FireFinder: An Open-Source Deterministic Cooking Event Detection Algorithm

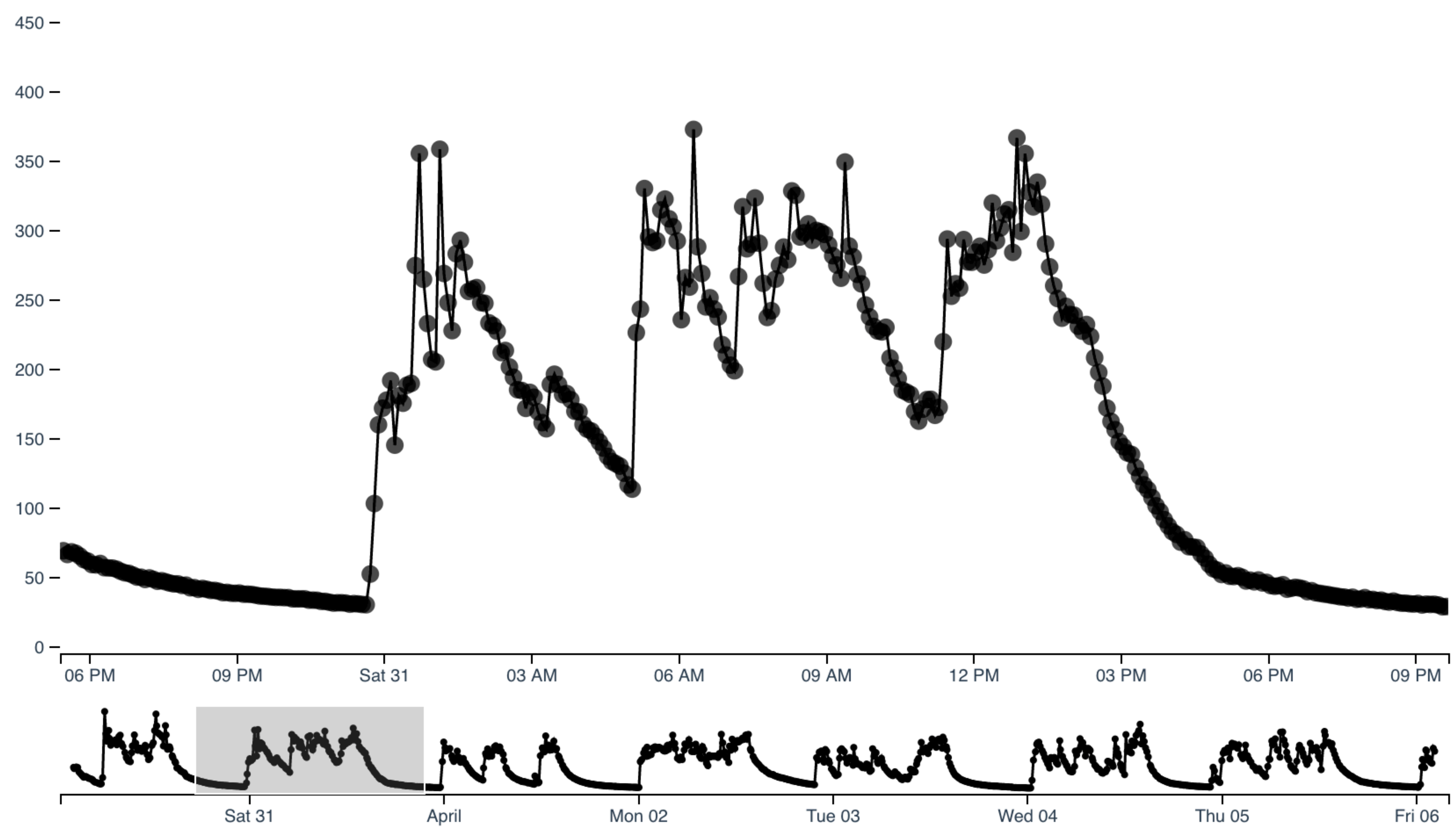
Danny Wilson, Ph.D.
Cofounder and CEO
danny@geocene.com

Common Challenges of Analyzing Sensor Data

What is “cooking?”

