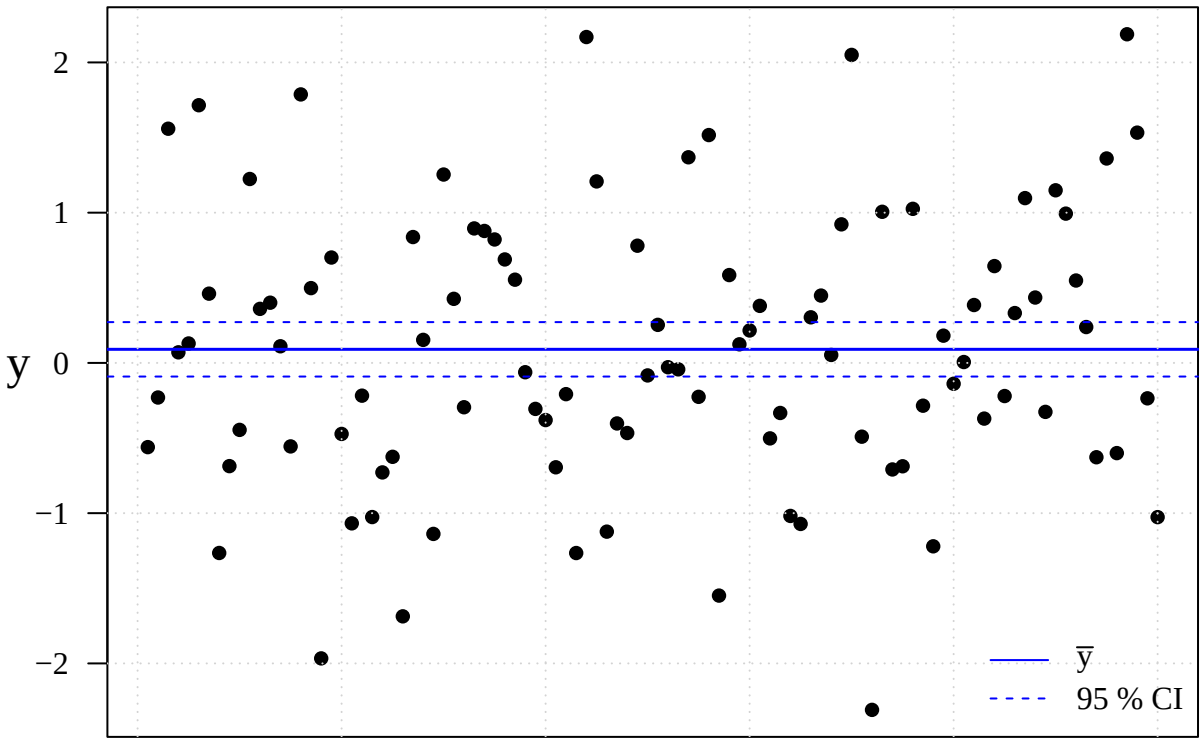


A $(1-\alpha)100\%$ confidence interval for μ:

$$\bar{x}_n \pm t_{n-1,\,1-\alpha/2} \cdot \frac{s}{\sqrt{n}}$$

```
se <- sd(y) / sqrt(n)
par(mar = c(3, 3.5, 0.5, 0.6), mgp = c(2, 1, 0), las = 1,
    family = "serif")
plot(y, pch = 16, ylab = "", xaxt = "n", cex.lab = 1.5)
mtext(expression(y), 2, line = 2, cex = 1.5)
```

```r
abline(h = mean(y), col = "blue", lwd = 1.5)
abline(h = mean(y) + c(-1, 1) * qt(0.975, 99) * se,
       col = "blue", lty = 2)
grid()
legend("bottomright",
       legend = c(expression(bar(y)), "95 % CI"),
       col = "blue", lty = c(1, 2), bty = "n")
```



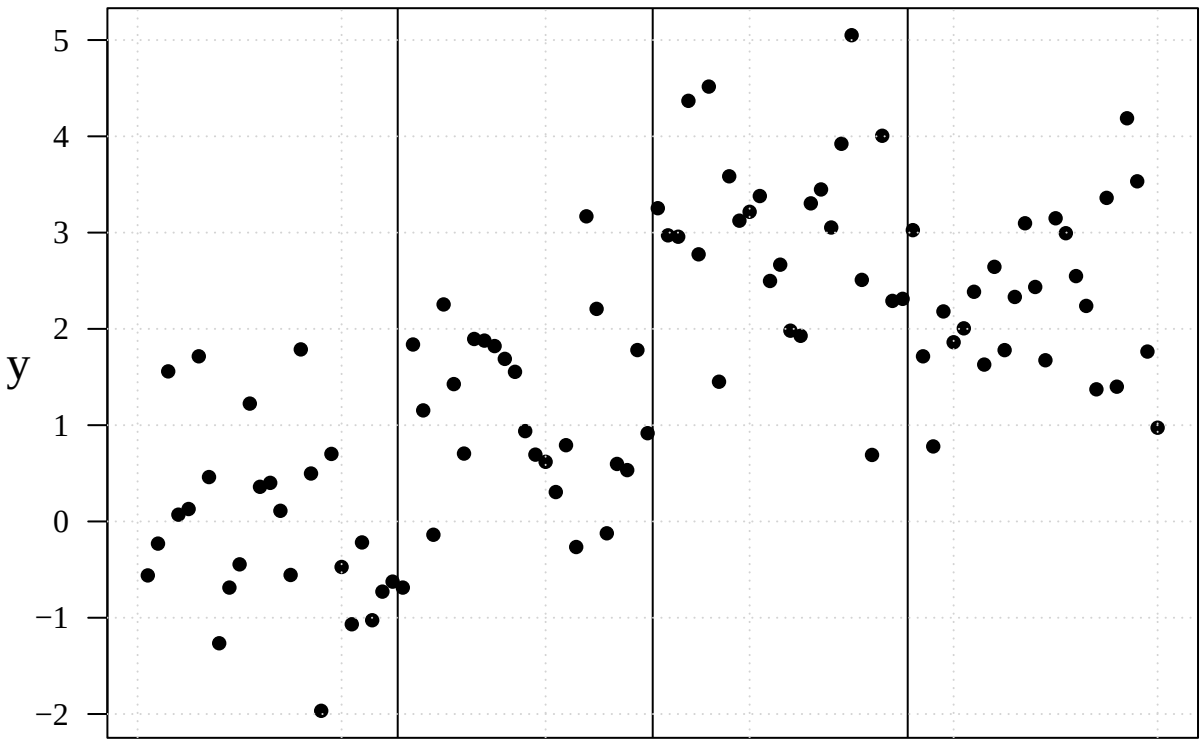
Index

Inference for Group Means (ANOVA)

For a sample $x_{ij}, ; i = 1, \ldots, k, ; j = 1, \ldots, n_i$, where $x_{ij} = \mu_i + \varepsilon_{ij}$ and the ε_{ij} are i.i.d., we want to compare the group means $\mu_1, \mu_2, \ldots, \mu_k$.

```r
means <- rep(c(0, 1, 3, 2), each = 25)
y1 <- y + means

par(mar = c(3, 3.5, 0.5, 0.6), mgp = c(2, 1, 0), las = 1,
    family = "serif")
plot(y1, pch = 16, ylab = "", xaxt = "n", cex.lab = 1.5)
mtext(expression(y), 2, line = 2, cex = 1.5)
abline(v = c(25.5, 50.5, 75.5))
grid()
```



Index

We estimate σ^2 using the **Mean Squared Error (MSE)**:

$$\hat{\sigma}^2 = MSE = \frac{SS_W}{N-k},$$

where SS_W is the within-group sum of squares, N is the total sample size, and k is the number of groups.

For group i, the sample mean is $\bar{x}_i$ with group size n_i.
The standard error (SE) of $\bar{x}_i$ under ANOVA is

