Name: Your name here
Due: 2024/09/25

Day 4 - Lab 1: Decomposition of time series

Introduction

The purpose of today's lecture is to understand how to further explore how to decompose a time series into its constituent components in R. To guide our exploration, we will again return to the Vermont temperatures data set.

```
# packages
library(tidyverse)
library(lubridate)

# load data
vt_temps <- readr::read_csv("vt_temps.csv")
```

---

0. [1 pt] With whom are you working today?

1. [14 pt] One (final?) time, we return to the Vermont temperatures data set. The purpose of this question is to reconstruct the decomposition of an additive time series by hand.

    a) [1 pt] Once again, create a `ts` object, called `vt_ts`, of the Vermont temperatures time series spanning 1900-01 to 2000-12.

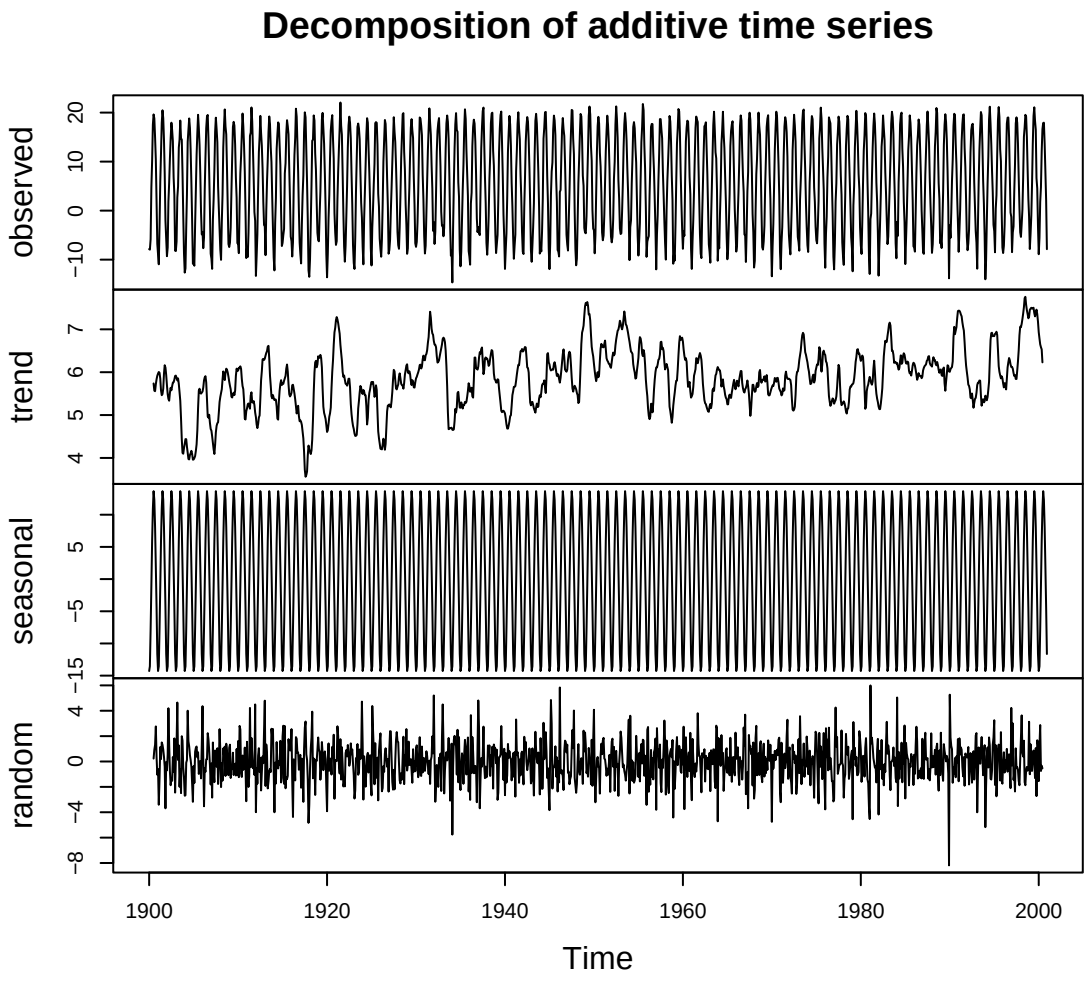
```
vt_ts_long <- ts(
  vt_temps$AverageTemperature,
  start = c(1850, 1),
  end = c(2013, 9),
  freq = 12
)

vt_ts <- window(
  vt_ts_long,
  start = c(1900, 1),
  end = c(2000, 12)
```

```
)
```

b) [2 pt] Plot the decomposition of the `vt_ts` object. Is there clear evidence of a trend or seasonal component?

```
vt_decompose <- decompose(vt_ts)
plot(vt_decompose)
```



The series is almost entirely driven by the seasonal component. There is perhaps a slight, meandering positive trend, but it is minimal if present at all.

c) [2 pt] Calculate the trend component for 1850-07 to 1851-07 **by hand**. Verify you have correctly calculated the trend component by printing out a table of the trend component calculated by hand and via the decompose function.