TRAINSET Help License Clear Export

Filename: sumsarizer_v1_012993.csv



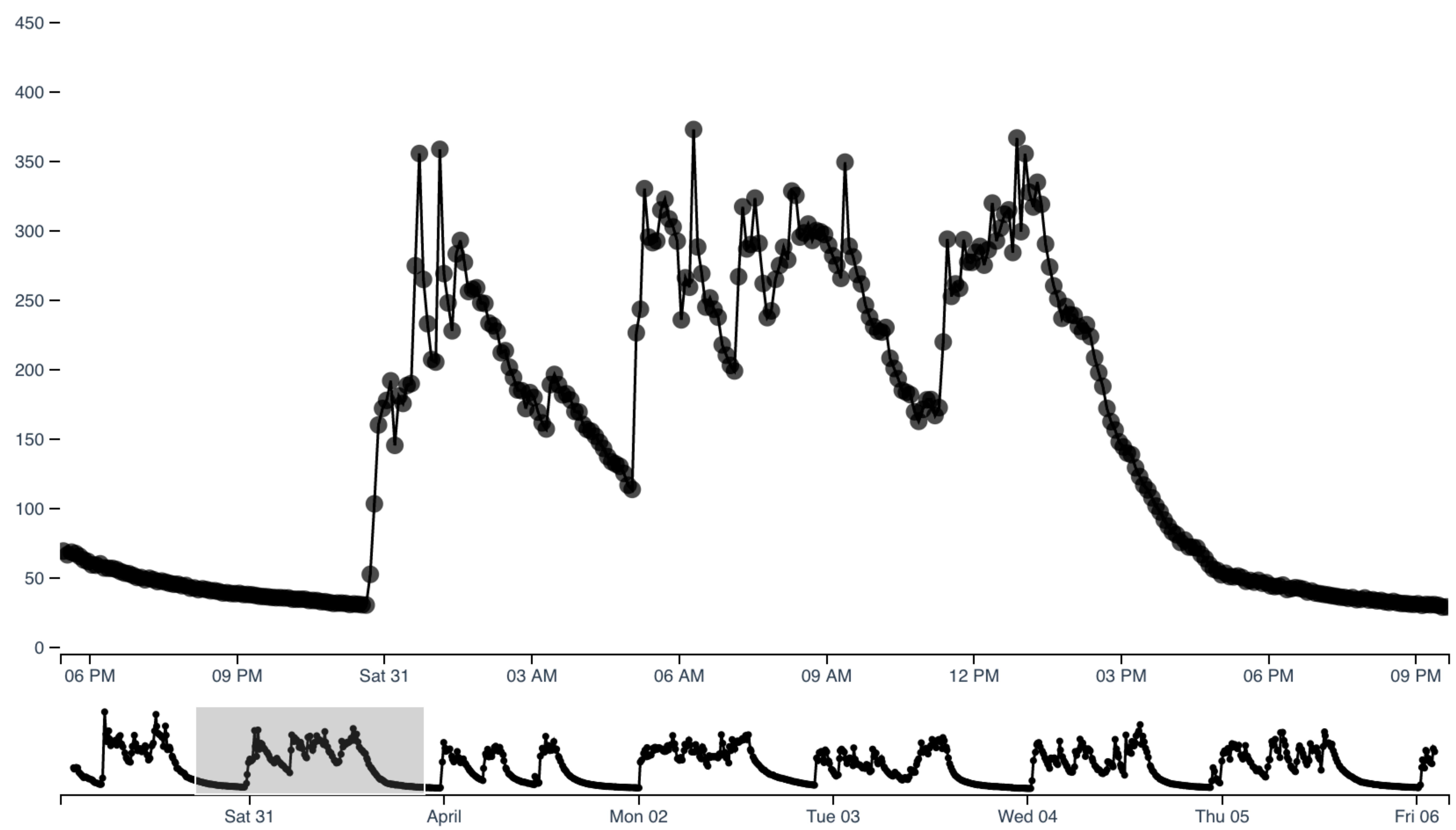
What is usage?

- When the stove is hot?
- When the fire is burning and making emissions?
- When food/drink is on the stove?
- When useful energy is being added to the pot?
- When meaningful amounts of fuel are being consumed?