$$SE(\bar{x}_i) = \sqrt{\frac{MSE}{n_i}}.$$

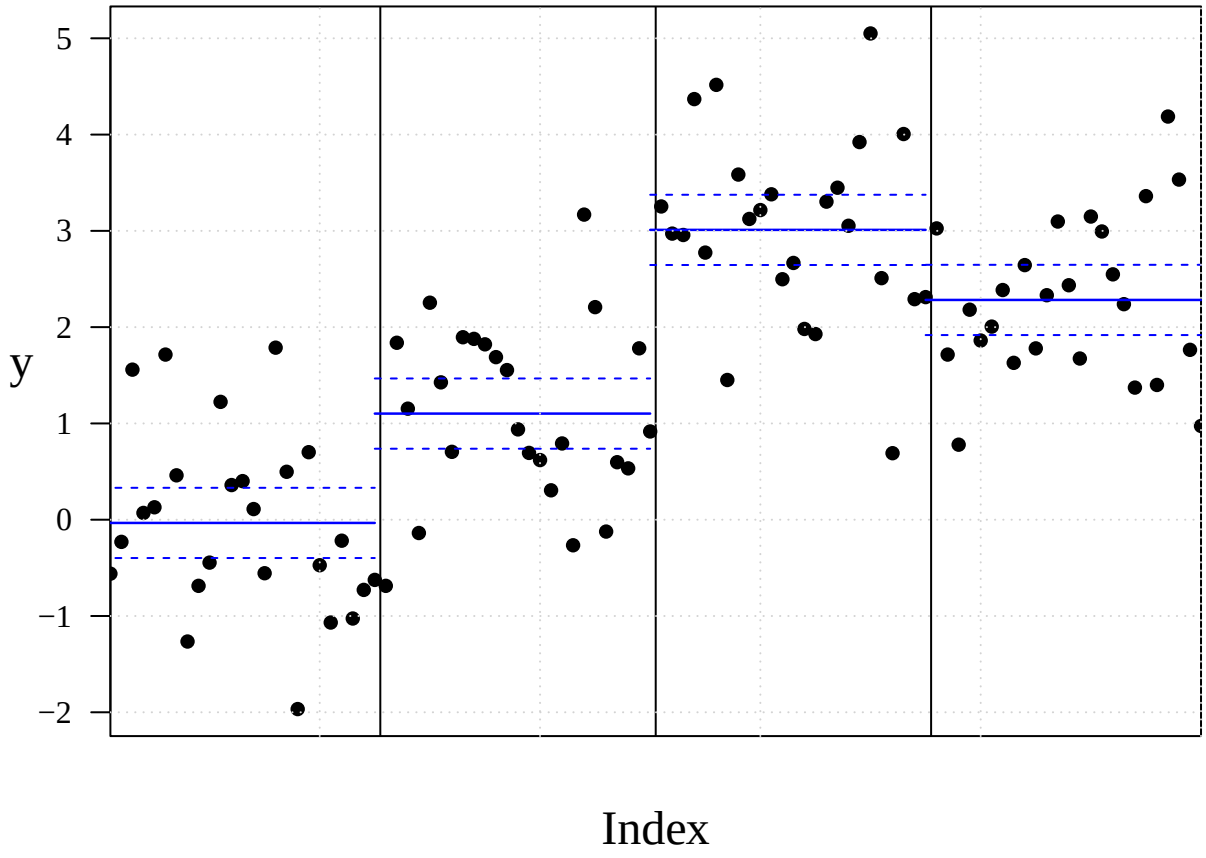
Thus, a $(1-\alpha)100\%$ confidence interval for the group mean μ_i is

$$\bar{x}_i \pm t_{N-k,\,1-\alpha/2} \cdot \sqrt{\frac{MSE}{n_i}}.$$

```
dat <- array(y1, dim = c(25, 4))
means <- apply(dat, 2, mean)
se <- sqrt((sum(apply(dat, 2, var) * 24) / 96) / 25)

par(mar = c(3, 3.5, 0.5, 0.6), mgp = c(2, 1, 0), las = 1,
    family = "serif")
plot(y1, pch = 16, ylab = "", xaxt = "n", cex.lab = 1.5, xaxs = "i")
mtext(expression(y), 2, line = 2, cex = 1.5)
abline(v = c(25.5, 50.5, 75.5))
for (i in 1:4){
  segments((i - 1) * 25, means[i], 25 * i, col = "blue", lwd = 1.25)
```

```
  segments((i - 1) * 25, means[i] + qt(0.975, 96) * se, 25 * i, col = "blue", lty = 2)
  segments((i - 1) * 25, means[i] - qt(0.975, 96) * se, 25 * i, col = "blue", lty = 2)
}
grid()
```



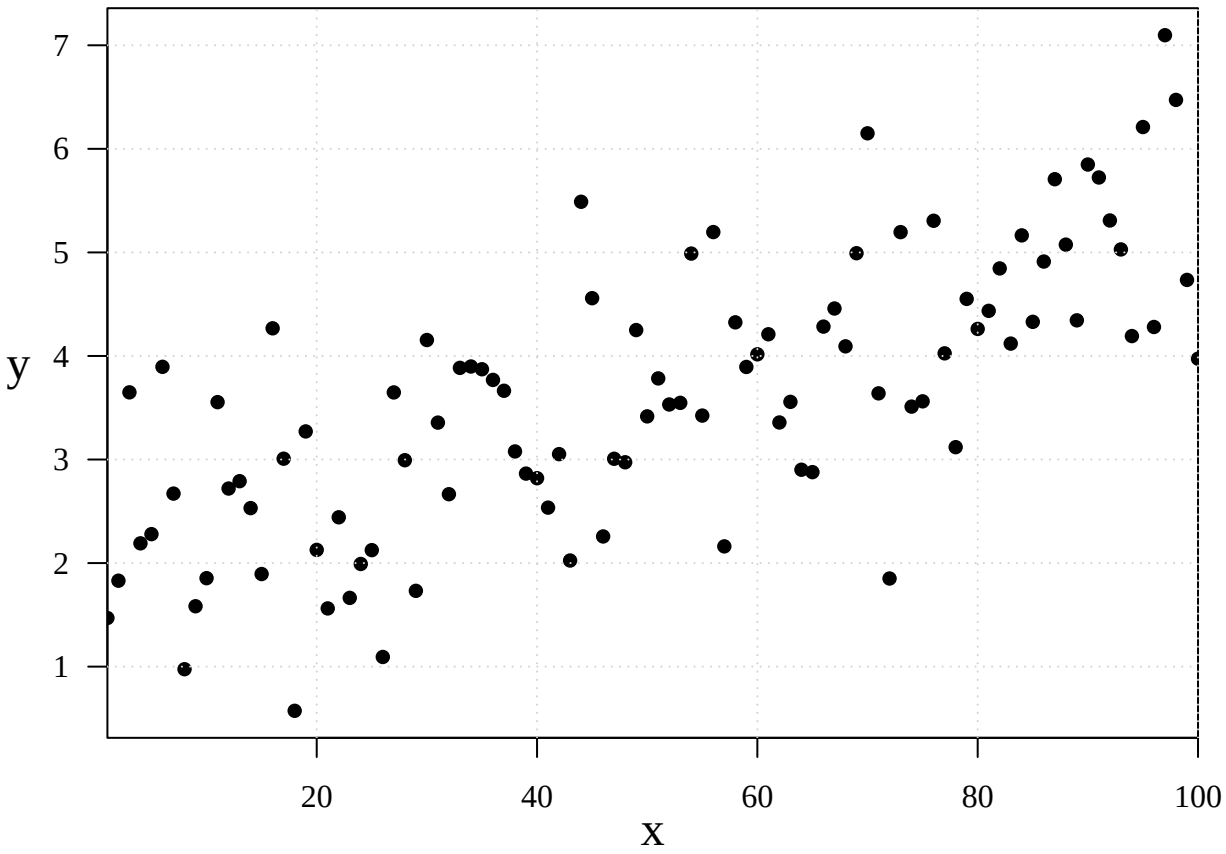
Inference for Regression Function (Simple Linear Regression)

Model:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim \text{i.i.d. } N(0, \sigma^2)$$

```
x <- 1:100
y2 <- 2 + 0.03 * x + y

par(mar = c(3, 3.5, 0.5, 0.6), mgp = c(2, 1, 0), las = 1,
    family = "serif")
plot(y2 ~ x, pch = 16, ylab = "", cex.lab = 1.5, xaxs = "i")
mtext(expression(y), 2, line = 2, cex = 1.5)
grid()
```

