Homework 4

2025-09-18

Question 7.1

> Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of α (the first smoothing parameter) to be closer to 0 or 1, and why?

In my work, we run daily models on telemetry data to monitor fleet health. Exponential smoothing would help reveal trends in overall health, showing whether it is improving or degrading. Since we already aggregate data across a large population, I would expect α to be close to 1, placing more weight on observed data. Because customer behavior is strongly periodic on a weekly cycle, γ would likely be closer to 0 because there is strong seasonality. The most critical parameter would be β, as it directly captures the trend we are trying to monitor.

Question 7.2

> Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (file temps.txt), build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years. (Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to combine it with other models if you'd like to. There's certainly more than one reasonable approach.)
>
> Note: in R, you can use either `HoltWinters` (simpler to use) or the `smooth` package's `es` function (harder to use, but more general). If you use `es`, the Holt-Winters model uses model="AAM" in the function call (the first and second constants are used "A"dditively, and the third (seasonality) is used "M"ultiplicatively; the documentation doesn't make that clear).

First, we load the necessary libraries:

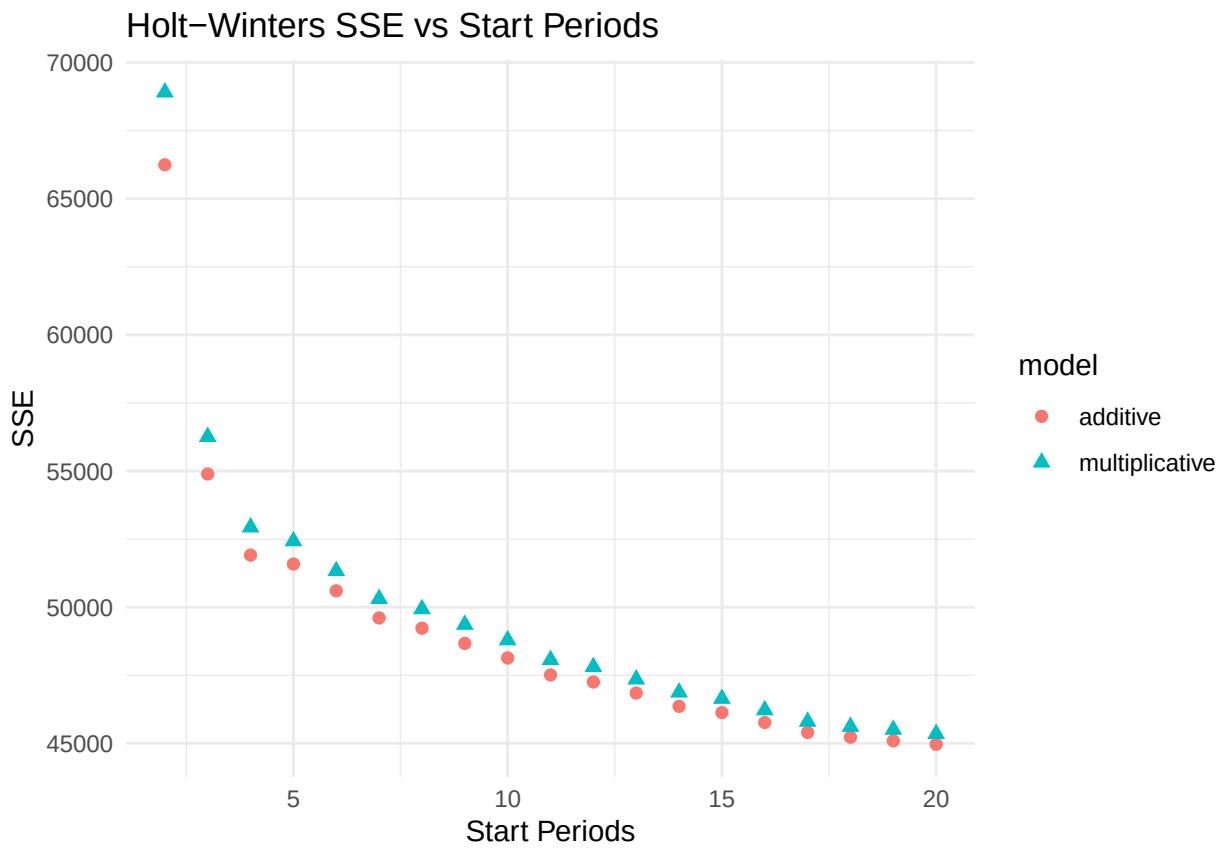
```
library(printr)      # pretty print for Rmd
library(lubridate)   # dates
library(ggplot2)     # plots
library(dplyr)       # dataframes
library(tidyr)
library(tidyverse)
library(stats, include.only = "ts")
library(stringr, include.only = "str_replace")
library(zoo)         # time series

# set seed for reproducibility
set.seed(42)
```