TRAINSET

Help License Clear Export

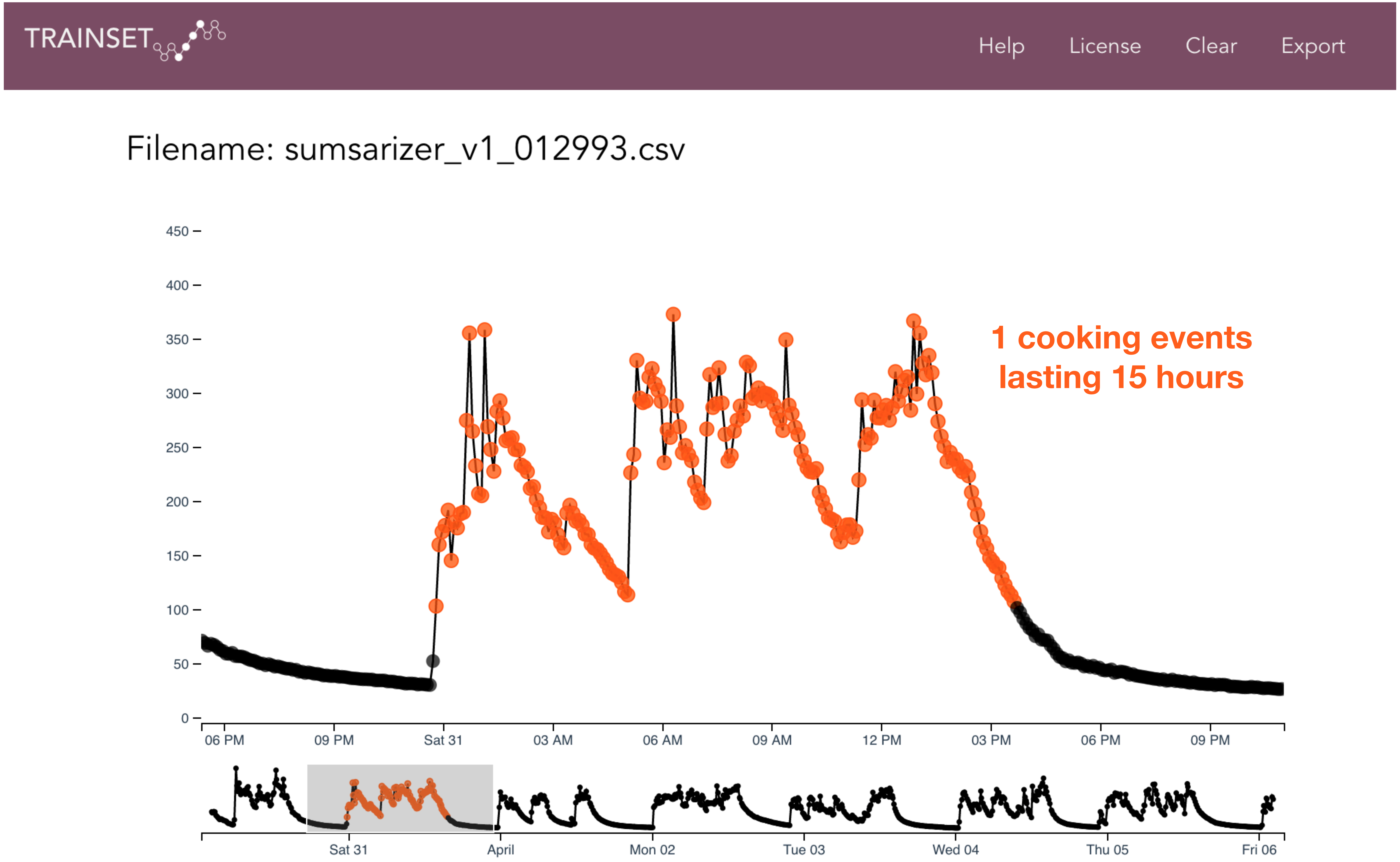
Filename: sumsarizer_v1_012993.csv



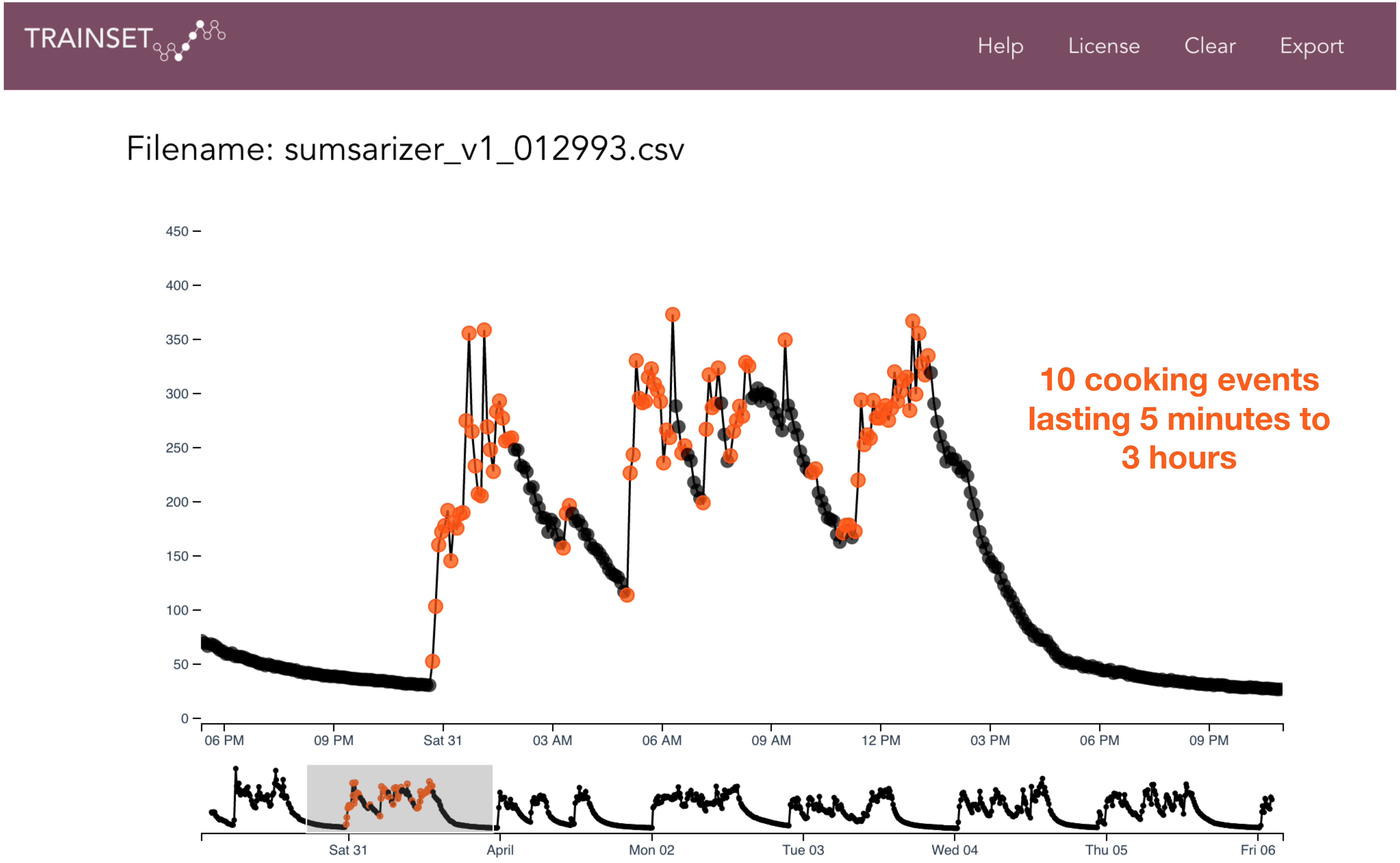
What is cooking?

- How many cooking events was that?
- How long did they last?
- What did all that mean? What was the user’s behavior?

Usage - When the stove is 'hot'?

