STA 209: Introduction to Time Series Analysis

# Homework 1

Rento Saijo

February 11, 2026

## Table of Contents

**Disclosure** **2**

**Problem 1** **3**

Problem 1 (i) . . . . . . . . 4
Problem 1 (ii) . . . . . . . . 5
Problem 1 (iii) . . . . . . . . 7
Problem 1 (a) . . . . . . . . 9
Problem 1 (b) . . . . . . . . 10
Problem 1 (c) . . . . . . . . 12
Problem 1 (d) . . . . . . . . 14

**Problem 2** **17**

Problem 2 (i) . . . . . . . . 17
Problem 2 (ii) . . . . . . . . 21
Problem 2 (iii) . . . . . . . . 22

**Project** **23**

Project (i) . . . . . . . . 23
Project (ii) . . . . . . . . 24
Project (iii) . . . . . . . . 25

# Disclosure

ChatGPT-5.2 was used to create the `YAML` portion and some `LaTeX` code to format the text/equations nicely; some formatting code were also provided by Derin Gezgin. In the setup chunk, libraries were loaded and some helper functions were defined including but not limited to `plot_site_static_ts()` and `plot_site_interactive_ts`. See the original `RMD` file here for more details.

# Problem 1

Problem 1

Consider the Southern oscillation index data (`soi`) available in the R package `astsa`. Use this data to do the following in R and then report your results.

The Southern Oscillation Index (SOI) dataset is a monthly time series with 453 observations covering the years 1950 through 1987. It is represented in the form `Time-Series [1:453]` (notated as running from 1950 to 1988 in the printed format) and is shown as a numeric sequence (e.g., `0.377, 0.246, 0.311, 0.104, -0.016, 0.235, ...`). The series is intended to pair with `rec` (Recruitment) for related analyses. The data were furnished by Dr. Roy Mendelssohn of the Pacific Fisheries Environmental Laboratory, NOAA (personal communication). Demonstrations of `astsa` capabilities are referenced under *FUN WITH ASTSA*, and the most recent version of the package is maintained on GitHub along with the `NEWS` and `ChangeLog` files; additional webpages for the associated texts and help using `R` for time series analysis are available at the author's site.

## Problem 1 (i)

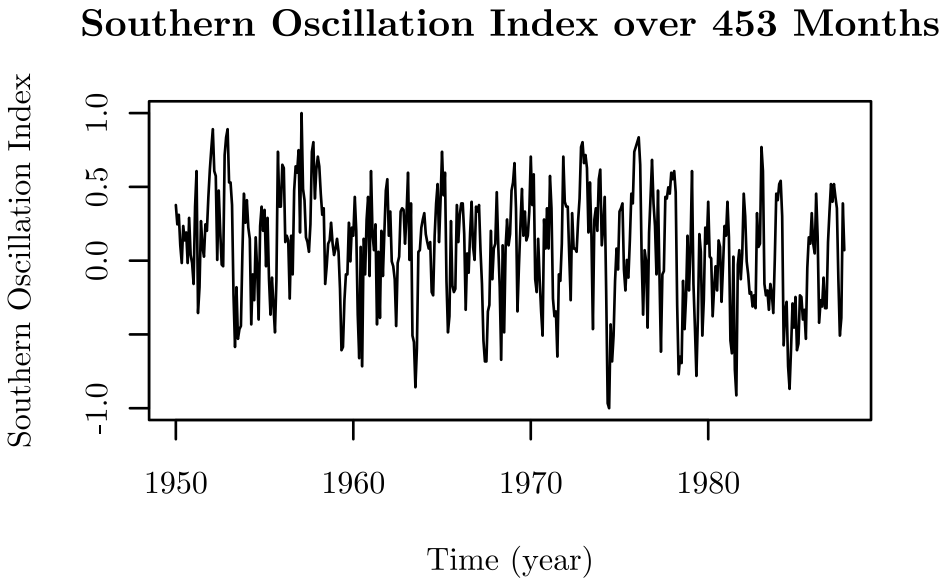
**Problem 1 (i)**

Make a time series plot and comment on the interesting features.

Cell 1

```r
# Load data.
data(soi)

# Make time series plot.
with_family({
  data(soi)
  plot(soi, main = 'Southern Oscillation Index over 453 Months', xlab = 'Time (year)', ylab = 'Southern Oscillation Index')
})
```



From the time series plot of the Southern Oscillation Index (SOI) over 453 months, there is no clear long-run upward or downward trend; instead, the series oscillates around a roughly constant mean near zero, so any trend component appears weak and essentially flat (if anything, slight downward trend). In terms of seasonality, there is a clear repeating pattern at an approximately annual frequency: the SOI exhibits a recurring within-year cycle that repeats from year to year (consistent with a 12-month seasonal structure), even though the strength of the cycle is sometimes masked by the larger irregular swings between positive and negative phases. Regarding variability, the SOI values remain in a relatively stable range of about $-1.0$ to 1.0 (so the overall spread is roughly 2 units), and the amplitude of fluctuations looks fairly constant across time, suggesting approximate homoskedasticity (no clear fanning in/out of variance). Finally, there is no strong evidence of volatility clustering: although there are occasional bursts of larger positive/negative deviations, these do not persist in a sustained way, and the variability returns quickly to its typical level.