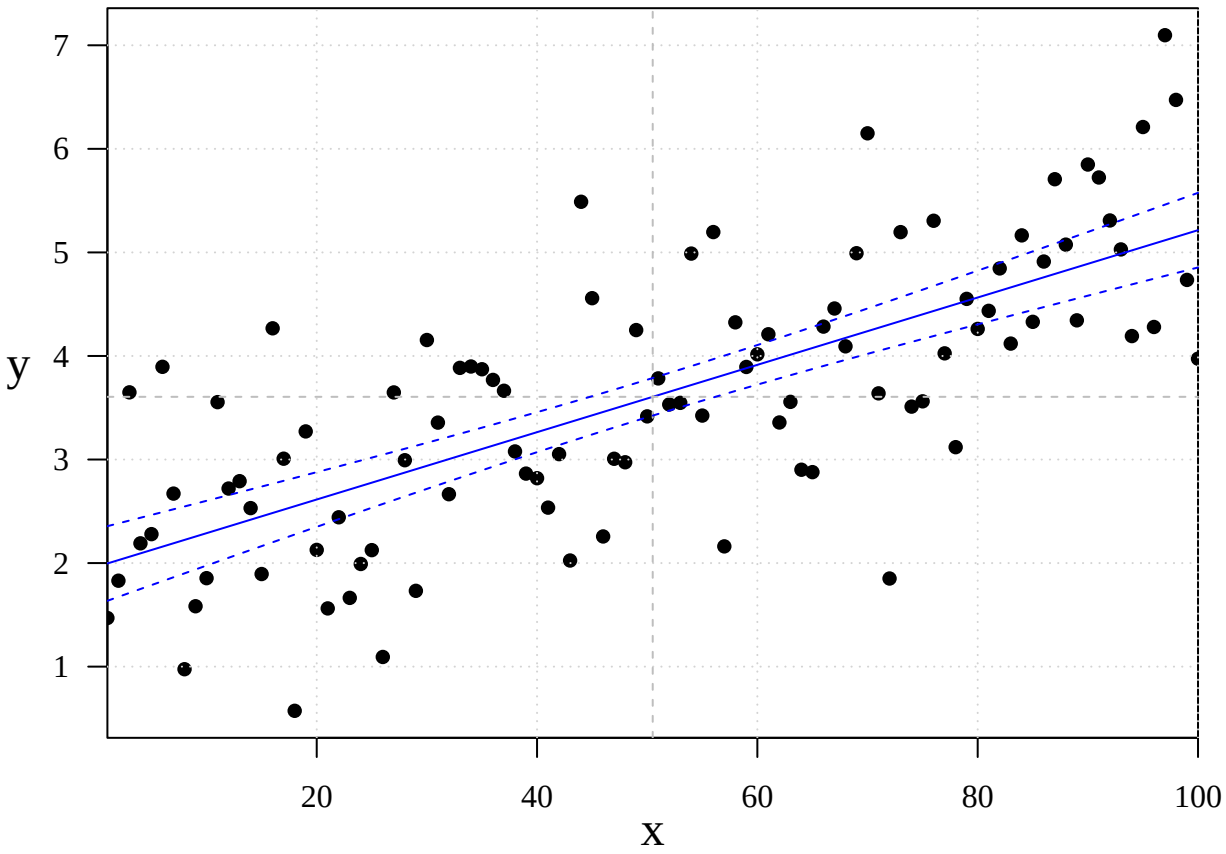


**Confidence interval for mean response at x_0:

$$\hat{y}_0 \pm t_{n-2,\,1-\alpha/2} \cdot s\sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

```r
lm <- lm(y2 ~ x)
x_grid = data.frame(x = seq(1, 100, len = 1000))
CI_band <- predict(lm, x_grid, interval = "confidence")

par(mar = c(3, 3.5, 0.5, 0.6), mgp = c(2, 1, 0), las = 1,
    family = "serif")
plot(y2 ~ x, pch = 16, ylab = "", cex.lab = 1.5, xaxs = "i")
mtext(expression(y), 2, line = 2, cex = 1.5)
grid()
abline(lm, col = "blue")
abline(v = mean(x), lty = 2, col = "gray")
abline(h = mean(y2), lty = 2, col = "gray")
lines(x_grid[, 1], CI_band[, 2], lty = 2, col = "blue")
lines(x_grid[, 1], CI_band[, 3], lty = 2, col = "blue")
```

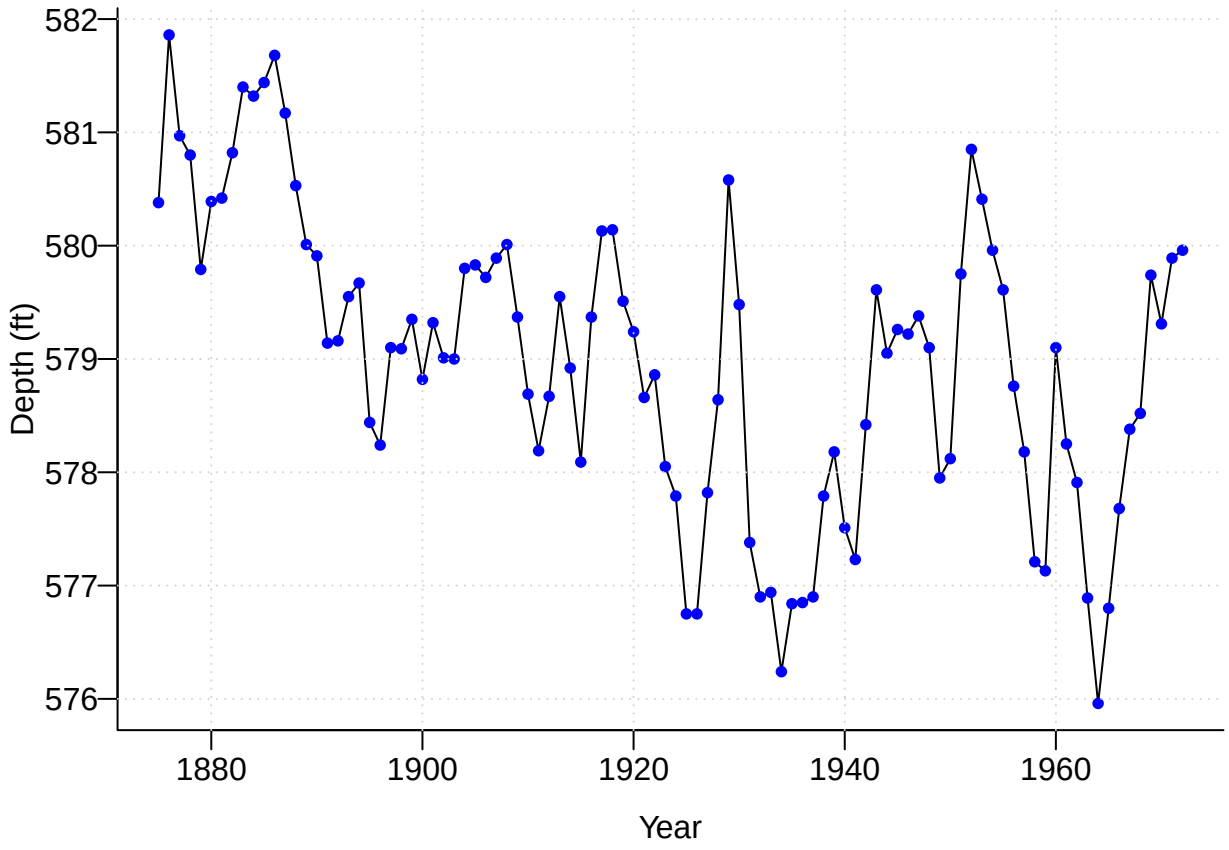


Time Series Data

Lake Huron Time Series

Annual measurements of the level of Lake Huron in feet

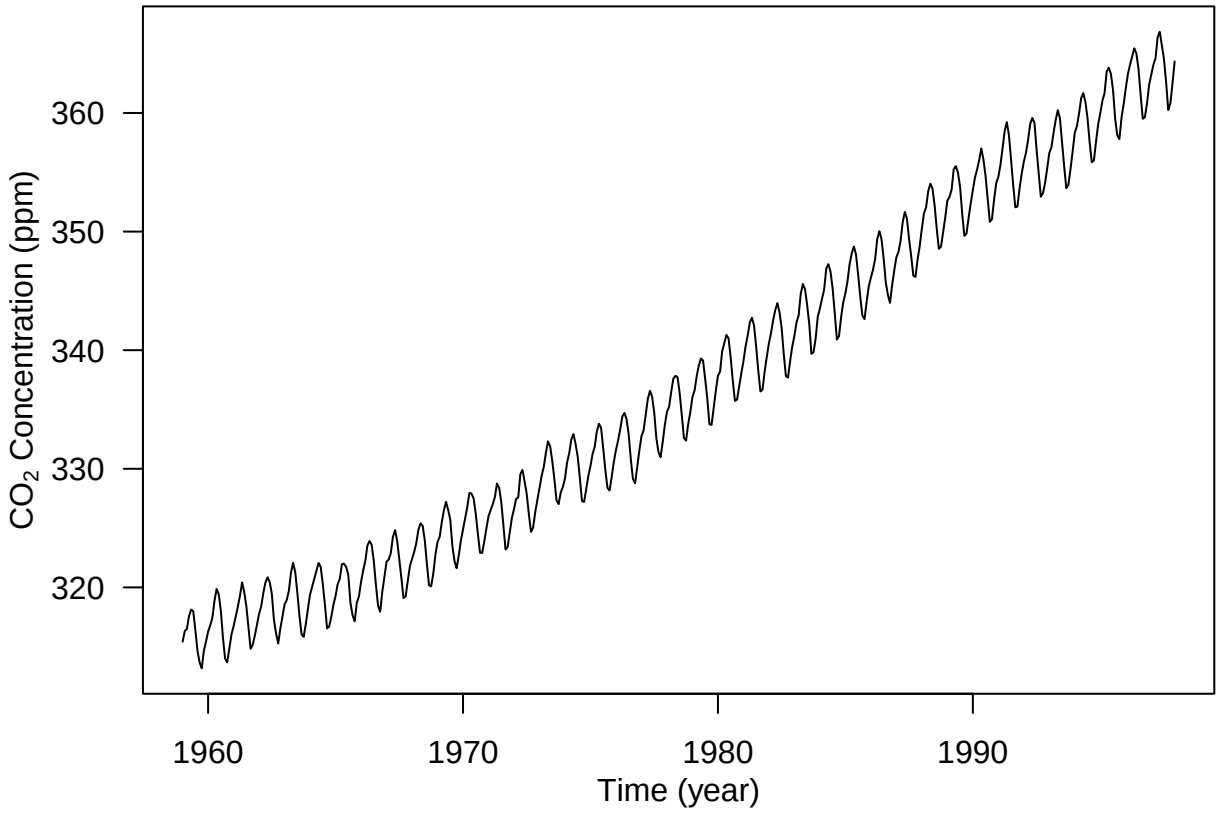
```
par(mar = c(3.2, 3.2, 0.5, 0.5), mgp = c(2, 0.5, 0), bty = "L")
data(LakeHuron)
plot(LakeHuron, ylab = "Depth (ft)", xlab = "Year", las = 1)
points(LakeHuron, cex = 0.8, col = "blue", pch = 16)
grid()
```