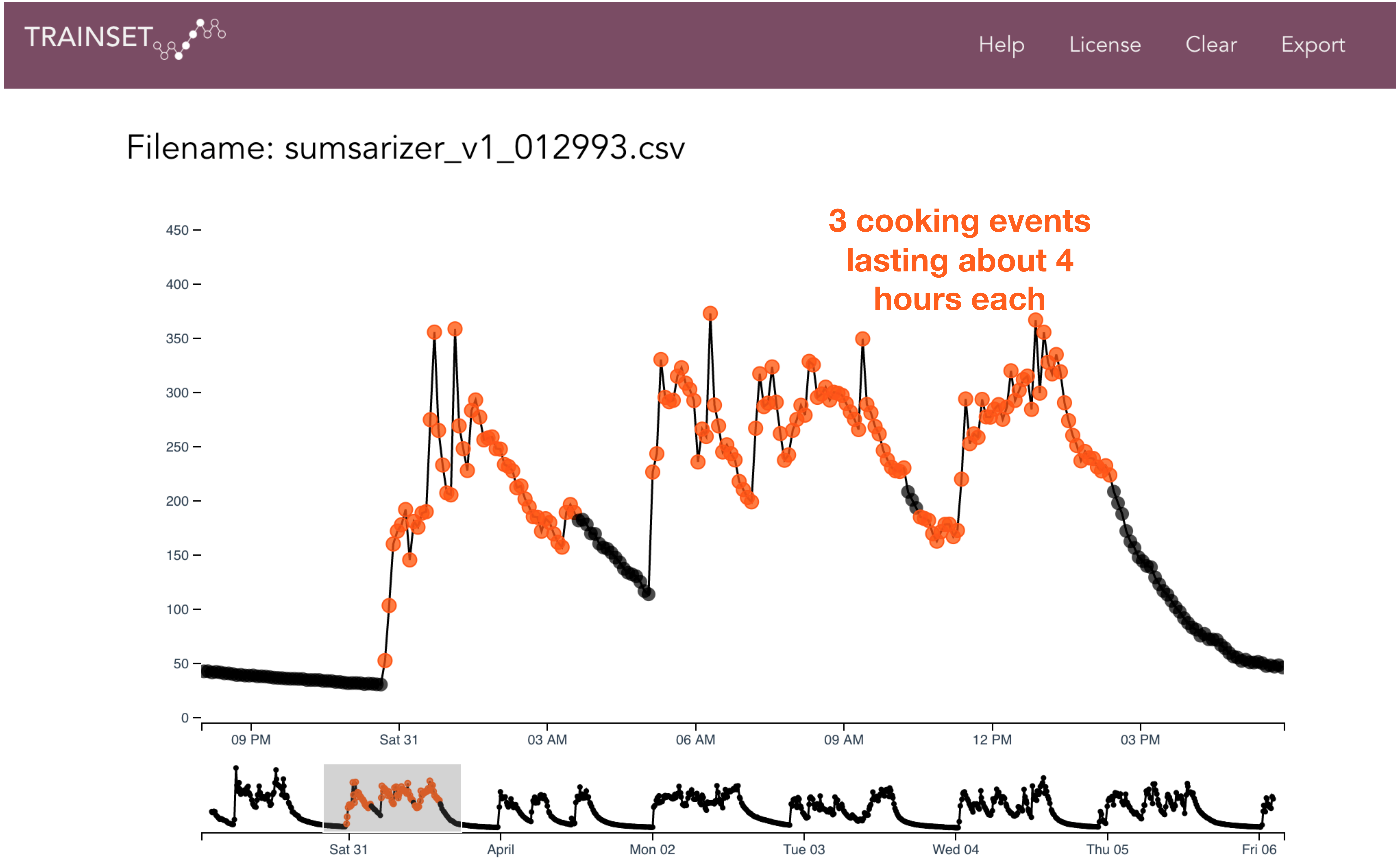


Usage - When heat is added?



10 cooking events lasting 5 minutes to 3 hours

Usage - when there is smoke?