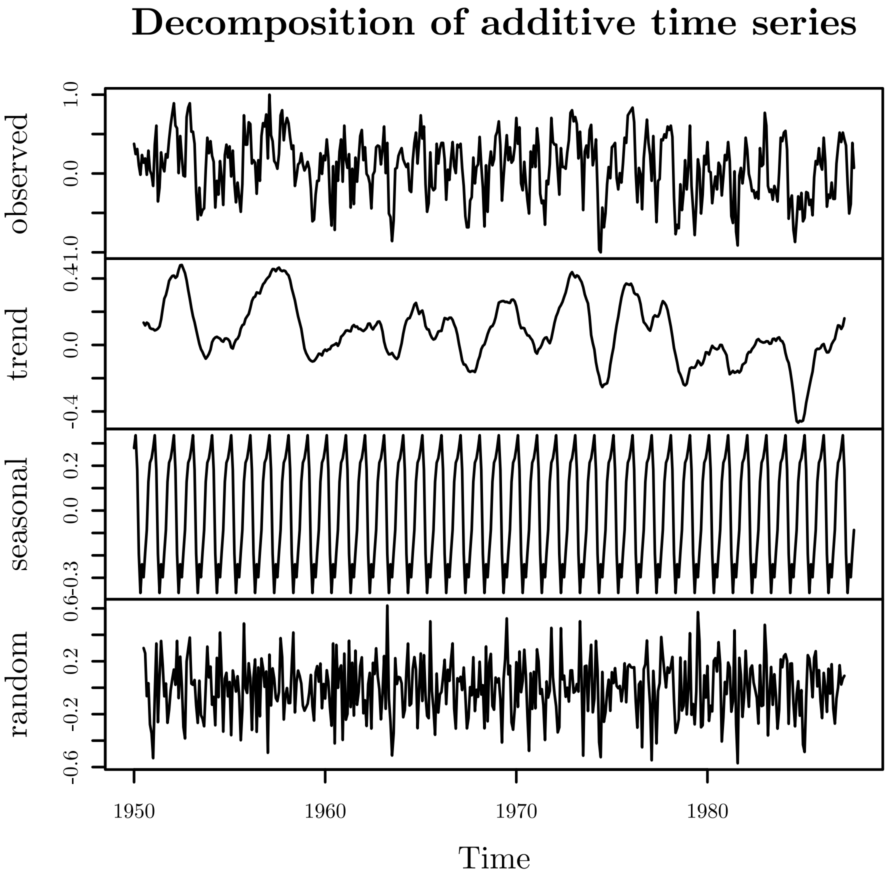
## Problem 1 (ii)

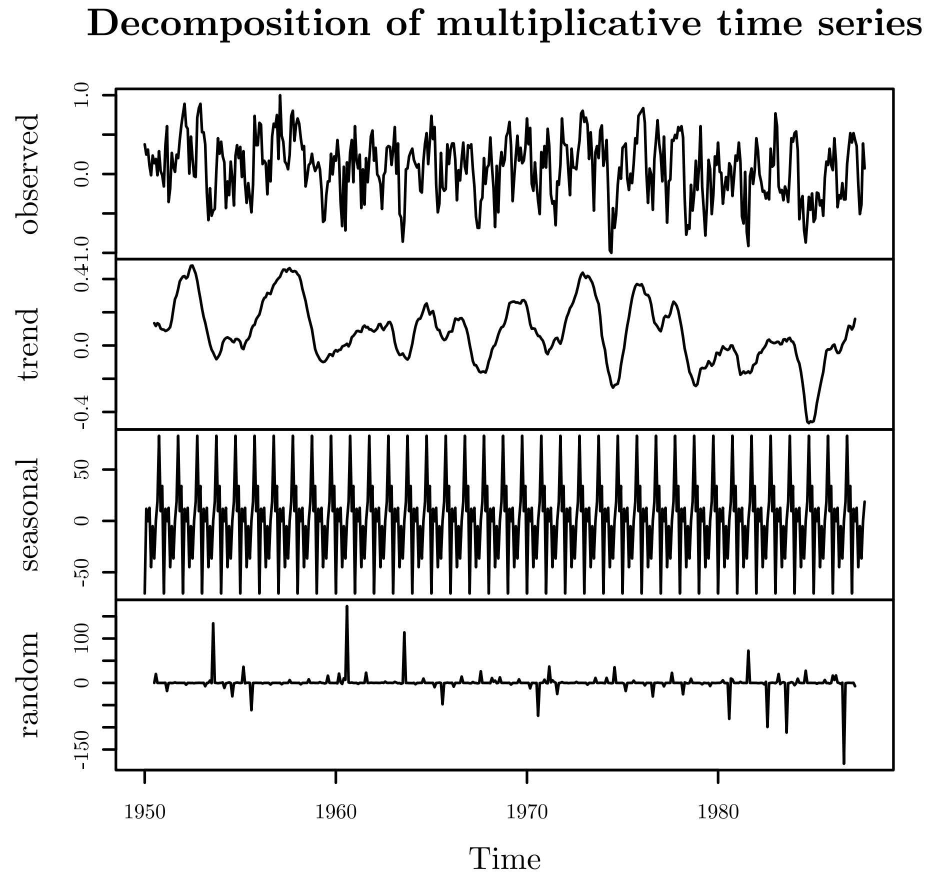
**Problem 1 (ii)**

Show decomposition of this time series and comment on the components.

Cell 2

```
# Make decomposition plots.
with_family({
  plot(stats::decompose(soi, type = 'additive'))
  plot(stats::decompose(soi, type = 'multiplicative'))
})
```





We were instructed in class that no comments were necessary for the decomposition graphs as they'd be a repeat of part (i).

## Problem 1 (iii)

### Problem 1 (iii)

Fit following three different models to the SOI data:

$$\textbf{Model I:} \quad SOI_t = \beta_0 + \beta_1 t + Noise_t$$

$$\textbf{Model II:} \quad SOI_t = \beta_0 + \beta_1 t + \frac{\beta_2 t^2}{2!} + Noise_t$$

$$\textbf{Model III:} \quad SOI_t = \beta_0 + \beta_1 t + \frac{\beta_2 t^2}{2!} + \frac{\beta_3 t^3}{3!} + Noise_t$$

