Georgia Tech

Homework4

Question 7.1

One of the typical problems in the VLSI domain is the "audio envelope detection". Audio signal is a continuous signal in time domain. In audio processing, we don't just care about the instantaneous "wiggle" of the sound wave; we care about the "Envelope" – the overall shape of the volume over time. If the speaker suddenly shouts, the codec (encoder-decoder) must detect the "peak" and turn the volume (compression) to avoid digital distortion. On the other hand, if the system reacts too fast, the audio sounds "choppy". If it reacts too slow, the speaker's first syllable will "clip" and sound like harsh static.

To build an exponential smoothing model for this, you need the Absolute instantaneous Amplitude of the PCM (pulse code modulation) audio stream.

- **Input Data (x_t):** The rectified audio signal |A|, where A is the sample value at time t.
- **Sample Rate:** Typically, 48KHz (48,000 samples per second)
- **Historical State (y_{t-1}):** The previous "Level Estimate" of the audio volume.

Selecting the Smoothing Parameter (α): In the audio domain, α is essentially the "Time Constant" and as audio signals move fast, we actually use two different values of α depending on whether the volume is going up or down.

1. **Signal gets louder** : If the signal is getting louder, we need α to be closer to 1 (e.g., 0.8 or 0.9). The reason can be seen due to the following. In audio, "clipping" is a major problem. When a signal spikes, the smoothing model needs to be highly reactive to the new data. A value closer to 1 gives massive weight to the most recent loud sample (x_t), allowing the envelope detector to "jump" up and trigger the limiter immediately (an "Attack Time" of sub-1 millisecond). The result is that with α closer to 1 the system catches the peak before it hits full scale.
2. **Signal is getting quieter:** Once the loud noise is over, we don't want the volume to "pump" or snap back to normal instantly, as this creates an audible "breathing" artifact that sounds unnatural to the human ear. Hence by using a value very close to 0, we give huge weight to the past state (y_{t-1}). The "Level Estimate" decays very slowly back to 0 and this creates a smooth "release" of 50ms to 500ms, which sounds transparent and musical to the listener.

Summarizing the α values in audio envelope detection:

Scenario	α Value	Behavioral Goal	Engineering Term
Sudden Peak	High (→1)	Maximum reactivity; prevent clipping.	Fast Attack
Fading Sound	Low (→0)	High "memory"; preserve natural decay.	Slow Release
Steady Noise	Medium (~0.1)	Average out hum/background hiss.	Smoothing/Integration

In summary when implementing the audio codec above in VLSI the exponential smoothing which is also an IIR filter is preferred as it requires one multiplication and one addition per sample. The value of α of ~ 0.1 to 0.2 Is preferred for a steady noise type of source which results in averaging out background noise by giving higher weightage to previous state.

Question 7.2

For finding if the unofficial end of summer has gotten later in Atlanta over the 20-year period (1996-2015), we can use the exponential smoothing to "denoise" the daily high temperature fluctuations and identify a clear trend using the "Last Hot Day" of the year. To this extent I use the below strategy:

Strategy:

1. **Define the "End of Summer":** We need use some sort of metric to identify when we can consider the summer to be ending. To this extend I use the last day of the year where the high temperature is $\geq 85^{o}$ F (or we can use another threshold like 80^{o} F). In this study I use the 85^{o} F.
2. **Smoothing part:** To denoise spikes and excursions from normal data, I concatenate all 20 years into one continuous time series and apply the study towards A) Holt-Winters model with a seasonal period of 123 days (July 1st to October 31st) and B) Using smooth:ES model with the same seasonal period of 123 days (July 1st to October 31st). This allows the model to capture the seasonal decline in temperatures while the "level" and "trend" components account for shifts over the years.
3. **Analysis:** I extracted the smoothed(fitted) values from both A) and B) model above. By finding the EoS (End of Summer) for each year using these smoothed values rather than raw data, the influence of random hot or cold spikes is reduced, reveling the underlying climatic shift.