> **!** Hint
>
> To calculate this moving average by hand, consider using a **for loop**. Additionally, it is helpful to know that 1850-07 corresponds to $t = 7$ and 1851-07 corresponds to $t = 18$. Skeleton code to calculate the moving average is provided below.

```
# create a storage vector of length length(7:18)
ma_byhand <- matrix(NA, nrow = 12, ncol = 1)
for(t in 7:18){
  ma_byhand[t-6,] <- (
    .5*vt_ts[t-6] + vt_ts[t-5] + vt_ts[t-4] + vt_ts[t-3] +
      vt_ts[t-2] + vt_ts[t-1] + vt_ts[t] + vt_ts[t+1] +
      vt_ts[t+2] + vt_ts[t+3] + vt_ts[t+4] + vt_ts[t+5] + .5*vt_ts[t+6]
  )*(1/12)
}

knitr::kable(tibble(byhand = ma_byhand, withr = vt_decompose$trend[7:18]))
```

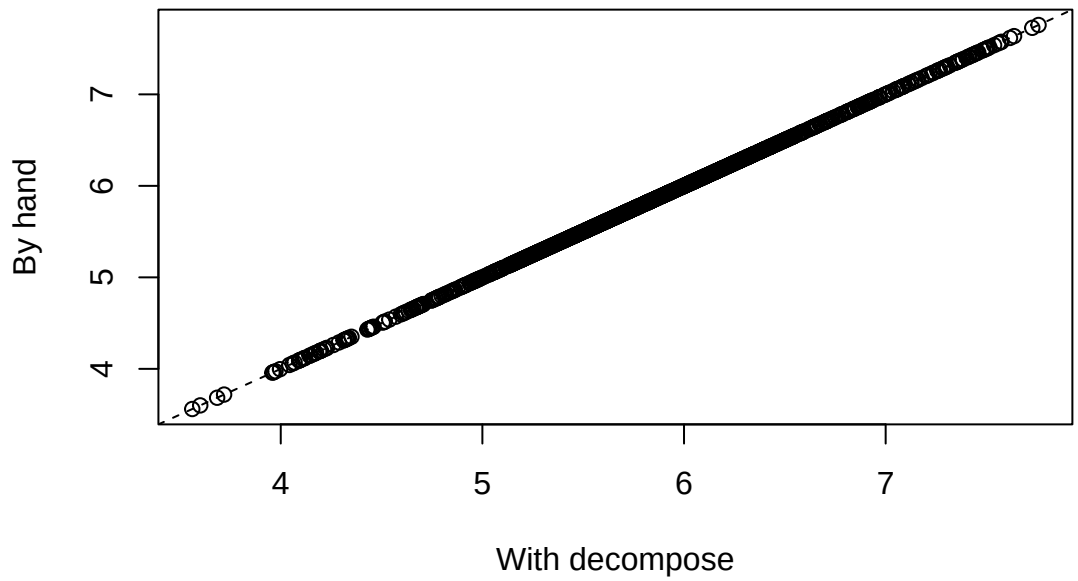
byhand	withr
5.737583	5.737583
5.568917	5.568917
5.556333	5.556333
5.719125	5.719125
5.836083	5.836083
5.917208	5.917208
5.967583	5.967583
6.005292	6.005292
5.985375	5.985375
5.848833	5.848833
5.577833	5.577833
5.463458	5.463458

d) [1 pt] Repeat this process for the entire time series. Plot the hand-calculated trend by the trend calculated by `decompose`. You should see a perfect 1-1 relationship.

```
vt_trend <- matrix(NA, nrow = length(vt_ts) - 12, ncol = 1)
for(t in 7:(length(vt_ts)-6)){
  vt_trend[t-6,] <- (
    .5*vt_ts[t-6] + vt_ts[t-5] + vt_ts[t-4] + vt_ts[t-3] +
      vt_ts[t-2] + vt_ts[t-1] + vt_ts[t] + vt_ts[t+1] +
      vt_ts[t+2] + vt_ts[t+3] + vt_ts[t+4] + vt_ts[t+5] + .5*vt_ts[t+6]
  )*(1/12)
}

# append the NAs we skipped
vt_trend_full <- c(rep(NA, 6), vt_trend, rep(NA, 6))

# plot
plot(
  vt_trend_full ~ vt_decompose$trend,
  xlab = "With decompose",
  ylab = "By hand"
)
abline(a = 0, b = 1, lty = 2)
```