Now let's load the data:

```
temps_long <- read.csv("./data 7.2/temps.txt", sep = "\t")
freq = nrow(temps_long)
dates <- as.Date(temps_long$DAY, "%d-%b")

temps <- temps_long %>%
    pivot_longer(
        cols = -DAY,
        names_to = "YEAR",
        values_to = "TMax"
    ) %>% mutate(
      YEAR = as.integer(str_replace(YEAR, "X", "")),
      DATE = as.Date(sprintf("%s-%d", DAY, YEAR), "%d-%b-%Y"),
      MONTH = month(DATE, label = TRUE, abbr = TRUE),
      MONTH_DAY = format(DATE, "%b-%d"),
      YEARLESS_DATE = update(DATE, 2024) # leap year
    ) %>% arrange(DATE)

# zoo to ts
temps_ts <- ts(coredata(zoo(temps$TMax, temps$DATE)), frequency = freq)
```

There are a few parameters that still need tuning: the seasonality type (multiplicative vs. additive) and the `start.periods` value. Let's compare the SSE scanning over these

```
periods <- seq(2,20)
additive_function <- function(n)
{
  HoltWinters(
    temps_ts,
    seasonal = "additive",
    start.periods = n
  )$SSE
}

multiplicative_function <- function(n) {
  HoltWinters(
    temps_ts,
    beta=FALSE,
    seasonal = "multiplicative",
    start.periods = n
  )$SSE
}

additive <- sapply(periods, additive_function)
multiplicative <- sapply(periods, multiplicative_function)

sse <- data.frame(
  periods=periods,
  additive=additive,
  multiplicative=multiplicative
)

sse_long <- pivot_longer(
  sse,
  cols = c("additive", "multiplicative"),
  names_to = "model",
  values_to = "SSE"
)

ggplot(sse_long, aes(x = periods, y = SSE, color = model, shape = model)) +
     geom_point(size = 2) +
     labs(title = "Holt-Winters SSE vs Start Periods",
          x = "Start Periods", y = "SSE") +
     theme_minimal()
```