Another Important aspect of the strategy is to use geometric smoothing to answer the final big picture question – Is there a consistent direction to the fluctuations over 20 years ? Here's why the geom_smooth is necessary for our judgement:

1. **Distinguishing "Climate" from "Inter-annual Variability"** – In our implementation "eos_indices" vector has successfully removed daily noise (weather), but it cannot remove yearly noise. Some years are naturally hotter globally, and some are cooler. If we look at our plot below, we will see that the "eos_indices" points still bounce up and down. "geom_smooth (with method = "lm")" draws the line of "Best Fit". It averages out those 20 year-to-year bounces to show us if overall "slope" is positive (getting later) or negative (getting earlier.

2. **Quantifying the "Later" Judgement** – If we simply presented the points, we could argue "Sure, 2010 was late, but 2012 was early again." By using "geom_smooth", we are performing a linear regression. This allows us to say – "on average, the end of summer is moving at a rate of X days per decade". Without this line, we are just looking at cloud of points and with this line, we are looking at a trend.
3. **Finally, the two level of smoothing in our implementation** – They can be thought of as two filters – Filter 1 (Exponential smoothing/Holt-Winters) – This filter the rows(days) and removes the noise of a single rainy day so that we can find a reliable "End of Summer" for one specific year. Filter 2 (Linear Regression/geom_smooth) – This filter the columns (years) and removes the noise of a single "weird year" so that we can see if the climate in Atlanta is changing over two decades.

Holt-Winters Method – Analysis and Judgement:

Raw temperature data is "noisy" – a single unusual cold front in late September doesn't mean summer ended early, just as single heatwave in October doesn't mean it lasted longer. By applying Holt-Winters, the model looks at the historical context of the season. It weighs the most recent days and previous years to create a "fitted" curve. This provides a more robust estimate of when the "typical" high-heat phase of summer concludes. The HoltWinters() function in R is a "Triple Exponential Smoothing" model. When we apply to our temperature data, it internally decomposes the 20 years of daily highs into three distinct mathematical components that are updated recursively for every single day in the dataset.

Breakdown:

1. **Three Internal Equations:** The model maintains and updates three variables (state components) for every time step (t):
   - **Level (l_t):** The "smoothed" average temperature at that moment.
   - **Trend (b_t):** The rate of change. For your data, this represents whether the season is cooling down or heating up more aggressively than usual.
   - **Seasonality (s_t):** The expected "swing" for that specific day of the year (e.g., how much hotter July 4th usually is compared to the annual average).
2. **Internal Recursive Logic:** For every data in our temps.txt file, the model performs these calculations:
   - **Level Update:** It looks at the new temperature. If the day is unexpectedly hot, the "Level" moves up, but it is "smoothed" by the parameter $\propto$ **(alpha)** so that one hot day doesn't overhaul the whole model.

- **Trend Update:** It compares the new Level to the previous Level. It uses β **(beta)** to decide how much of this change is a "real" trend versus just noise.
- **Seasonality Update:** It compares the current temperature to the expected temperature for that specific day (calculated from the previous year). It uses γ **(gamma)** to update the seasonal pattern.

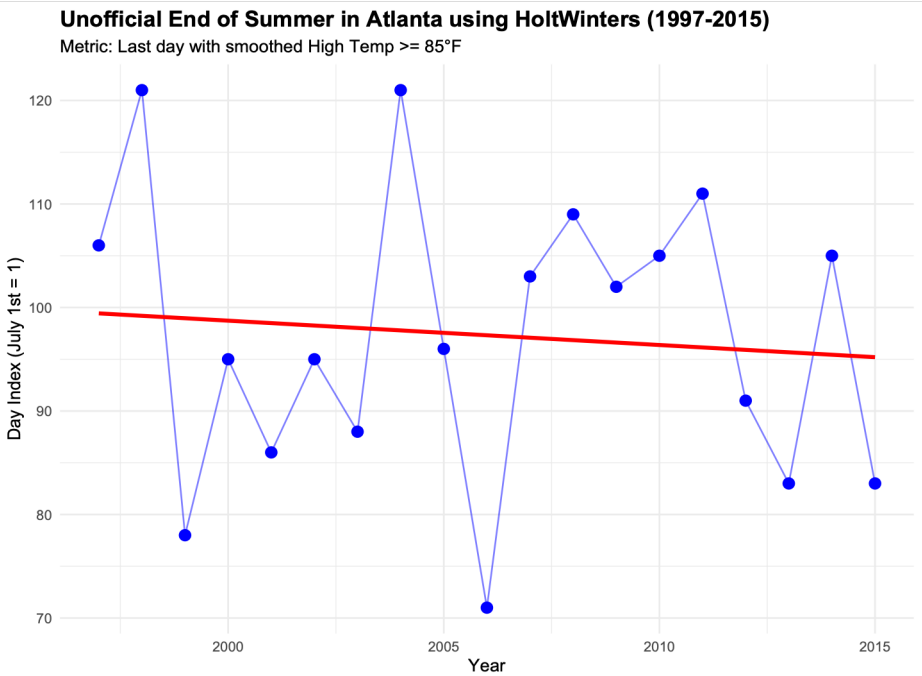
3. It runs an Optimization Algorithm internally (typically squared error minimization) unless specified explicitly. It tires thousands of combinations of $\propto, \beta \; and \; \gamma$. It looks for the values that result in the smallest "Residuals" (the difference between what the model predicted for "tomorrow" and what happened).