CO_2 Concentration

Atmospheric concentrations of CO2 are expressed in parts per million (ppm) and reported in the preliminary 1997 SIO manometric mole fraction scale.

```
data(co2)
par(mar = c(3.8, 4, 0.8, 0.6))
plot(co2, las = 1, xlab = "", ylab = "")
mtext("Time (year)", side = 1, line = 2)
mtext(expression(paste("CO"[2], " Concentration (ppm)")), side = 2, line = 2.5)
```

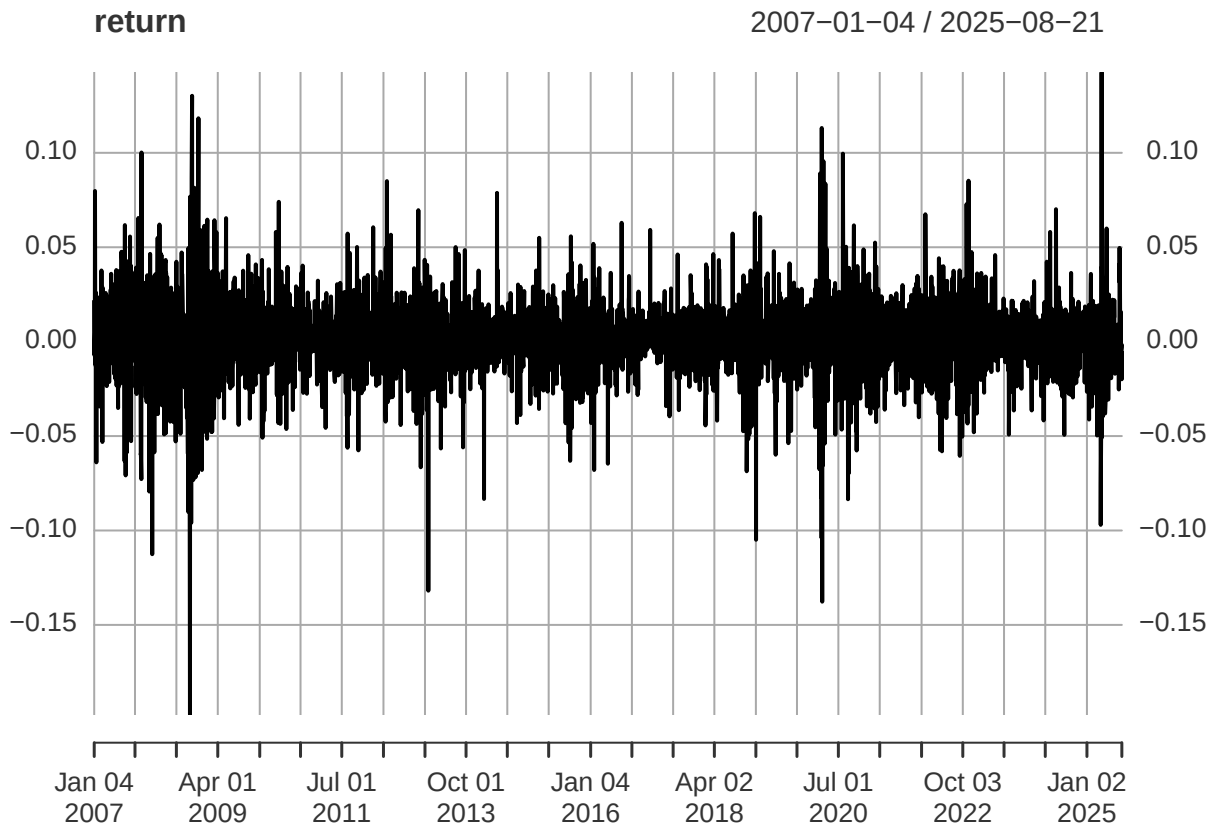


Apple daily log returns

```
library(quantmod)
getSymbols("AAPL", src = "yahoo")
```

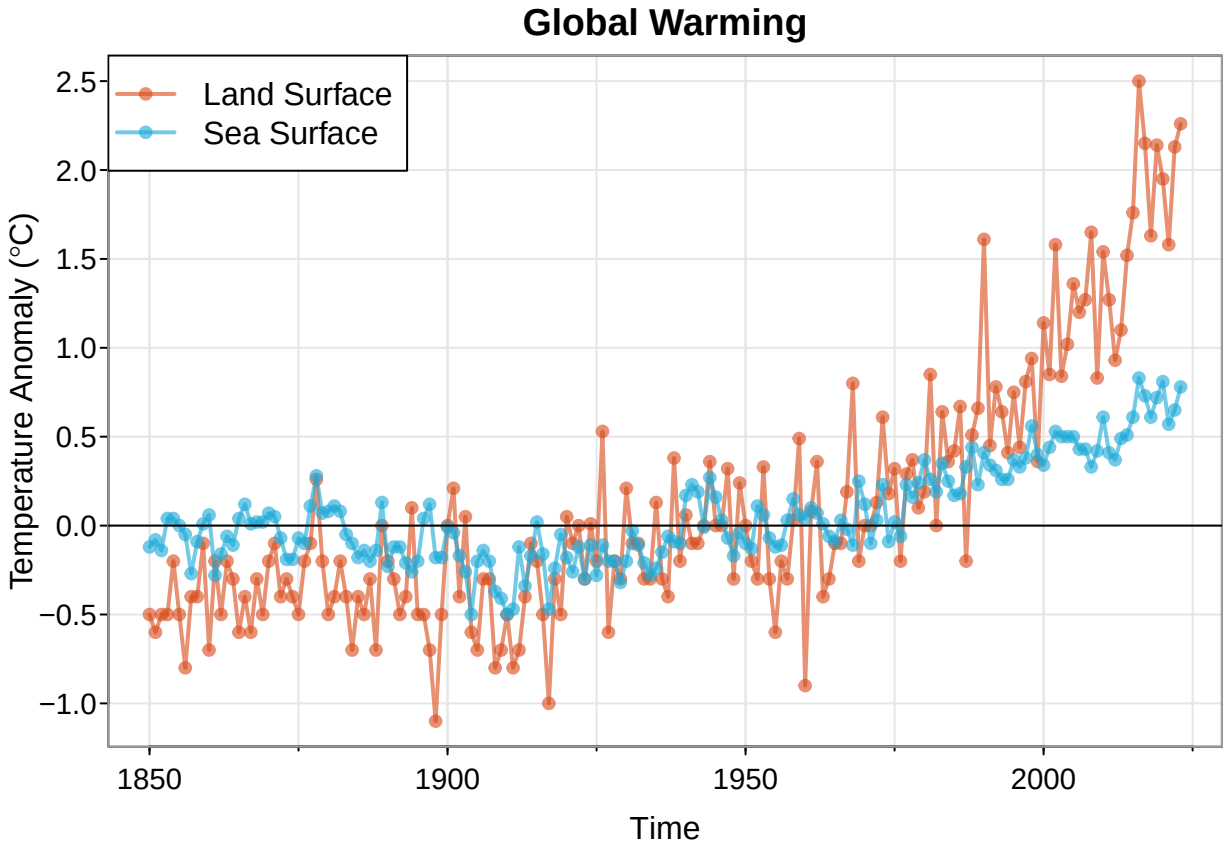
```
## [1] "AAPL"
```

```
closing <- AAPL$AAPL.Close
return <- diff(log(closing))[!is.na(diff(log(closing))$AAPL.Close)]
par(las = 1, mgp = c(2.2, 1, 0), mar = c(3.6, 3.6, 0.8, 0.6))
plot(return)
```