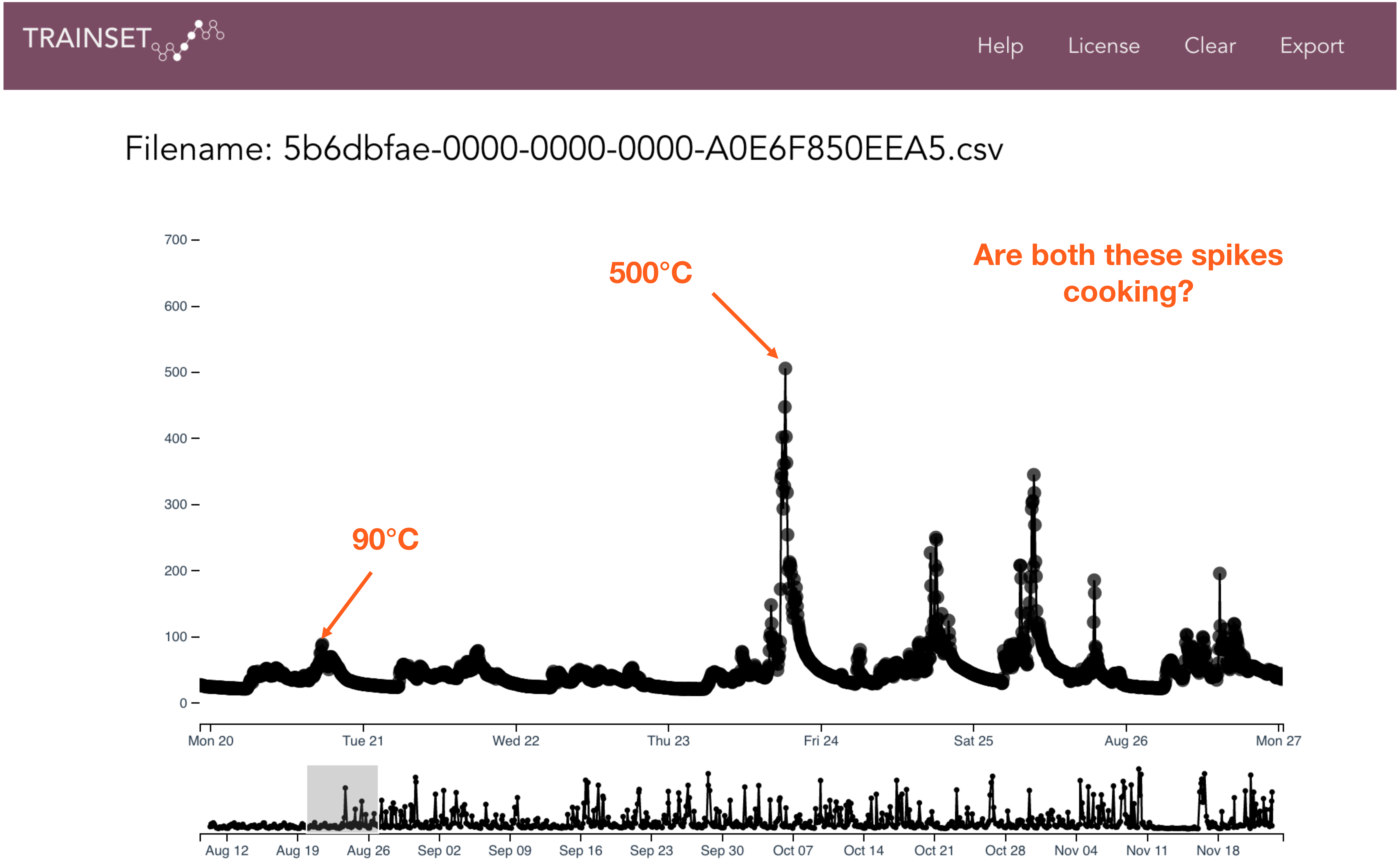


“Cooking is whatever you care about.”