Cell 3

```r
# Fit models.
t1 <- stats::time(soi)
t2 <- t1^2 / factorial(2)
t3 <- t1^3 / factorial(3)
m1 <- lm(soi ~ t1)
m2 <- lm(soi ~ t1 + t2)
m3 <- lm(soi ~ t1 + t2 + t3)
summary(m1)
```

```
##
## Call:
## lm(formula = soi ~ t1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04140 -0.24183  0.01935  0.27727  0.83866
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 13.70367    3.18873   4.298 0.0000212 ***
## t1          -0.00692    0.00162  -4.272 0.0000236 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3756 on 451 degrees of freedom
## Multiple R-squared:  0.0389, Adjusted R-squared:  0.03677
## F-statistic: 18.25 on 1 and 451 DF,  p-value: 0.00002359
```

Cell 4

```r
summary(m2)
```

```
##
## Call:
## lm(formula = soi ~ t1 + t2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0529 -0.2443  0.0129  0.2658  0.8412
##
## Coefficients:
##                 Estimate   Std. Error t value Pr(>|t|)
## (Intercept) -496.7608959  644.3532809  -0.771    0.441
## t1             0.5116415    0.6545674   0.782    0.435
## t2            -0.0002634    0.0003325  -0.792    0.429
##
## Residual standard error: 0.3758 on 450 degrees of freedom
## Multiple R-squared:  0.04024,    Adjusted R-squared:  0.03597
## F-statistic: 9.433 on 2 and 450 DF,  p-value: 0.00009698
```

Cell 5

```r
summary(m3)
```

```
##
## Call:
## lm(formula = soi ~ t1 + t2 + t3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.07047 -0.23026  0.01631  0.27232  0.85650
##
## Coefficients:
##                  Estimate     Std. Error t value Pr(>|t|)
## (Intercept) 130775.1717210 132546.0272739   0.987    0.324
## t1            -199.5206669    201.9724492  -0.988    0.324
## t2               0.2029392      0.2051726   0.989    0.323
## t3              -0.0001032      0.0001042  -0.990    0.323
##
## Residual standard error: 0.3758 on 449 degrees of freedom
## Multiple R-squared:  0.04233,    Adjusted R-squared:  0.03593
## F-statistic: 6.616 on 3 and 449 DF,  p-value: 0.0002215
```

Using $t$ to denote the time index (in the same units returned by `time(soi)`), the fitted regression models were:

$$\begin{aligned}
\widehat{\textbf{Model}}\textbf{ I:}\quad & \widehat{SOI}_t = 13.70367 \;-\; 0.00692\,t,\\
\widehat{\textbf{Model}}\textbf{ II:}\quad & \widehat{SOI}_t = -496.76090 \;+\; 0.51164\,t \;-\; 0.0002634\,\frac{t^2}{2!},\\
\widehat{\textbf{Model}}\textbf{ III:}\quad & \widehat{SOI}_t = 130775.17172 \;-\; 199.52067\,t \;+\; 0.20294\,\frac{t^2}{2!} \;-\; 0.0001032\,\frac{t^3}{3!}.
\end{aligned}$$

The adjusted $R^2$ values are small in all cases (Model I: 0.03677; Model II: 0.03597; Model III: 0.03593), indicating that time (and higher-order time polynomials) explains only a small fraction of the variation in SOI. The residual standard error is essentially unchanged across models as well (Model I: 0.3756; Model II: 0.3758; Model III: 0.3758), meaning the typical size of the residual fluctuations around the fitted trend is about 0.376 SOI units regardless of whether we include quadratic or cubic terms. Overall, adding the higher-order terms provides negligible improvement after accounting for the extra parameters.

## Problem 1 (a)

Problem 1 (a)

How many regression parameters are present in each of these models?

Each model is a linear regression with an intercept plus one coefficient for each time term:

- **Model I** has **2** regression parameters: $\beta_0, \beta_1$.
- **Model II** has **3** regression parameters: $\beta_0, \beta_1, \beta_2$.
- **Model III** has **4** regression parameters: $\beta_0, \beta_1, \beta_2, \beta_3$.

## Problem 1 (b)

**Problem 1 (b)**

Report ANOVA for all three models.

Cell 6

```r
# Check ANOVA.
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: soi
##            Df Sum Sq Mean Sq F value     Pr(>F)
## t1          1  2.576 2.57583  18.254 0.00002359 ***
## Residuals 451 63.640 0.14111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cell 7