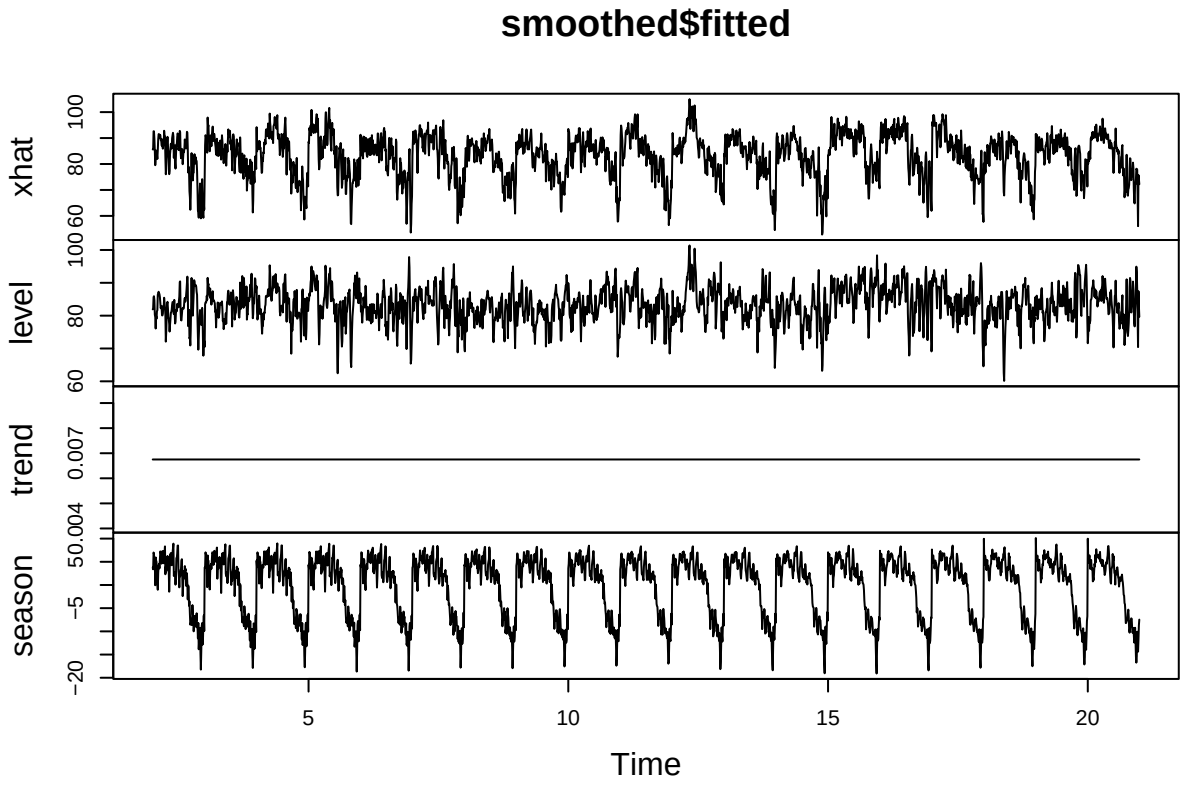


We can see a kink in the SSE at 4 start periods for both types of seasonality models, and that additive models are generally slightly lower error. So let's use those parameters

```
smoothed <- HoltWinters(temps_ts, seasonal = "additive", start.periods = 4)
```

Let's look at the decomposed components produced by the Holt-Winters method.

```
plot(smoothed$fitted)
```