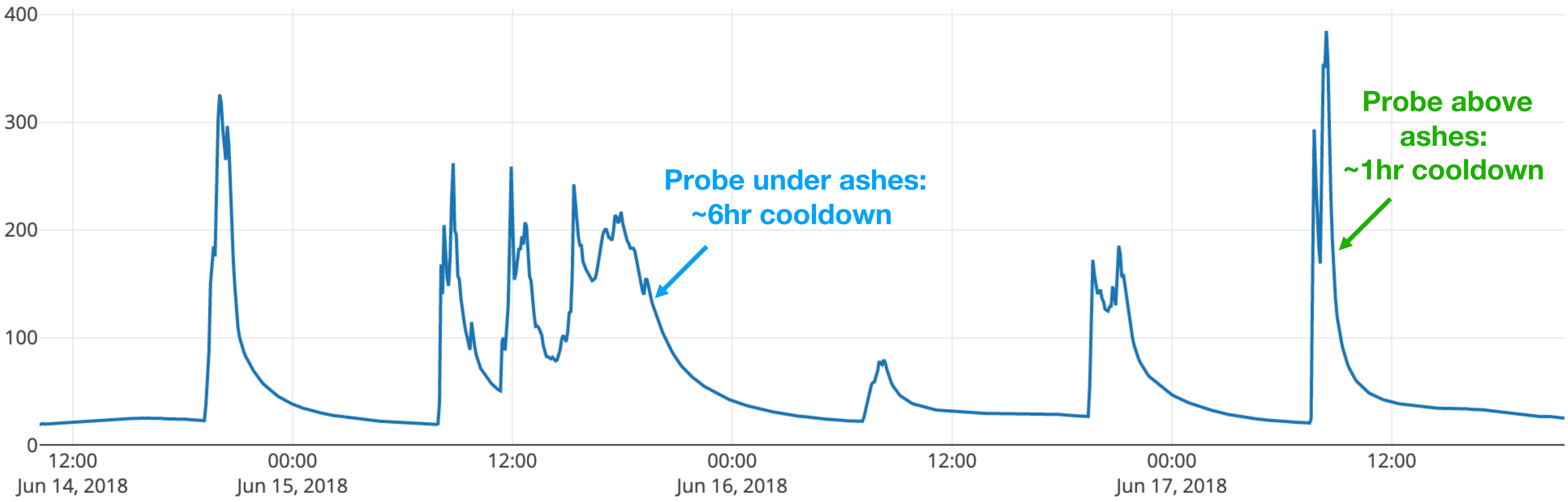
–Ancient Cookstove Researcher Proverb

Challenges

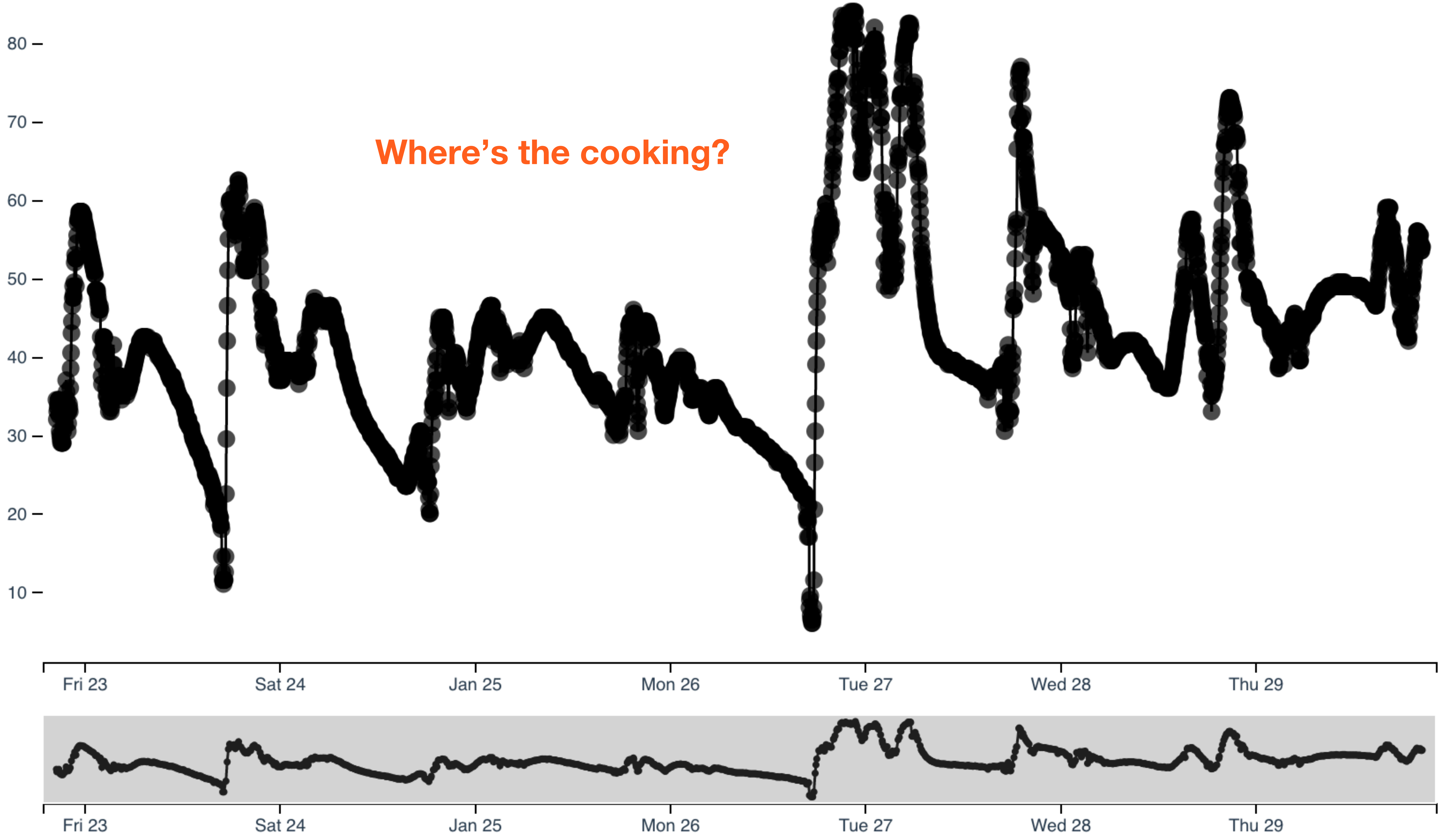
Context changes



Context Changes



Bad Probe Placement & Low Temperatures



“If your eyes can’t tell, neither can the algorithm.”

–(Another) Ancient Cookstove Researcher Proverb

How to Analyze Data

Primary Goals

Identify events

Limit bias: balance false positives with false negatives. You’re never going to get every point perfect, but try to limit bias.