```r
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: soi
##            Df Sum Sq Mean Sq F value     Pr(>F)
## t1          1  2.576 2.57583 18.2392 0.00002378 ***
## t2          1  0.089 0.08863  0.6276     0.4286
## Residuals 450 63.551 0.14122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cell 8

```r
anova(m3)
```

```
## Analysis of Variance Table
##
## Response: soi
##            Df Sum Sq Mean Sq F value    Pr(>F)
## t1          1  2.576 2.57583 18.2384 0.0000238 ***
## t2          1  0.089 0.08863  0.6276    0.4287
## t3          1  0.139 0.13853  0.9809    0.3225
## Residuals 449 63.413 0.14123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### Model 1

| Source | Df | SS | MS | F-stat | P-value |
|---|---|---|---|---|---|
| Regression | 1 | 2.576 | 2.57583 | 18.254 | 0.00002359 |
| Error | 451 | 63.640 | 0.14111 | | |
| Total | 452 | 66.216 | 0.14650 | | |

### Model 2

| Source | Df | SS | MS | F-stat | P-value |
|---|---|---|---|---|---|
| Regression | 2 | 2.665 | 1.33223 | 9.433 | 0.00009698 |
| Error | 450 | 63.551 | 0.14122 | | |
| Total | 452 | 66.216 | 0.14650 | | |

### Model 3

| Source | Df | SS | MS | F-stat | P-value |
|---|---|---|---|---|---|
| Regression | 3 | 2.804 | 0.93433 | 6.616 | 0.0002215 |
| Error | 449 | 63.413 | 0.14123 | | |
| Total | 452 | 66.216 | 0.14650 | | |

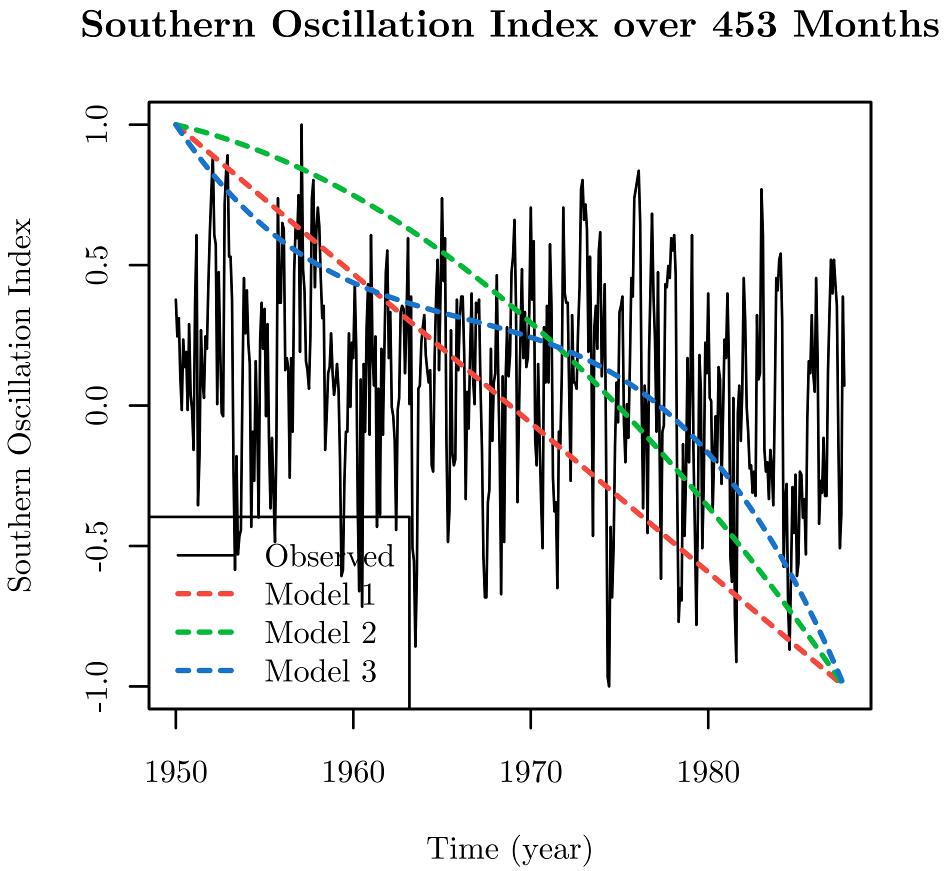
## Problem 1 (c)

**Problem 1 (c)**

Plot all three fits on the same plot and discuss which one fits SOI data the best.

Cell 9

```r
# Plot fitted lines.
with_family({
  plot(soi, main = 'Southern Oscillation Index over 453 Months', xlab = 'Time (year)', ylab = 'Southern Oscillation Index')
  par(new = TRUE); plot(m1$fitted, type = 'l', lwd = 2, lty = 2, col = 2, main = '', xlab = '', ylab = '', xaxt = 'n', yaxt = 'n')
  par(new = TRUE); plot(m2$fitted, type = 'l', lwd = 2, lty = 2, col = 3, main = '', xlab = '', ylab = '', xaxt = 'n', yaxt = 'n')
  par(new = TRUE); plot(m3$fitted, type = 'l', lwd = 2, lty = 2, col = 4, main = '', xlab = '', ylab = '', xaxt = 'n', yaxt = 'n')
  legend('bottomleft', c('Observed', 'Model 1', 'Model 2', 'Model 3'), lty = c(1, 2, 2, 2), lwd = c(1, rep(2, 3)), col = 1:4)
})
```



From the fitted-lines plot, all three trend models capture only a very smooth long-run component, while the observed SOI series fluctuates substantially around that trend. Model I imposes a straight (linear) downward trend, whereas Models II and III allow increasing curvature; visually, the higher-order models can appear to follow the broad shape slightly more in some sub-intervals, but they still miss the dominant month-to-month variability. The numerical fit measures confirm that this extra flexibility does not translate into better overall fit: Model I has the largest adjusted $R^2$ (0.03677) compared with Model II (0.03597) and Model III (0.03593), and the residual standard errors are essentially the same for all three models (about 0.376). Thus, despite the added polynomial terms, Models II and III do not provide a meaningful improvement over the simpler linear trend model, and Model I is preferred on both adjusted $R^2$ and simplicity (although it may seem somewhat counterintuitive to what the graph may suggest; to be fair, we are splitting hairs at this point).

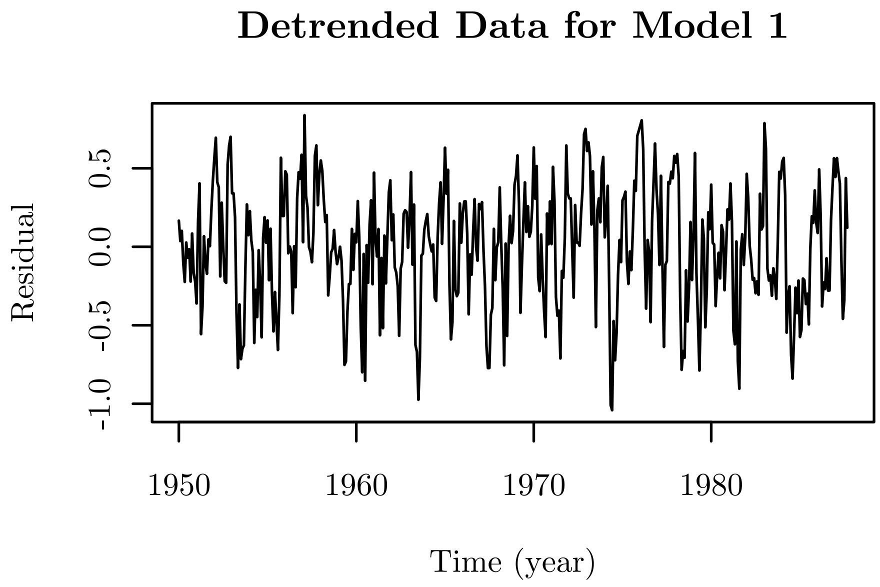
## Problem 1 (d)

**Problem 1 (d)**

Plot detrended data from all three on separate plots, and compare residuals to discuss which model did better.

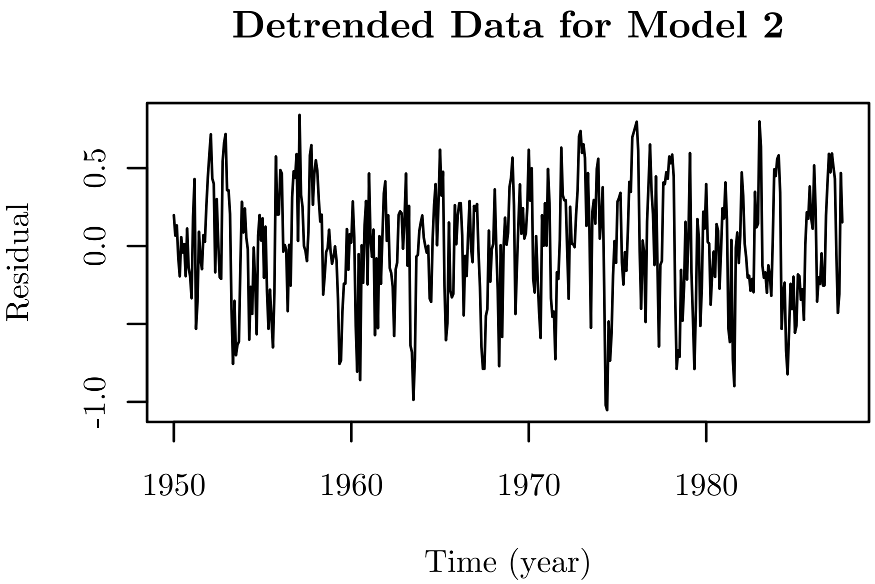
Cell 10