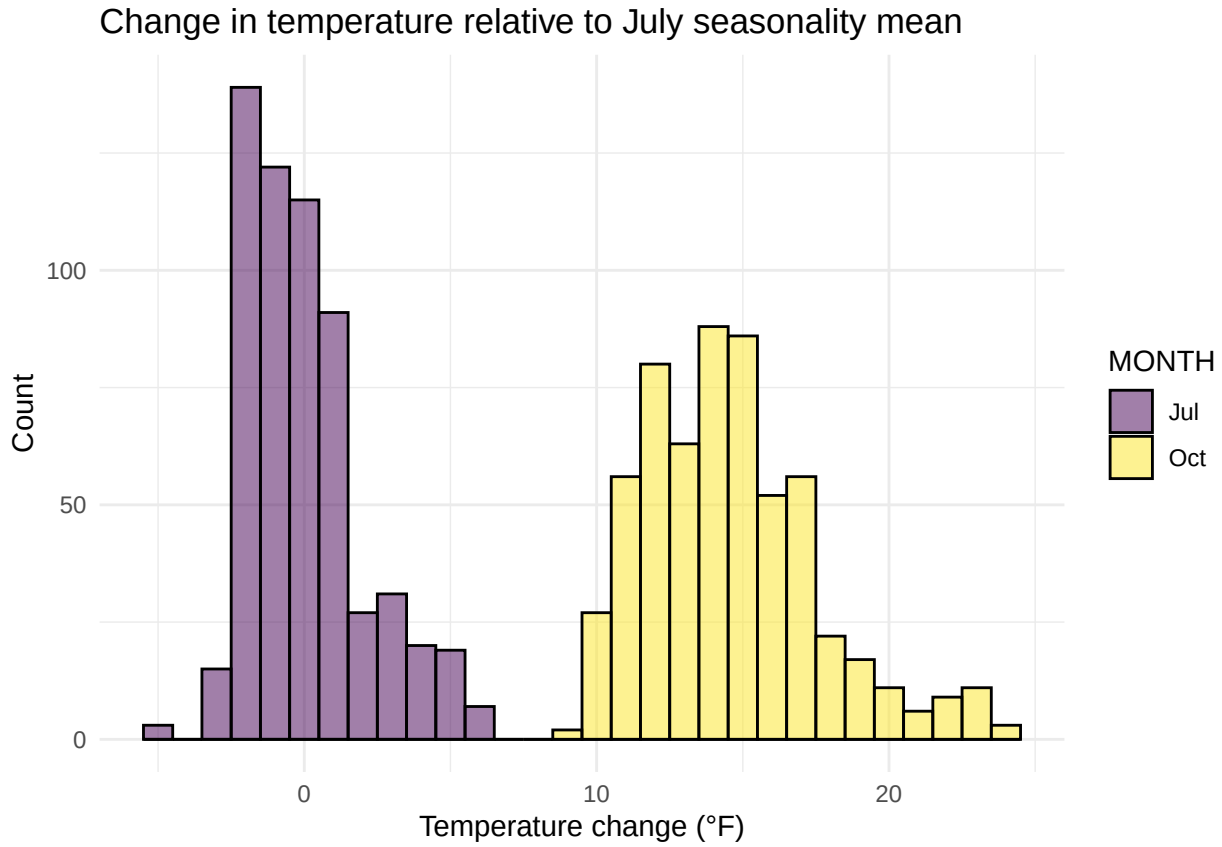


We can see that a lot of the noise is absorbed by the level component, leaving a much cleaner signal in the seasonal data. Let's add dates back to the data and run CUSUM on the seasonality! Note that we need to remove 1996 because the first period is not included as an output of a seasonal Holt-Winters model.

```
smoothed_data <- temps %>% filter(YEAR > 1996) %>%
  mutate(
    xhat = smoothed$fitted[,"xhat"],
    level = smoothed$fitted[,"level"],
    trend = smoothed$fitted[,"trend"],
    season = smoothed$fitted[,"season"]
  )

statistics <- smoothed_data %>%
  filter(MONTH == "Jul") %>%
  summarise(
    mu = mean(season),
    std = sd(season)
  )

mu <- statistics$mu
std <- statistics$std
```

Using the same method as Homework 3, I'm going to assume that July is summer and October is fall. Therefore, let's compare the temperatures for those months using a histogram. I am going to perform the $\mu - x_t$ shift in the temperature so that we can use this to choose C.

```r
ggplot(
  smoothed_data[(smoothed_data$MONTH == "Jul") | (smoothed_data$MONTH == "Oct"),],
  aes(x = mu - season, fill = MONTH)
) +
geom_histogram(
  position = "identity",
  alpha = 0.5,
  binwidth = 1,
  color = "black"
) +
labs(
  x = "Temperature change (°F)",
  y = "Count",
  title = "Change in temperature relative to July seasonality mean"
) +
theme_minimal()
```



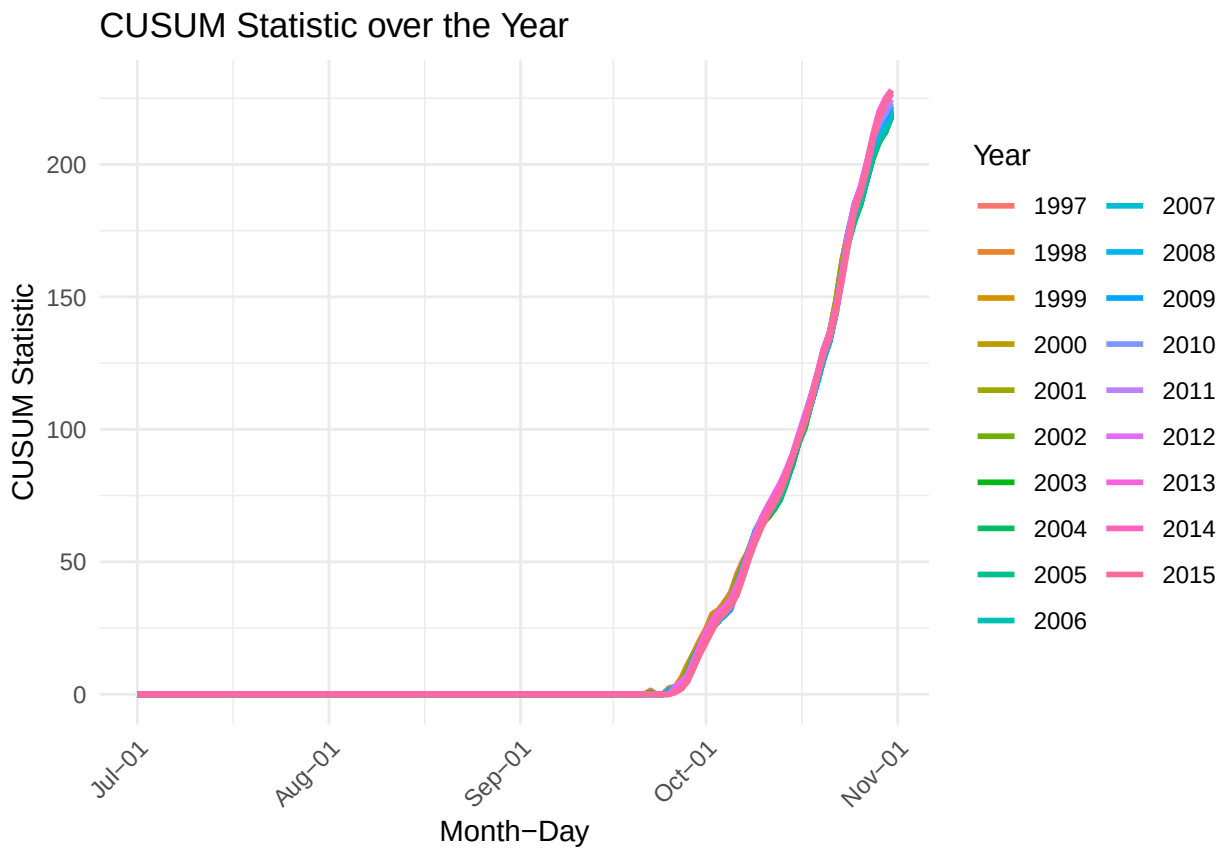
Unlike with the raw data we have a complete separation of July and October! Given the clean separation I'm going to choose C to be the middle of the gap, therefore,

```r
C <- 8
```