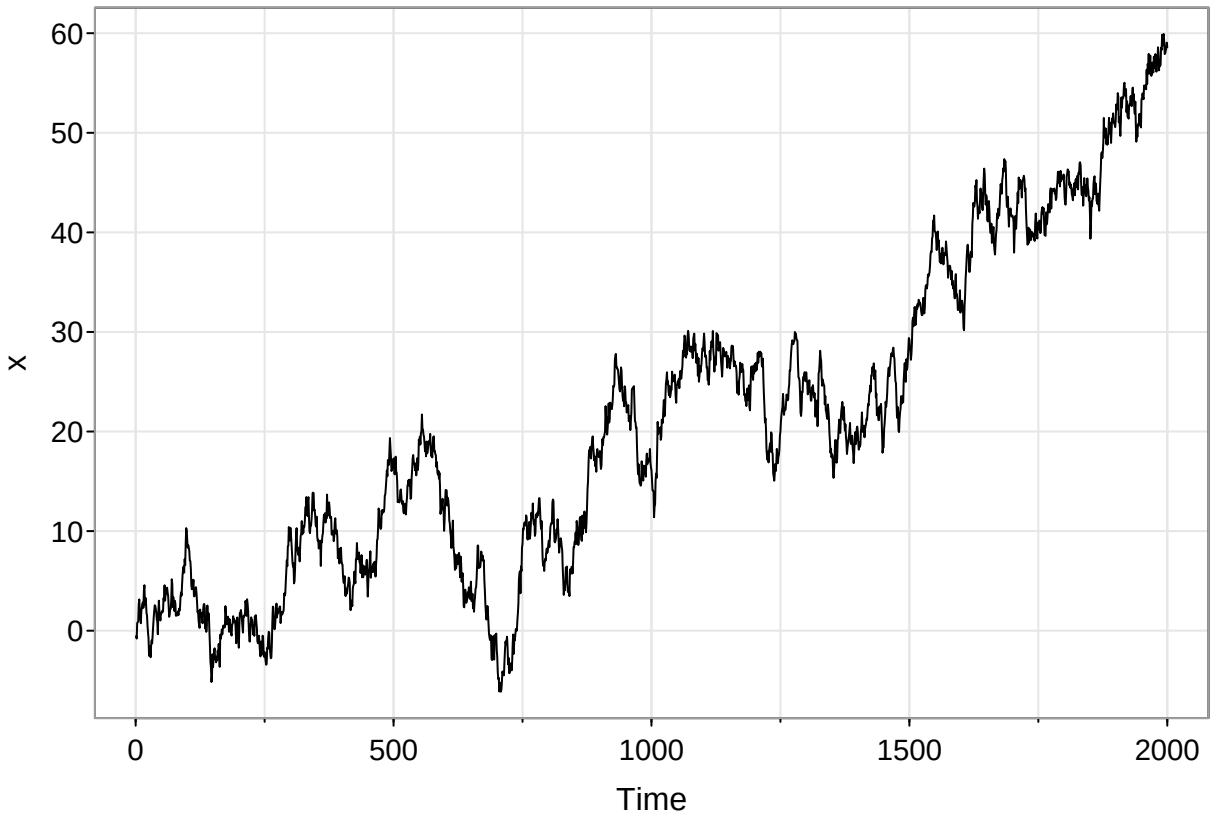
Global mean land temperature anomalies

```
library(astsa)
culer = c(rgb(217, 77, 30, 160, max = 255), rgb(30, 170, 217, 160, max = 255))
tsplot(gtemp_land, col = culer[1], lwd = 2, type = "o", pch = 20, las = 1,
ylab = expression(paste("Temperature Anomaly (", degree, "C)")), main = "Global Warming")
lines(gtemp_ocean, col = culer[2], lwd = 2, type = "o", pch = 20)
abline(h = 0)
legend("topleft", col = culer, lty = 1, lwd = 2, pch = 20,
       legend = c("Land Surface", "Sea Surface"), bg = "white")
```



Simulated time series

```
set.seed(123)
w <- rnorm(2000); x <- cumsum(w); tsplot(x, las = 1)
```



A time series is a collection of observations $\{y_t : t \in T\}$ taken sequentially in time t, where the index set T denotes the time points. These observations can be gathered at fixed (equidistant) time points (e.g., $T = \{0, 1, 2, \cdots\}$), at irregular time points, or even over an entire time interval (e.g., $T = [0, T]$).

Different types of time sampling require different approaches to data analysis. In this course, we will focus on some of the most commonly used methods for dealing with the first scenario: *discrete-time* equal-spaced time series. A discrete-time time series might be intrinsically discrete (e.g., number of planes departing ATL every day) or might arise from an underlying *continuous-time* time series via

- *Sampling*: For example, instantaneous wind speed
- *Aggregation*: Such as daily accumulated precipitation amount
- *Extrema*: Like the daily maximum temperature

It is worth noting that the methods for continuous-time time series (where T represents an entire interval) are useful in situations where data were observed at irregular times.

Depending on the type of value that y_t can take, we have:

- *Real-valued*: a value in $\mathbb{R}$ or a subset thereof (e.g., temperature)
- *Complex-valued* (some applications in electrical engineering)
- *Non-negative integer* (See Jia et al. (2021) for a recent development for modeling such kind of count time series data)
- *Categorical* (e.g., outcome of scheduled basketball match: win, lose, cancellation)
- *Circular* (e.g., wind direction. See Breckling (2012))

Exploratory Time Series Analysis

Suppose we have time series data $\{y_t : t \in T\}$. We will start with a *time series plot* of y_t against t. It is important to label the axes accurately (i.e., the time unit on the x-axis, variable name and its unit on the

y-axis) and adjust the aspect ratio to create an aesthetically pleasing time series plot. In this exploratory data analysis (EDA) stage, we look the following:

1. Are there abrupt changes? (e.g., shifts in mean and/or variance)
2. Are there *outliers*? (i.e., values that are unusual relative to the rest of the data)
3. Is there a need to transform the data (e.g., should we apply a transformation to stabilize the variance))?
4. Features of the time series: trend, seasonal components, and noise process

Trend, Seasonality, and Noise

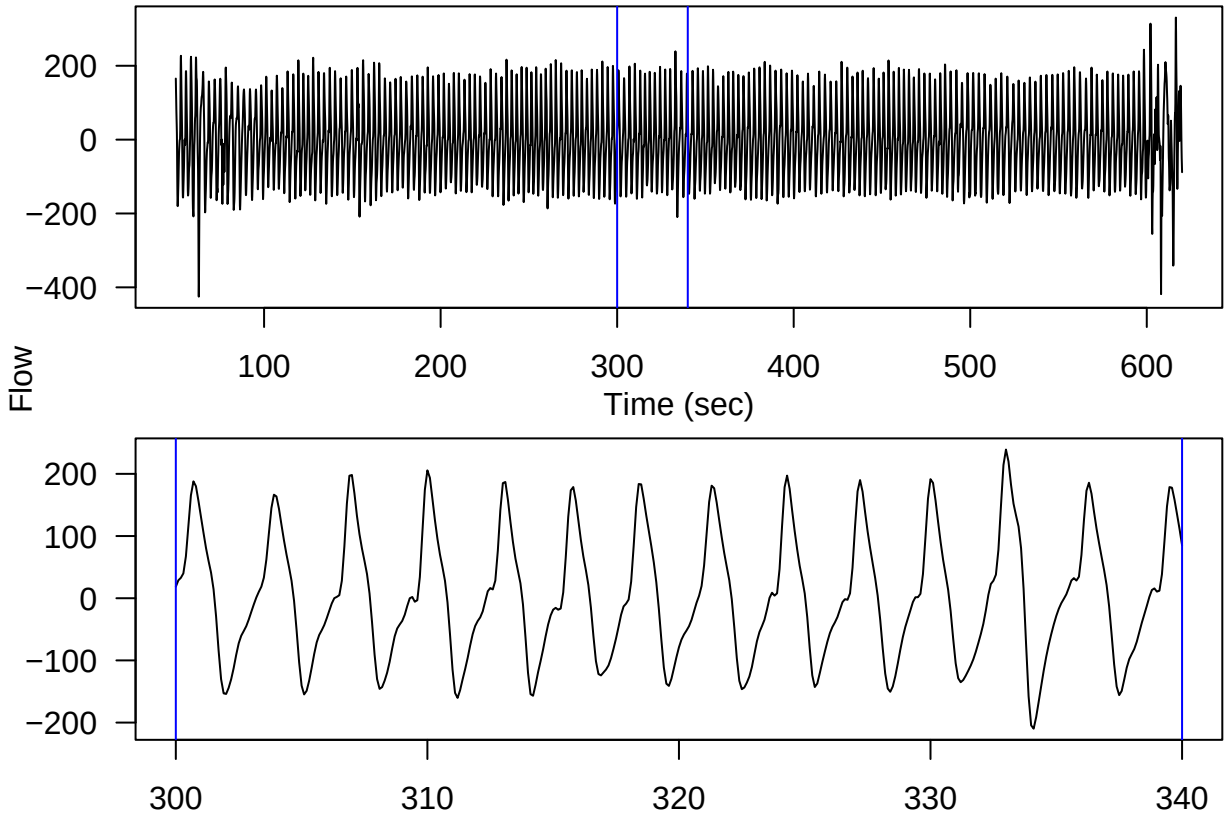
Trends

One can think of trend, μ_t, as representing continuous changes, usually in the mean, over longer time scales. *The essential idea of a trend is that it should be smooth.* Typically, the form of the trend is unknown and needs to be specified and then estimated for the data. When the trend is estimated and then removed, we obtain a detrended series

Seasonal compoents

A seasonal component, s_t, constantly repeats itself in time, (i.e., $s_t = s_{t+kd}$ for all t and k). Sometimes the seasonality exhibits a very clear structure form (e.g., a sin or cosine wave), while at other times, the structure is more complex (see the sleep airflow data example used in Huang et al. (2022)).

```
load("Flow_sub1.RData")
par(mfrow = c(2, 1), mar = c(2.6, 3.6, 0.8, 0.6))
id <- 500:6200
plot(id / 10, flow[id], type = "l", las = 1, xlab = "", ylab = "")
mtext("Time (sec)", 1, line = 2)
mtext("Flow", 2, line = 2.5, at = -650)
abline(v = c(3000, 3400) / 10, col = "blue")
id2 <- 3000:3400
plot(id2 / 10, flow[id2], type = "l", las = 1, xlab = "", ylab = "")
abline(v = c(3000, 3400) / 10, col = "blue")
```