```r
# Plot detrended data for model 1.
with_family({
  plot(time(soi), m1$residuals, type = 'l',
       main = 'Detrended Data for Model 1',
       xlab = 'Time (year)', ylab = 'Residual')
})
```



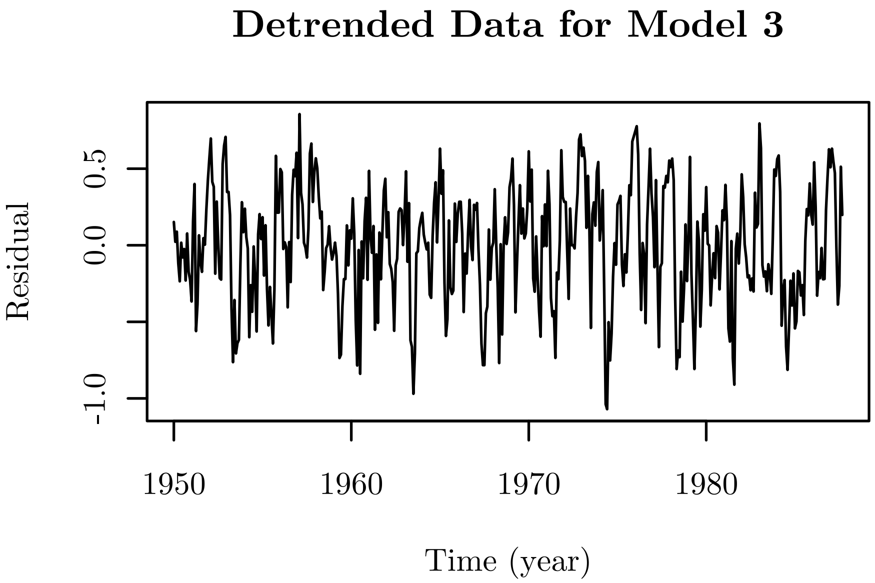
Cell 11

```r
# Plot detrended data for model 2.
with_family({
  plot(time(soi), m2$residuals, type = 'l',
       main = 'Detrended Data for Model 2',
       xlab = 'Time (year)', ylab = 'Residual')
})
```



Cell 12