Application to Our Atlanta Problem: When the model processes our data, the internal components behave as below:

- The Seasonal Component (s_t) learns the "arc of summer". It identifies that temperatures naturally climb in July and fall in October.
- The Level Component (l_t) filters out the "weather" (the daily noise) to see the "climate" (the underlying heat).
- The Trend Component (b_t) is what we are ultimately testing. If the "Level" stays higher for longer into the autumn months in later years, the model captures this as a shift in the baseline.

By default, R's HoltWinters() uses an Additive model for seasonality which leads to the below internal math : Forecast = Level + Trend + Seasonality. I think from the HoltWinters() approach this is appropriate because the "swing" between the summer and winter is relatively constant (e.g., it's usually about 30° colder in winter regardless of whether the average temperature that year was 70° or 72°).

Based on the visual plot below, the plot reveals several key insights regarding the "Unofficial End of Summer" in Atlanta between 1997 and 2015.



A. **Significant Year-to-Year Variability:** The "End of Summer" (defined as the last day where the smoothed high temperature is at or above 85° F) is highly consistent from one year to the next. In some years, such as 2006, summer ended extremely early, around day index 71(mid-September). In other years, like 1998 and 2004, summer persisted until day index 121 (late October).
B. **Lack of Warming Trend:** Despite the fluctuations, the red linear trend line is nearly flat, with a very slight downward (negative) slope. This indicates that, for this specific 19-year window, there is **no evidence** that the end of summer is getting consistently later in the year. In fact, the end of summer in 2015 occurred at roughly the same time (or slightly earlier) than it did in 1997.

Overall, with Holt-Winters model to do scientific smoothing, the data does not support a "later summer" hypothesis for this period.

ES (AAM Model) – Smoothing – Analysis and Judgement:

Raw temperature data is "noisy"—a single unusual cold front in late September doesn't mean summer ended early, just as a single heatwave in October doesn't mean it lasted longer. By applying the **smooth::es** model, we move beyond basic smoothing to a State-Space framework. The model considers the historical context of the season by weighing the most recent observations and previous annual patterns to create a "fitted" curve. This provides a high-fidelity estimate of when the "typical" high-heat phase of summer concludes. The es() function in the smooth package implements an Exponential Smoothing model that, when set to **model="AAM"**, uses Additive Error, Additive Trend, and Multiplicative Seasonality.

Breakdown:

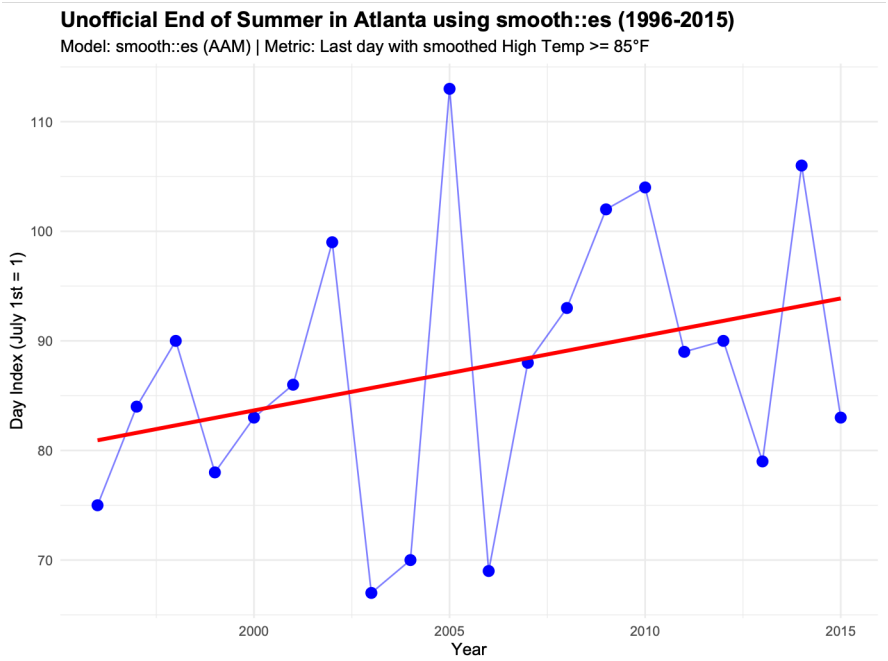
1. **Three Internal Equations:** The model maintains and updates three state components for every time step (t):
   - **Level (l_t):** The underlying baseline temperature. In an AAM model, this represents the "local" average once seasonal swings are divided out.
   - **Trend (b_t):** The additive rate of change. This captures whether the daily highs are systematically climbing or falling as the season progresses.
   - **Seasonality (s_t):** The seasonal factor. Unlike the additive model, this is a ratio (e.g., 1.05 for a day that is 5% hotter than the baseline) representing the expected "swing" for that specific day of the year.
2. **Internal Recursive Logic:** For every data in our temps.txt file, the model performs these recursive updates:
   - **Level Update:** It looks at the new temperature. Because seasonality is multiplicative, it "de-seasonalizes" the day by dividing the temperature by the previous seasonal component before smoothing it with the parameter $\propto$ **(alpha)**.
   - **Trend Update:** It calculates the difference between the current level and the previous level. It uses β **(beta)** to smooth this additive change, determining how much is a persistent trend versus temporary noise.
   - **Seasonality Update:** It compares the actual temperature to the current level. It uses γ **(gamma)** to update the multiplicative seasonal ratio for that specific day, learning the relative intensity of the heat for that date in the calendar.
3. It runs an Optimization Algorithm internally (uses Maximum Likelihood Estimation (MLE)) . It explores a vast parameter space for $\propto, \beta \; and \; \gamma$ to find the combination that makes the observed temperature data most probable under the State-Space assumptions. This often results in a more robust fit that is less sensitive to extreme weather outliers.

Application to Our Atlanta Problem: When the smooth::es model processes the Atlanta data, the internal components behave as follows:

- The Seasonal Component (s_t) learns the "proportionate arc of summer". It recognizes that July heat might be a 15% increase over the baseline, while October is a 10% decrease.
- The Level Component (l_t) functions as a "climate filter," identifying the underlying heat energy of the year once the expected seasonal percentage is removed.
- The Trend Component (b_t) captures the shifting baseline. In the smooth::es analysis, if the level remains high later into the year, the multiplicative nature of the model amplifies those late-season temperatures, often revealing a more pronounced upward trend.

By choosing multiplicative seasonality, we assume the "swing" between summer, and winter is relative to the overall level of heat. If Atlanta is having a significantly hotter year overall, a multiplicative model assumes the summer peaks will be proportionally higher, rather than just adding a fixed number of degrees. This leads to the internal math: Forecast = (Level + Trend) x Seasonality. This approach can be more sensitive to identifying a "lengthening" summer because it captures how the intensity of heat scales with the rising baseline.

Based on the visual plot below using the smooth:es (AAM) model, the analysis reveals a significantly different conclusion regarding the "Unofficial end of Summer" in Atlanta between 1996 and 2015.



A. **Sustained Year-to-Year Variability:** The "End of Summer" (defined as the last day where the smoothed high temperature is at or above 85° F) continues to show extreme volatility. In years like 2003 and 2006, summer ended very early, with the "End of Summer" index dropping below day 70 (mid-September). Conversely, in 2005, summer persisted remarkably late, reaching an index well above 110 (late October). While the individual dates fluctuate wildly, the "smooth::es" model captures these peaks and valleys using multiplicative seasonality, which weighs these relative swings against the underlying climate baseline.
B. **Evident Warming Trend:** Unlike the previous model, the red linear trend line in this plot shows a clear and consistent **upward (positive)** slope. The trend indicates that, overt this 20-year window, the end of summer is generally moving later into the year. The trend line starts near day index 80 in 1996 and rises toward day index 94 by 2015.