With these parameters chosen we just need to select a decision interval. I'm going to calculate and plot the change in the CUSUM statistic and to see if there is a clear threshold. I found this very useful function to do a resetting cumulative sum[1]. It needed some modification, but pointed me in the right direction

[1] https://stackoverflow.com/a/49076872/1543042

```r
end_of_summer <- smoothed_data[
  ("Jul" <= smoothed_data$MONTH) &
    (smoothed_data$MONTH <= "Oct"),
  ]

end_of_summer <- end_of_summer %>%
  group_by(YEAR) %>%
  mutate(
    CUSUM_STAT = accumulate(
      season,
      function(St_min1, xt, year) {
        max(0, St_min1 + (mu - xt - C))
      },
      .init = 0
    )[-1]
  )



ggplot(
  end_of_summer,
  aes(x = YEARLESS_DATE, y = CUSUM_STAT, color = factor(YEAR))
) +
  geom_line(linewidth = 1) +
  labs(
    x = "Month-Day",
    y = "CUSUM Statistic",
    color = "Year",
    title = "CUSUM Statistic over the Year"
  ) +
  scale_x_date(date_labels = "%b-%d") +
  theme_minimal() +
  guides(color = guide_legend(ncol = 2)) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



This is a beautiful CUSUM statistic, we can choose almost any non-zero value of the critical value. Let's choose