```r
# Plot detrended data for model 3.
with_family({
  plot(time(soi), m3$residuals, type = 'l',
       main = 'Detrended Data for Model 3',
       xlab = 'Time (year)', ylab = 'Residual')
})
```



The detrended series (residuals) from all three models look very similar: each fluctuates around zero with roughly constant spread over time, and none of the three residual plots shows a dramatic reduction in structure compared to the others. This matches the near-identical residual standard errors (about 0.376 for all three models) and the fact that adjusted $R^2$ does not improve when moving from Model I to Models II and III. Overall, Model I performs at least as well as the more complex models, and the residual plots support choosing the simplest trend specification among the three.

# Problem 2

PROBLEM 2

Consider the Northeast temperature case study.

## Problem 2 (i)

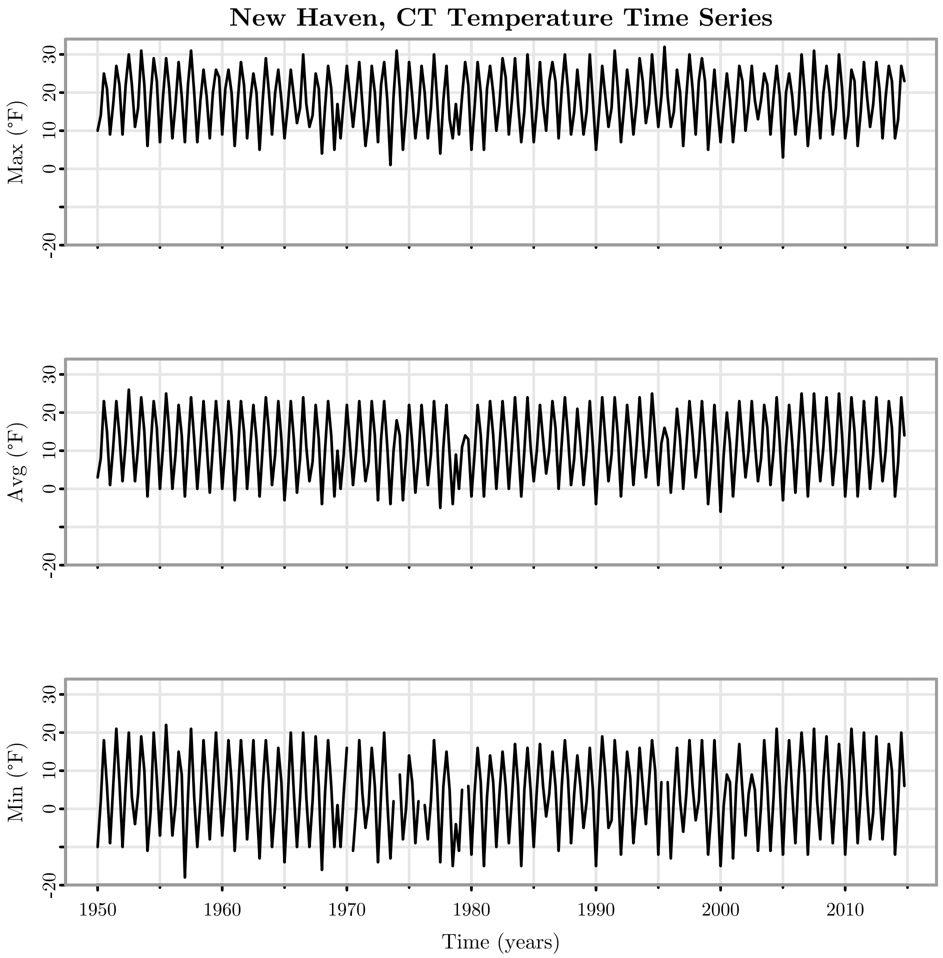
PROBLEM 2 (I)

Report time series plots for three sites (data in Moodle). Include plots (multiple in one plot) for max, min, and avg. temperature for each site.

Cell 13

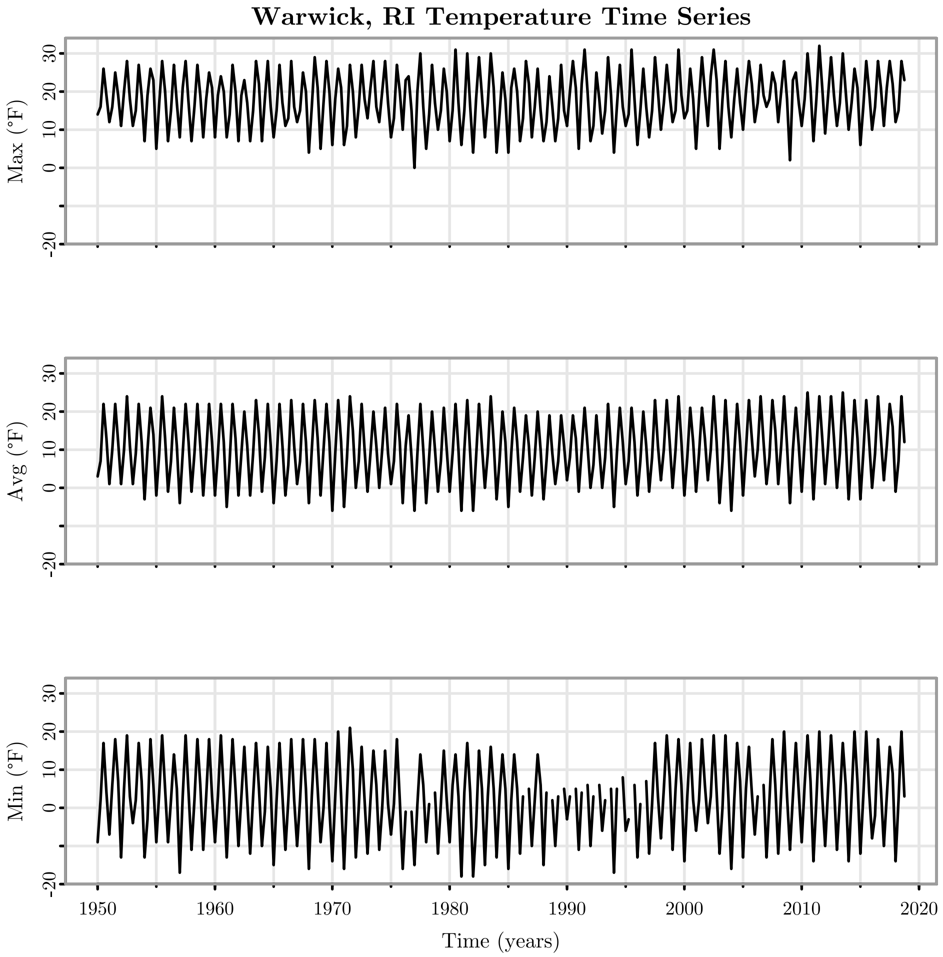
```r
# Load data.
NewHaven  <- ts_seasonal('data/Climate_Northeast_NewHavenCT.xlsx')
Warwick   <- ts_seasonal('data/Climate_Northeast_WarwickRI.xlsx')
Worcester <- ts_seasonal('data/Climate_Northeast_WorcesterMA.xlsx')