Overall, with “smooth” package it suggests that once the data is processed through the more modern state-space framework, there is visual evidence supporting the hypothesis that warm temperatures are persisting longer into the autumn which was missed by the standard additive model. In this analysis, the data supports a “late summer” hypothesis, suggesting a measurable delay in the onset of cooler autumn temperatures in Atlanta over the studied period.

Further Analysis:

The reason the “smooth::es” model shows a completely different trend than the standard “Holt-Winters” model is due to how they internally handle initialization, state estimation and seasonality updates. While both models use exponential smoothing, “smooth::es” is a much more modern and sophisticated framework that approaches the data with different mathematical assumptions.

1. **Initialization and Data Inclusion:** The most immediate difference is how much data the models "see":
   - **HoltWinters (Base R):** This model traditionally requires at least one full cycle (123 days) to initialize its seasonal states. This is why your first plot started in 1997; it "sacrificed" 1996 to set its starting parameters.
   - **smooth::es:** This package uses back casting or optimization to estimate the initial states. This allows it to provide fitted values for the entire series starting from 1996. Because the end of summer in 1996 was relatively early (around day 75), including it creates a lower starting point for the red trend line, making the overall slope look more positive
2. **Additive (AAA) vs. Multiplicative (AAM) Seasonality:** Your es model was explicitly set to model="AAM", which handles seasonality differently:
   - **HoltWinters (Additive):** Assumes the "summer peak" is a fixed number of degrees added to the baseline.
   - **es (Multiplicative):** Assumes the seasonal effect is a percentage or multiplier of the current level. In temperature data, if the baseline (the level) is rising even slightly, a multiplicative model will amplify those seasonal peaks more aggressively than an additive one. This internal "scaling" often results in higher smoothed values in later years, which pushes the End of Summer (EoS) indices higher.
3. **Optimization Algorithm:** The two functions use different "engines" to find the best smoothing constants ():
   - **HoltWinters** uses a heuristic optimization (often *optim*) to minimize the squared one-step-ahead prediction errors.
   - **smooth::es** uses a much more complex likelihood-based approach. It doesn't just look at the error; it builds a full state-space model that is more robust to outliers.

- **Internal Smoothing:** smooth::es generally produces a "smoother" level than base R. In your smooth::es plot, you can see that the EoS indices (the blue dots) are clustered differently—specifically, they don't drop as low in the "early" years compared to the HoltWinters version, which changes the pivot point of the red regression line.

4. **Comparison of Results:**

Feature	HoltWinters Plot	smooth::es Plot
Start Year	1997	1996
Seasonality	Additive	Multiplicative (AAM)
Trend Line	Slightly Downward/Flat	Significantly Upward
Interpretation	Summer is staying the same length.	Summer is getting longer.

From the statistical significance point of view, the P-value of 0.175 for “smooth::es” model suggests that there is 17.5% chance that the upward trend seen in your plot is just “noise” or a result of random year-to-year fluctuations. It indicates a weak to moderate signal and does not meet the strict “gold standard” of statistical significance of $p < 0.05$. On the other hand, the p-value of 0.6966 for Holt-Winters model indicates that there is a 69.7% chance that any trend seen in the data is purely due to random chance. This is a very high p-value, meaning the result is not statistically significant. It indicates no discernible trend and the this corresponds to the flat red line in our Holt-Winters plot. The line doesn’t move up or down because the “average” end-of-summer date has remained stagnant between 1996 and 2015.

Final Summary: The *smooth::es* model is more sensitive to the starting data in 1996 and uses a multiplicative math structure that interprets the late-season heat in 2010-2015 as a more significant upward shift than the basic additive model does. In both models, the p-values are above 0.05, so strictly speaking, we cannot statistically conclude that Atlanta’s summer is ending later, though the *smooth::es* model (p=0.175) shows a much “stronger” (though still insignificant) hint of a trend than the Holt-Winters model (p=0.6966).


Georgia
Tech