Noises

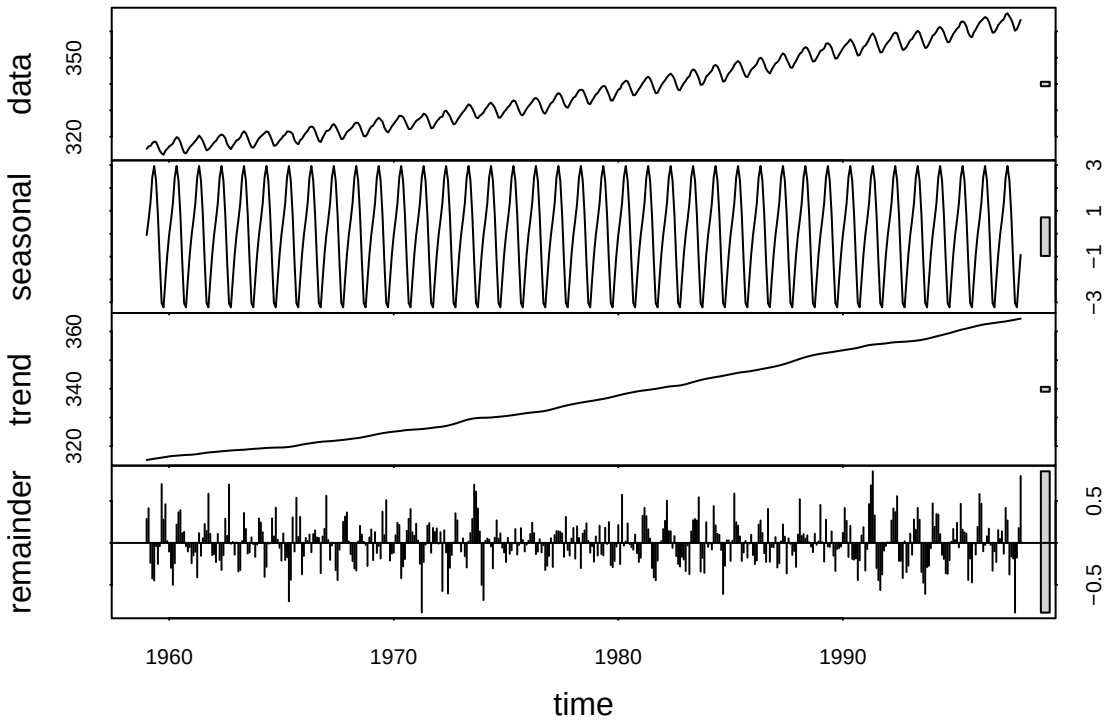
The noise process, η_t, is the component that is neither trend nor seasonality. Our focus will be on identifying plausible (typically stationary) statistical models for this process.

Combining Trend, Seasonality, and Noise Together

There are two common approaches to combine these components:

1. The *additive model*, $y_t = \mu_t + s_t + \eta_t, t = 1, \cdots, T$.

```
# Seasonal and Trend decomposition using Loess (STL)
par(mar = c(4, 3.6, 0.8, 0.6))
stl <- stl(co2, s.window = "periodic")
plot(stl, las = 1)
```



2. The *multiplicative model*, $y_t = \mu_t s_t \eta_t, t = 1, \cdots, T$.

Note that if all the variables are *positive*, we can obtain the additive model by taking logarithms:

$$\log(y_t) = \log(\mu_t) + \log(s_t) + \log(\eta_t), t = 1, \cdots, T.$$

Keep in mind that we need to transform back so that we can interpret the results on the *original measurement scale*.

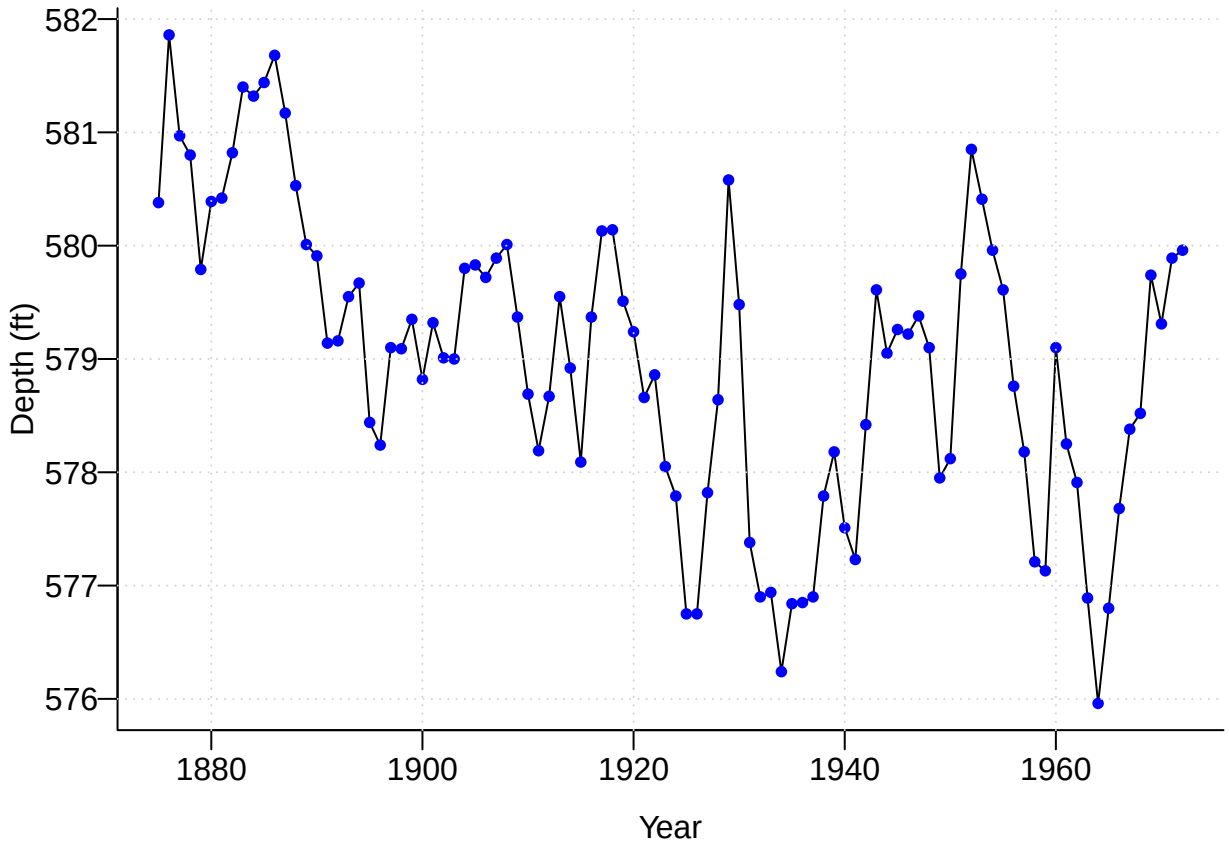
Lake Huron Time Series Example

This example is taken from the book by Brockwell et al. (2002). The data consist of annual measurements of the depth, in feet (ft), of Lake Huron from 1875-1972.

```
par(mar = c(3.2, 3.2, 0.5, 0.5), mgp = c(2, 0.5, 0), bty = "L")
data(LakeHuron); str(LakeHuron)
```

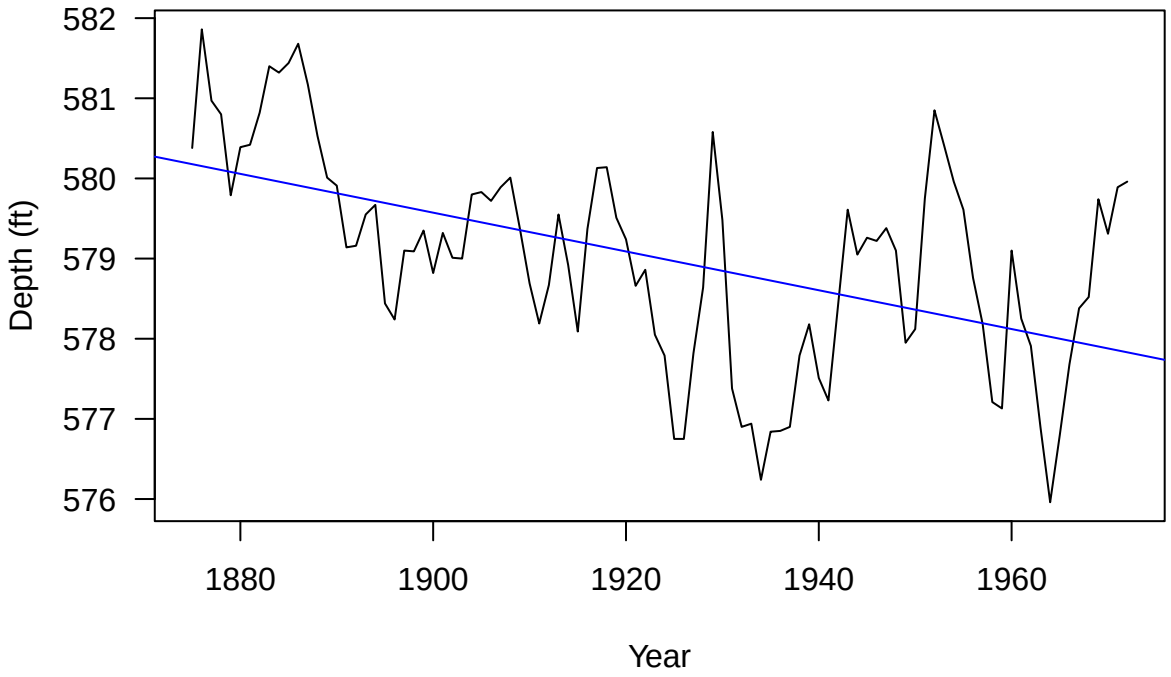
```
##  Time-Series [1:98] from 1875 to 1972: 580 582 581 581 580 ...
```

```
plot(LakeHuron, ylab = "Depth (ft)", xlab = "Year", las = 1)
points(LakeHuron, cex = 0.8, col = "blue", pch = 16)
grid()
```