# Make static time series plot for New Haven.
with_family({
  par(mfrow = c(3, 1), mar = c(2, 4, 2, 1))
  ylim <- c(min(make_site_ts(NewHaven)$ts_min, make_site_ts(NewHaven)$ts_avg, make_site_ts(NewHaven)$ts_max, na.rm = TRUE), max(make_site_ts(NewHaven)$ts_min, make_site_ts(NewHaven)$ts_avg, make_site_ts(NewHaven)$ts_max, na.rm = TRUE))
  plot_site_static_ts(NewHaven, 'New Haven, CT', which = 'max', show_x = FALSE, show_title = TRUE, ylim = ylim)
  plot_site_static_ts(NewHaven, 'New Haven, CT', which = 'avg', show_x = FALSE, show_title = FALSE, ylim = ylim)
  plot_site_static_ts(NewHaven, 'New Haven, CT', which = 'min', show_x = TRUE, show_title = FALSE, ylim = ylim)
})
```



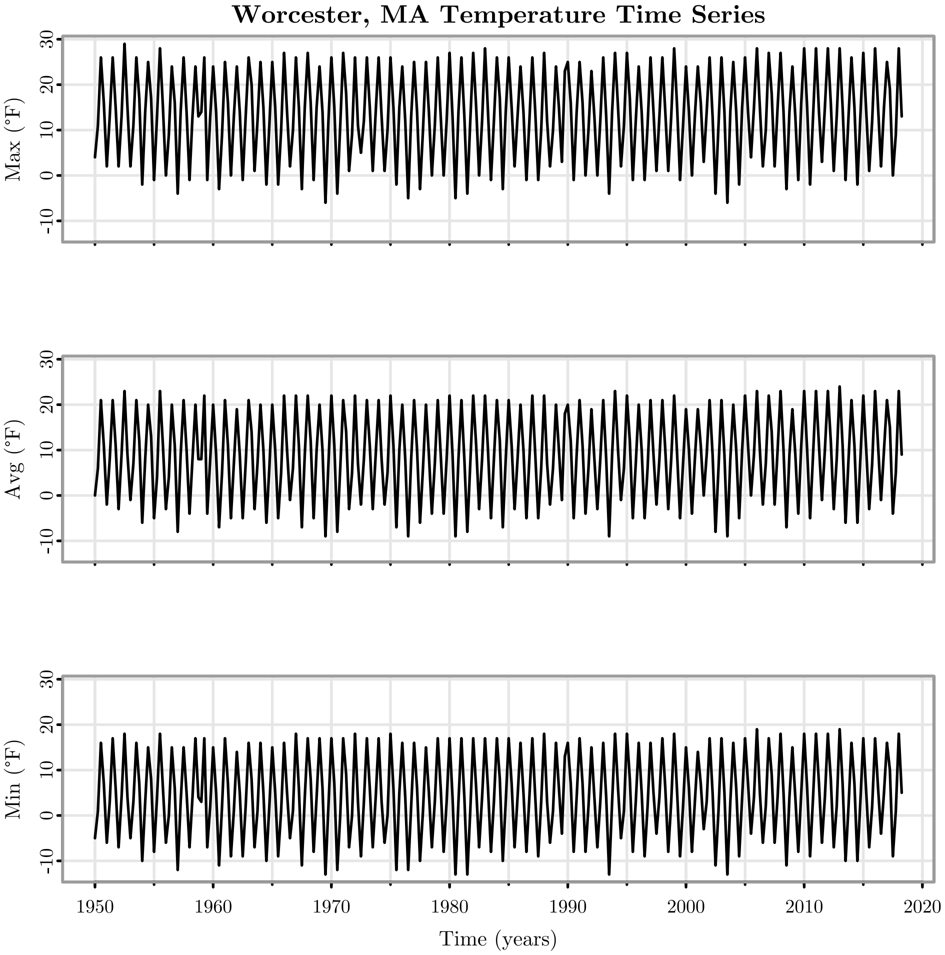
Cell 14

```
# Make static time series plot for Warwick.
with_family({
  par(mfrow = c(3, 1), mar = c(2, 4, 2, 1))
  ts_list <- make_site_ts(Warwick)
  ylim <- c(min(ts_list$ts_min, ts_list$ts_avg, ts_list$ts_max, na.rm = TRUE),
max(ts_list$ts_min, ts_list$ts_avg, ts_list$ts_max, na.rm = TRUE))
  plot_site_static_ts(Warwick, 'Warwick, RI', which = 'max', show_x = FALSE,
  show_title = TRUE, ylim = ylim)
  plot_site_static_ts(Warwick, 'Warwick, RI', which = 'avg', show_x = FALSE,
  show_title = FALSE, ylim = ylim)
  plot_site_static_ts(Warwick, 'Warwick, RI', which = 'min', show_x = TRUE,
  show_title = FALSE, ylim = ylim)
})
```



Cell 15

```
# Make static time series plot for Worcester.
with_family({
  par(mfrow = c(3, 1), mar = c(2, 4, 2, 1))
  ts_list <- make_site_ts(Worcester)
  ylim <- c(min(ts_list$ts_min, ts_list$ts_avg, ts_list$ts_max, na.rm = TRUE),
max(ts_list$ts_min, ts_list$ts_avg, ts_list$ts_max, na.rm = TRUE))
  plot_site_static_ts(Worcester, 'Worcester, MA', which = 'max', show_x = FALSE,
  show_title = TRUE, ylim = ylim)
  plot_site_static_ts(Worcester, 'Worcester, MA', which = 'avg', show_x = FALSE,
  show_title = FALSE, ylim = ylim)
  plot_site_static_ts(Worcester, 'Worcester, MA', which = 'min', show_x = TRUE,
  show_title = FALSE, ylim = ylim)
})
```


Worcester, MA Temperature Time Series
Max (°F)
Avg (°F)
Min (°F)
-10
0
10
20
30
1950
1960
1970
1980
1990
2000
2010
2020
Time (years)

## Problem 2 (ii)

Problem 2 (ii)

Make interactive time series plots and share links for interactive plots.

Cell 16