```
T <- 5
```

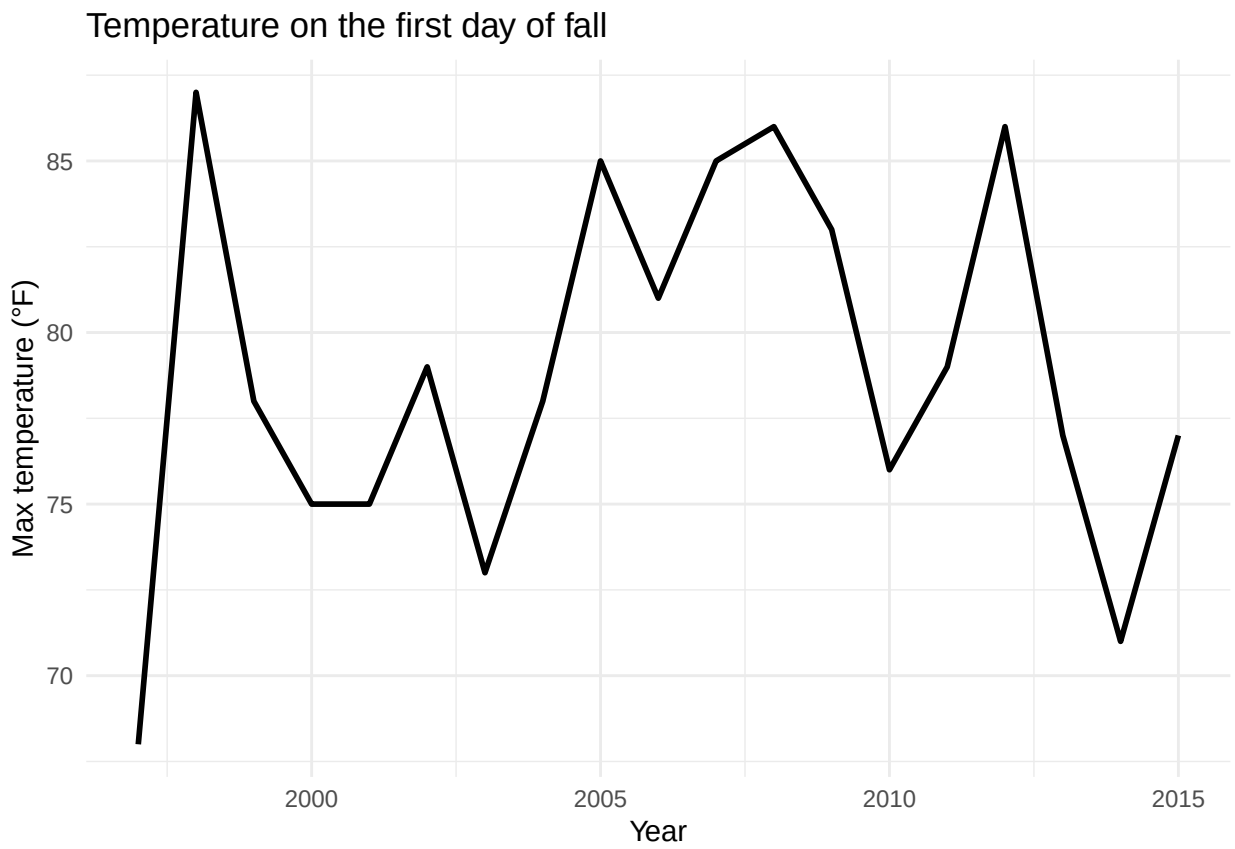
Now we have the parameters for the CUSUM model, let's see when fall started!

```
first_day_of_fall <- end_of_summer %>%
  filter(CUSUM_STAT >= T) %>%
  group_by(YEAR) %>%
  slice_min(DATE, n = 1) %>%
  ungroup() %>%
  select(
    YEAR,
    TMax,
    MONTH_DAY
  )

left <- first_day_of_fall[1:10,]
right <- first_day_of_fall[11:20,]

cbind(left,right)
```

YEAR	TMax	MONTH_DAY	YEAR	TMax	MONTH_DAY
1997	68	Sep-28	2007	85	Sep-28
1998	87	Sep-27	2008	86	Sep-28
1999	78	Sep-27	2009	83	Sep-28
2000	75	Sep-28	2010	76	Sep-28
2001	75	Sep-27	2011	79	Sep-28
2002	79	Sep-28	2012	86	Sep-28
2003	73	Sep-28	2013	77	Sep-28
2004	78	Sep-28	2014	71	Sep-29
2005	85	Sep-28	2015	77	Sep-29
2006	81	Sep-28	NA	NA	NA

Using periodic data we get a much narrower range of dates for the end of fall (in Homework 3 I said between Sept 17 and Oct 13); however, that comes at the cost of a wider range of possible temperatures on the first day of fall because we averaged out the trend fluctuations. This would point towards fall not starting later, let's have one last check and see what the temperature on the first day of fall is.

```
ggplot(
  first_day_of_fall,
  aes(x = YEAR, y = TMax)
) +
  geom_line(linewidth = 1) +
  labs(
    x = "Year",
    y = "Max temperature (°F)",
    title = "Temperature on the first day of fall"
  ) +
  theme_minimal()
```



Visually that does not appear to have any trend, but to formalize that, let's see if a linear regression shows anything more subtle.

```
summary(lm(TMax ~ YEAR, data = first_day_of_fall))
```

```
##
## Call:
## lm(formula = TMax ~ YEAR, data = first_day_of_fall)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8526 -3.2579 -0.4737  4.8737  9.0316
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -153.3789   463.8711  -0.331    0.745
## YEAR           0.1158     0.2312   0.501    0.623
##
## Residual standard error: 5.521 on 17 degrees of freedom
## Multiple R-squared:  0.01453,    Adjusted R-squared:  -0.04343
## F-statistic: 0.2507 on 1 and 17 DF,  p-value: 0.623
```

The estimated slope is small relative to its standard error, resulting in a p-value of 0.623. Therefore, we cannot conclude that the slope is significantly different from zero. Therefore not only is fall not starting later, the temperature on the first day of fall is not increasing over the two decades we considered.