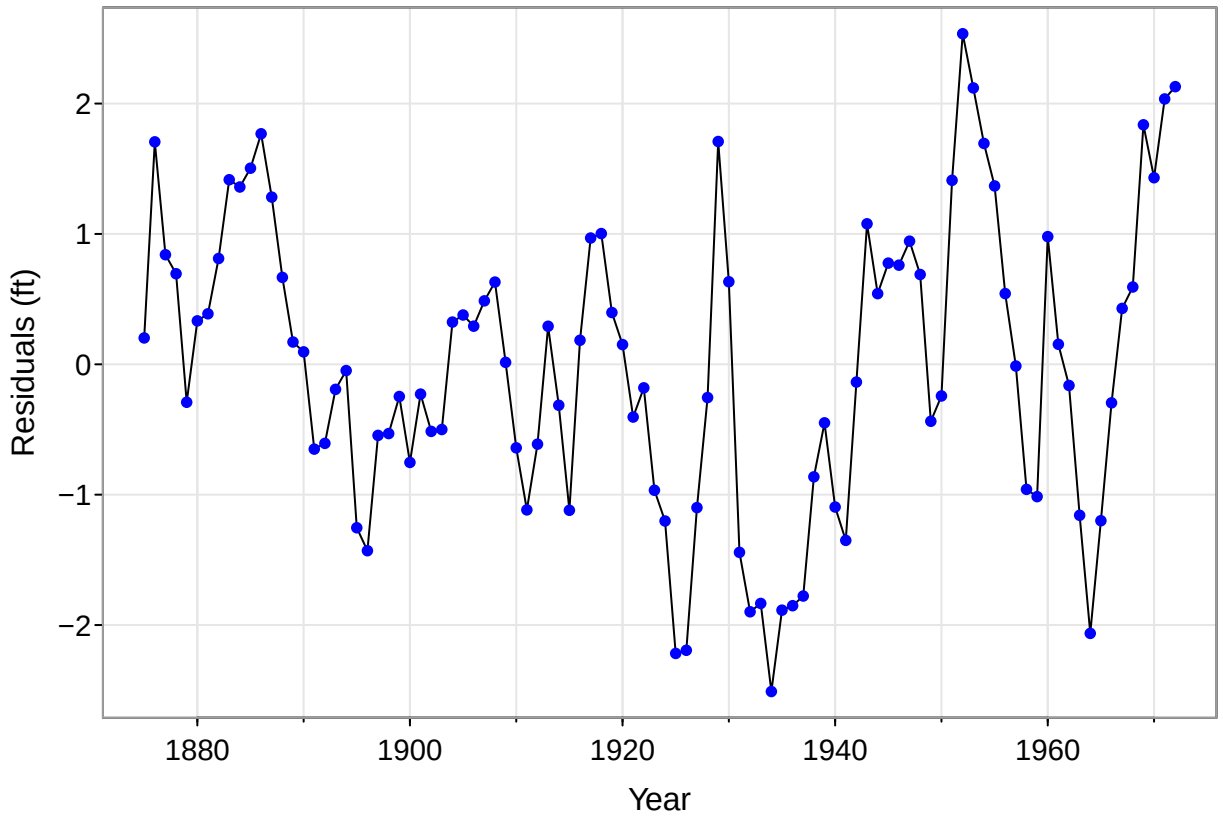


This is an example of a *discrete-time*, equal-spaced *real-valued* time series. Some key features of this time series include: 1) decreasing trend, and 2) *random* fluctuations around the decreasing trend. Let's conduct a simple analysis to this data to gain a better understanding of the nature of these random *noises*. Specifically, we will assume a linear trend for this time series, allowing us to estimate and subsequently remove the trend by performing a *simple linear regression*.

```
library(astsa)
yr <- time(LakeHuron)
lm <- lm(LakeHuron ~ yr)
plot(LakeHuron, ylab = "Depth (ft)", xlab = "Year", las = 1)
abline(lm, col = "blue")
```

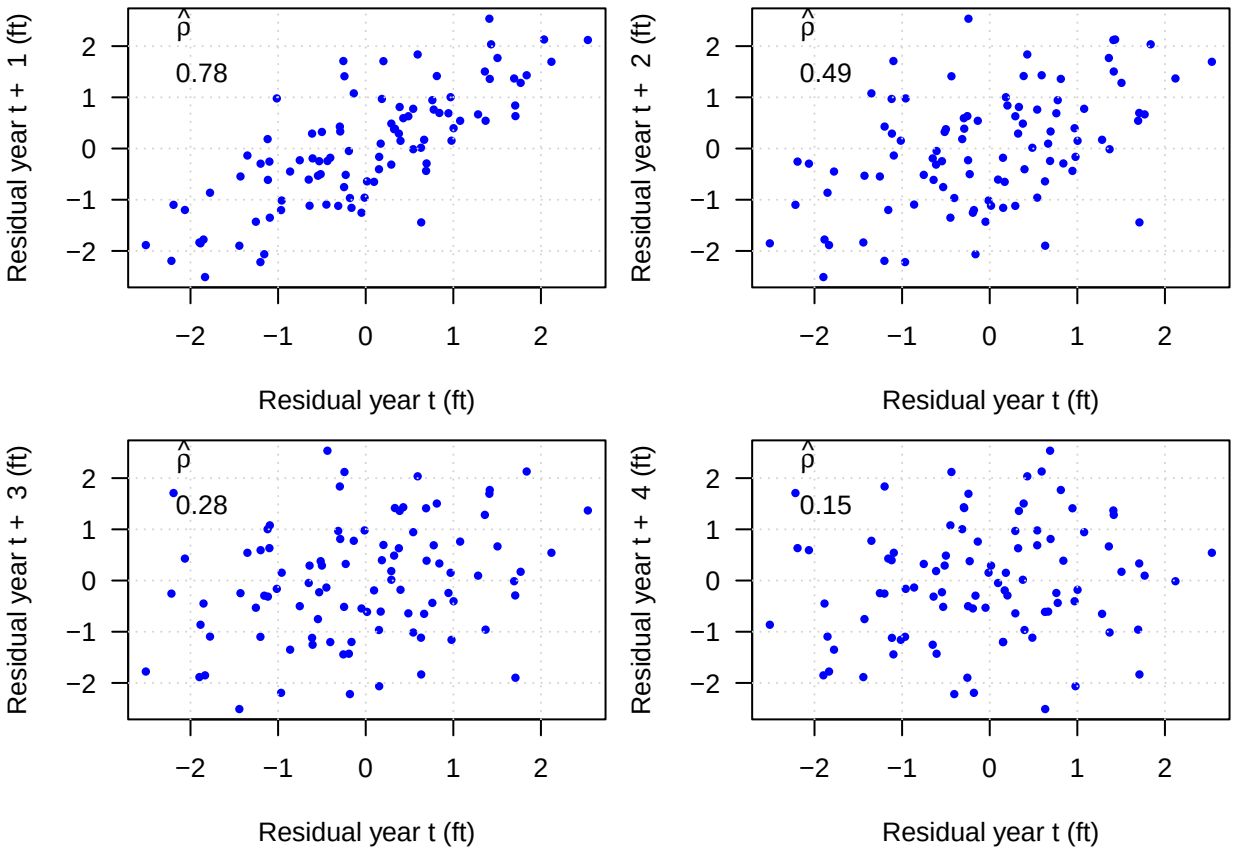



```
lm$residuals <- ts(lm$residuals, start = 1875, end = 1972)
tsplot(lm$residuals, ylab = "Residuals (ft)", xlab = "Year", las = 1)
points(lm$residuals, cex = 0.8, col = "blue", pch = 16)
```