```r
# Make interactive plots.
p_NewHaven <- plot_site_interactive_ts(NewHaven, 'New Haven, CT')
p_Warwick <- plot_site_interactive_ts(Warwick, 'Warwick, RI')
p_Worcester <- plot_site_interactive_ts(Worcester, 'Worcester, MA')
dir_create('plots')
htmlwidgets::saveWidget(p_NewHaven, 'plots/NewHaven.html', selfcontained = TRUE)
htmlwidgets::saveWidget(p_Warwick, 'plots/Warwick.html', selfcontained = TRUE)
htmlwidgets::saveWidget(p_Worcester, 'plots/Worcester.html', selfcontained = TRUE)
```

Here are the interactive plots for New Haven, Warwick, and Worcester.

## Problem 2 (iii)

Problem 2 (iii)

Compare the key features of temperature time series for your sites.

All three sites (New Haven, Warwick, and Worcester) show a very strong and highly regular seasonal pattern, with temperatures rising in the warm season and falling in the cold season each year; this seasonality is present in the Max, Avg, and Min series and is the dominant visible feature across the entire sample. In terms of long-run trend, none of the sites shows an obvious dramatic monotone change, although the centers of several of the series appear at most to drift slightly upward over the decades (subtle relative to the seasonal swings and really if you stare it hard enough xD, but largely no movement). The main differences across sites are in level and variability: Worcester is generally colder than the coastal sites, with noticeably lower winter lows (Min dips further below 0°F) and a wider overall annual range, while New Haven and Warwick tend to have milder minima and slightly higher baselines consistent with a more coastal climate. Variability looks broadly stable over time for New Haven and Worcester (no clear fanning in/out of the seasonal amplitude), whereas Warwick shows an evident mid-sample disruption in the Min series where the seasonal swings compress and the level shifts for an extended period before returning to the earlier seasonal behavior; aside from that segment, the variability is fairly consistent and there is no strong sign of persistent volatility clustering in any of the three sites beyond occasional isolated extreme seasons.

# Project

## Project (i)

### Project (i)

Discuss possible time series topics you are interested in.

I am interested in time series problems in hockey analytics where player performance and team context evolve over repeated games. One topic is understanding whether apparent hot streaks and slumps reflect real changes in underlying performance or are mostly random variation, and how to summarize short-term form in a statistically sound way. Another topic is how offensive and defensive impact measures change over time, especially when distinguishing noisy outcomes (goals, points) from process-based measures such as expected goals. I am also interested in identifying meaningful shifts in a player's role or effectiveness over a season or across seasons (for example, changes that may coincide with usage, linemates, or special-teams opportunities), and in forecasting near-future game performance using information from recent games while accounting for the fact that consecutive games are not independent.

## Project (ii)

Project (ii)

Share potential inquiry questions.

Using the per-game log as an equally spaced time series (one observation per game), I would like to investigate the following inquiry questions. First, do Martin Nečas's underlying performance measures show persistent short-term dependence from game to game (i.e., are strong games more likely to be followed by strong games), or do they behave approximately like independent noise around a stable level? Second, are there identifiable stretches of games where his baseline level appears to shift (for example, sustained increases or decreases in individual expected goals, on-ice expected goals for/against, or their derived differentials), suggesting changes in role, effectiveness, or context? Third, how stable are "recent-form" summaries: over what window of recent games do rolling averages become informative while still responding quickly to real changes? Fourth, can we forecast next-game performance metrics (such as individual expected goals or on-ice expected goal differential) better than a simple long-run average by incorporating recent game history, and if so, how much improvement do we get in predictive accuracy? Finally, how does variability change over time: are there periods where game-to-game fluctuations become noticeably larger or smaller, and do those periods align with changes in usage or season-to-season transitions?

## Project (iii)

**Project (iii)**

Discuss potential datasets that could be used for the project.

Cell 17

```r
# Aggregate data.
game_logs <- data.frame()
seasonIds <- nhlscraper::player_seasons(player = 8480039) %>%
  dplyr::pull(seasonId)
for (seasonId in seasonIds) {
  game_logs <- dplyr::bind_rows(game_logs, nhlscraper::player_game_log(player = 8480039, season = seasonId, game_type = 2))
}
game_logs <- game_logs %>%
    dplyr::arrange(gameId) %>%
    dplyr::select(gameId, points, goals, assists, plusMinus, shots, shifts, toi)

# Display tail.
table_latex(game_logs %>% slice_tail(n = 5), c('Game ID', 'Points', 'Goals', 'Assists', '+/-', 'Shots', 'Shifts', 'Time on Ice'))
```

| Game ID | Points | Goals | Assists | +/- | Shots | Shifts | Time on Ice |
|---|---|---|---|---|---|---|---|
| 2025020789 | 0 | 0 | 0 | -1 | 1 | 27 | 27:03 |
| 2025020805 | 2 | 0 | 2 | 0 | 9 | 22 | 25:16 |
| 2025020819 | 0 | 0 | 0 | 1 | 2 | 22 | 20:57 |
| 2025020841 | 0 | 0 | 0 | -3 | 0 | 25 | 22:37 |
| 2025020845 | 0 | 0 | 0 | -3 | 3 | 20 | 20:57 |

A primary dataset for this project is the aggregated NHL game log data for Martin Nečas (playerId 8480039), collected across seasons using `nhlscraper` and combined into a single time-ordered table with 493 observations (games). The dataset includes per-game outcomes such as goals, assists, points, plus/minus, power-play production, shots, penalty minutes, shifts, and time-on-ice, along with contextual fields such as home/road and opponent identifiers. Because the unit of observation is a game, an equidistant time variable can be defined as game number (1 through 493), which makes standard time series methods directly applicable without irregular spacing. In addition, I can augment each game with expected-goals features derived from my existing fine-tuned LightGBM xG model, including individual xG and on-ice xG for and against (and functions of these such as xG differential, share, percentage, and rates per time-on-ice). This augmentation would allow me to model a cleaner underlying performance signal than raw goals/points, since xG is designed to reduce outcome noise. If needed, further enrichment could include schedule-derived variables (days of rest, back-to-back indicator, travel proxies), and opponent strength proxies, which would help separate true player-level dynamics from changing context.