e) [2 pt] Next, calculate the seasonal component associated with each time point t in $[7, 1206]$ component for the entire time series **by hand**. The skeleton code below should help. Print the first 6 rows of the resulting matrix of seasonal effects.

```
# calculate the difference between the raw ts and the calculated trend
seasonal_differences <- c(vt_ts - vt_trend_full)

# coerce into a matrix and print
seasonal_differences_mat <- matrix(seasonal_differences, ncol = 12, byrow = T)
round(seasonal_differences_mat [1:6,], 2)
```

```
      [,1]   [,2]  [,3]  [,4] [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11]  [,12]
[1,]    NA     NA    NA    NA   NA    NA 13.89 13.25  9.38  4.66 -4.12 -12.69
[2,] -14.88 -16.96 -9.14  0.10 6.08 12.43 14.99 13.26  8.43  1.48 -8.16 -11.68
[3,] -14.99 -12.99 -3.58  0.33 5.11  9.05 12.37 10.80  8.42  1.08 -2.65 -14.09
[4,] -14.17 -12.18 -3.13 -0.93 7.16  8.93 13.05  9.83 10.06  3.54 -4.93 -12.90
[5,] -16.99 -16.21 -7.87 -0.93 9.90 12.52 14.85 12.81  8.19  1.46 -6.25 -14.82
[6,] -14.65 -15.24 -7.00  0.03 7.08 11.26 14.40 11.34  7.78  1.94 -5.80 -10.38
```

f) [2 pt] Calculate the average seasonal effect associated with each month, and compare these 12 values to those obtained from the `decompose` function. Are they the same?

```
# create a storage vector of length length(7:18)
seasonal_means <- colMeans(seasonal_differences_mat, na.rm = T)

knitr::kable(tibble(byhand = seasonal_means, withr = vt_decompose$figure))
```

byhand	withr
-14.283539	-14.287062
-13.543696	-13.547219
-7.769890	-7.773412
-1.045978	-1.049501
5.897694	5.894172
10.929412	10.925890
13.667832	13.664309
12.323743	12.320221
8.069445	8.065923
1.894978	1.891456
-4.463157	-4.466679
-11.634576	-11.638098

No, but they are *very* close...

g) [1 pt] What is the mean of the current estimated seasonal effects?

```
mean(seasonal_means)
```

```
[1] 0.003522361
```

> **Note**
>
> It is typical to force the mean of the seasonal effects to equal exactly 0, a process called **centering**.
>
> For additive models, seasonal effects are centered by subtracting the mean of all the average seasonal effects from each average seasonal effect. For multiplicative models, seasonal effects are centered by dividing each seasonal mean by the mean of all the seasonal means. (You may need to read this paragraph a few times)

h) [1 pt] Center the seasonal effects. Compare the newly centered seasonal means to the values obtained from `decompose`. Are they the same now?

```
seasonal_means_centered <- seasonal_means - mean(seasonal_means)
knitr::kable(tibble(byhand = seasonal_means_centered, withr = vt_decompose$figure))
```

byhand	withr
-14.287062	-14.287062
-13.547219	-13.547219
-7.773412	-7.773412
-1.049501	-1.049501
5.894172	5.894172
10.925890	10.925890
13.664309	13.664309
12.320221	12.320221
8.065923	8.065923
1.891456	1.891456
-4.466679	-4.466679
-11.638098	-11.638098

Yes!

i) [2 pt] We have nearly reconstructed the output from `decompose`! All that remains is the residual error series. Calculate the residual error series from the previously created objects representing the trend and seasonal effects. Compare the by-hand calculation to the results from `decompose`. Are they the same?

```
# calculate residual error series
res <- vt_ts - vt_trend_full - seasonal_means_centered

# check some equalities
cbind(head(res, n = 10), head(vt_decompose$random, n = 10))
```

```
           [,1]      [,2]
 [1,]        NA        NA
 [2,]        NA        NA
 [3,]        NA        NA
 [4,]        NA        NA
 [5,]        NA        NA
 [6,]        NA        NA
 [7,] 0.2301074 0.2301074
 [8,] 0.9278628 0.9278628
 [9,] 1.3147436 1.3147436
[10,] 2.7674194 2.7674194
```

```
cbind(tail(res, n = 10), tail(vt_decompose$random, n = 10))
```

```
           [,1]       [,2]
 [1,]  2.8751624  2.8751624
 [2,] -0.7494993 -0.7494993
 [3,] -0.3346301 -0.3346301
 [4,] -0.5394318 -0.5394318
 [5,]         NA         NA
 [6,]         NA         NA
 [7,]         NA         NA
 [8,]         NA         NA
 [9,]         NA         NA
[10,]         NA         NA
```

Looking ahead

The goal of this lab was to elucidate how additive time series are decomposed, and how the data may be used to estimate each of the terms in a decomposition. We will continue to develop our time series toolkit next week as we consider serial correlation.