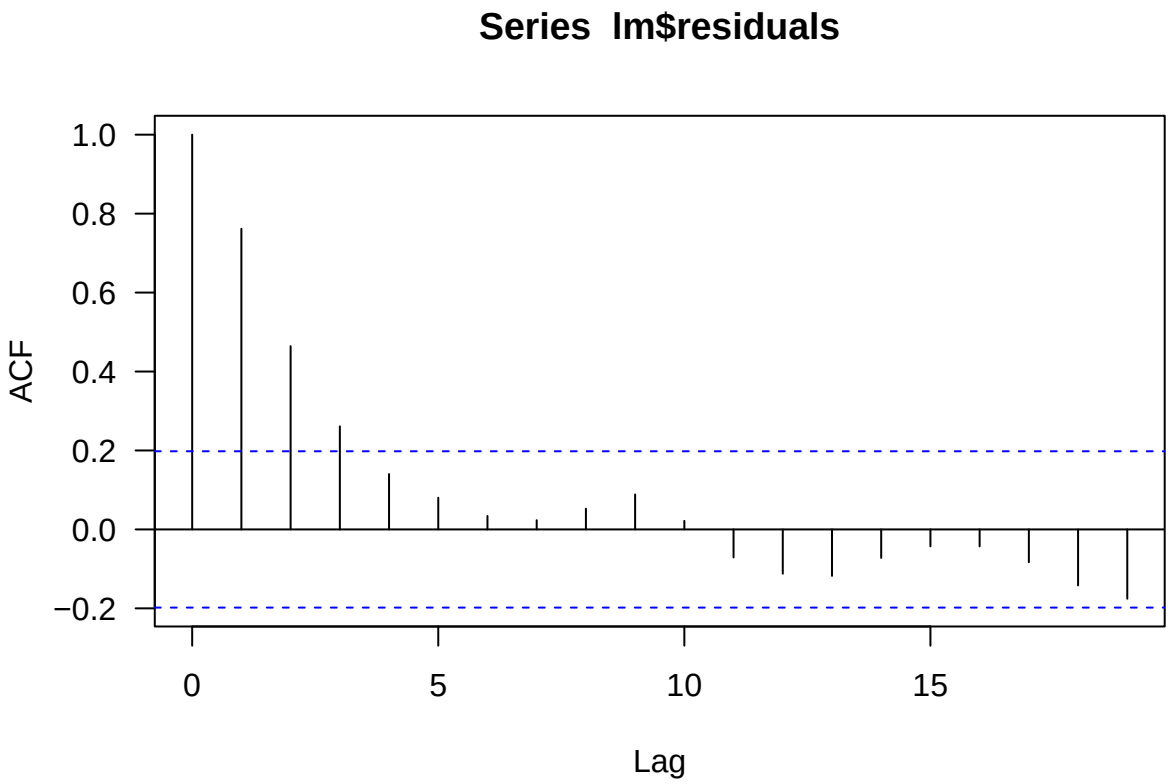
These residual values exhibit a temporal dependence structure, meaning that the nearby (in time) values tend to be more alike than those that are far part. To observe this, let's create a few time lag plots.

```
n <- length(LakeHuron)
h <- 1:4
par(mfrow = c(2, 2), mar = c(4, 4, 0.8, 0.6))
for (i in h){
  plot(lm$residuals[-((n - i + 1):n)], lm$residuals[-(1:i)], pch = 16, col = "blue", cex = 0.7,
       las = 1, xlab = "Residual year t (ft)",
       ylab = paste("Residual year t + ",  h[i], "(ft)"))
  legend("topleft", legend = round(cor(lm$residuals[-((n - i + 1):n)], lm$residuals[-(1:i)]), 2),
         title = expression(hat(rho)), bty = "n")
  grid()
}
```



Later in this course, we will learn how to use the *autocovariance plot* to investigate and model this temporal dependence structure.

```
acf(lm$residuals, las = 1)
```



Time series analysis is the area of statistics that largely deals with analyzing the *dependency* between observations (i.e., $\{\eta_t\}$ in time series data).

Time Series Models

A time series model is a probabilistic model that describes the ways in which the series data $\{y_t\}$ could have been generated. More specifically, a time series model typically takes the form of a probability model for $\{Y_t : t \in T\}$, a collection of random variables indexed in time.

Stationarity

Here, we are faced with a highly challenging task, as we must *estimate* the characteristics of a $|T|-$ dimensional random vector with only a single realization (i.e., one observed value at each time point). Therefore, we will restrict our model class by assuming stationarity. Stationarity implies that certain characteristics of the distribution of a time series does not depend on t, but rather on the *distance* between time points. While most time series are not stationary, one either remove or model the non-stationary parts (e.g., detrend or deseasonalize) to retain only the stationary component (e.g., $\{\eta_t\}$). Additionally, one typically further assume that the process is second order stationary, which means that the mean function (as a function of t) is a constant, and the covaraince function depends solely on the *distance* between time points.

Objectives of time series analysis

Modeling

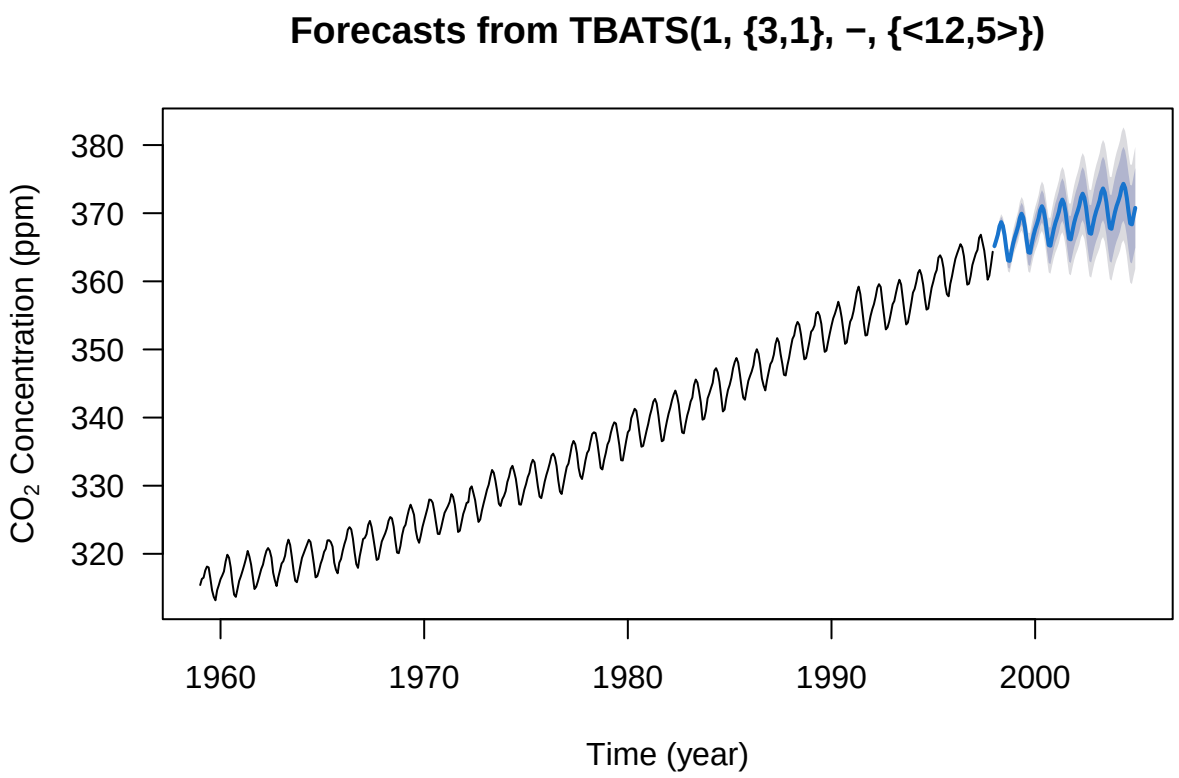
Find a statistical model that adequately explains the observed time series data. For example, identify a model that can account for the correlation between the depths of Lake Huron across years and the presence

of a decreasing long-term trend. The fitted model can be used for further statistical inference, such as addressing questions like *Is there evidence of a decreasing trend in the Lake Huron depths?*

Forecasting

This is perhaps the most common objective. We observe a time series of given length and wish to *predict* or *forecast* future values of the time series based on those we have already observed. A statistical model enables us to make probabilistic forecasts (i.e., provides both point and interval estimates)

```
library(forecast)
TBATSfit <- tbats(co2, use.box.cox = F, use.trend = F, use.damped.trend = F,
                  seasonal.periods = 12)
plot(forecast(TBATSfit, 84), las = 1, xlab = "Time (year)",
     ylab = expression(paste("CO"[2], " Concentration (ppm)")))
```



Adjustment

An example would be seasonal adjustment, where the seasonal component is estimated and then removed to better understand the underlying trend.

Simulation

We can use a time series model (which adequately describes a physical process) as a surrogate to simulate repeatedly, allowing us to approximate how the physical process behaves.

Control

We can adjust various input (control) parameters to make the time series fit more closely to a given standard (many examples come from statistical quality control).

References

Breckling, Jens. 2012. *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*. Vol. 61. Springer Science & Business Media.

Brockwell, Peter J, Peter J Brockwell, Richard A Davis, and Richard A Davis. 2002. *Introduction to Time Series and Forecasting*. Springer.

Huang, Whitney K, Yu-Min Chung, Yu-Bo Wang, Jeff E Mandel, and Hau-Tieng Wu. 2022. "Airflow Recovery from Thoracic and Abdominal Movements Using Synchrosqueezing Transform and Locally Stationary Gaussian Process Regression." *Computational Statistics & Data Analysis* 174: 107384.

Jia, Yisu, Stefanos Kechagias, James Livsey, Robert Lund, and Vladas Pipiras. 2021. "Latent Gaussian Count Time Series." *Journal of the American Statistical Association*, no. just-accepted: 1–28.