Turn time-wise event booleans into file-wise event summaries

Turn file-wise event summaries into meaningful insights

Analytics Tools

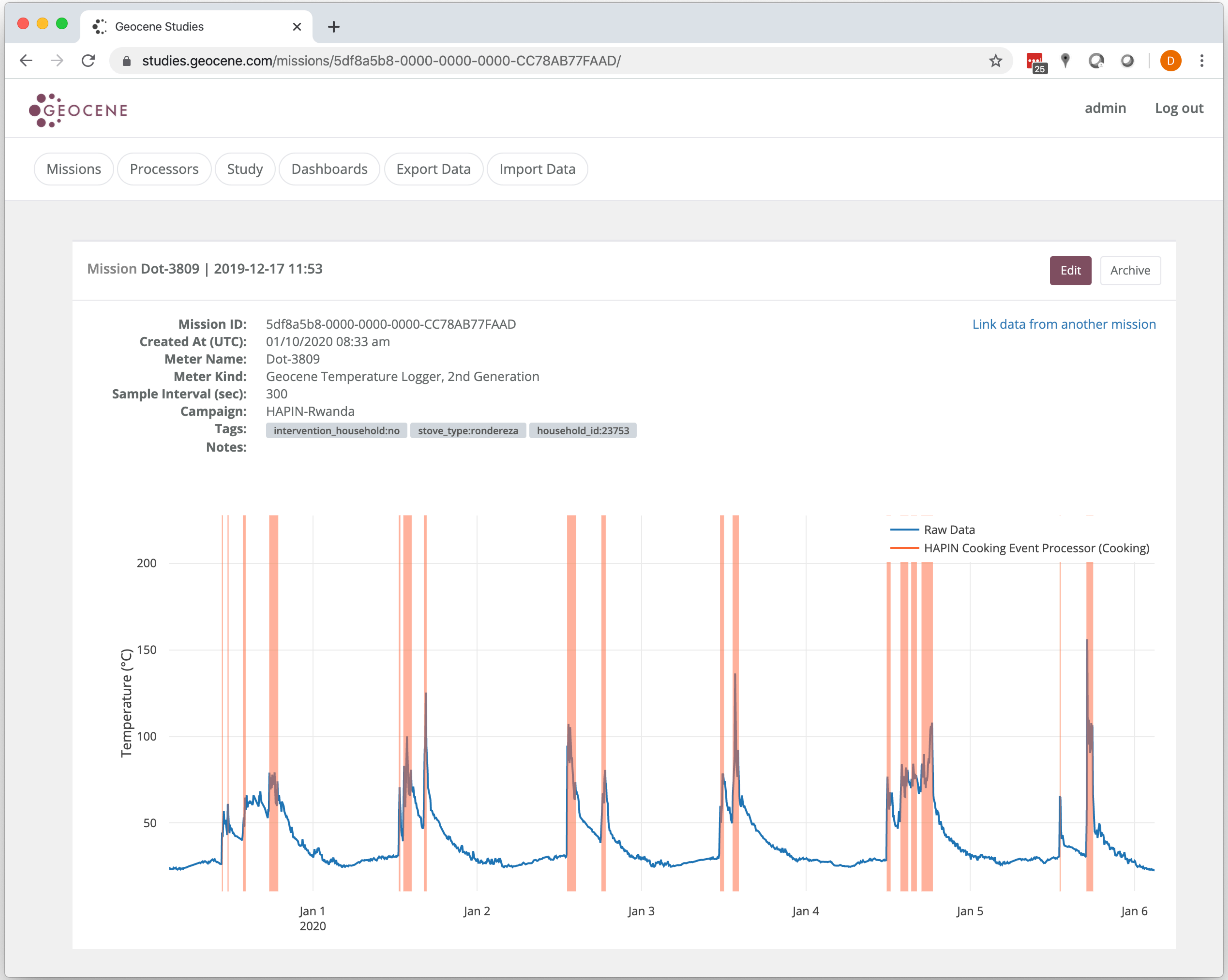
Excel/spreadsheets technical skill = moderate not realistic for more than ~10 data loggers

Code in **Python**, **R**, etc technical skill = very high

SUMIT technical skill = low iButtons only, R-Shiny app - not realistic for large datasets

SUMSarizer technical skill = high open-source R package. SUMSarizer *powers* Geocene Studies.

Geocene Studies technical skill = low point-and-click; no coding needed

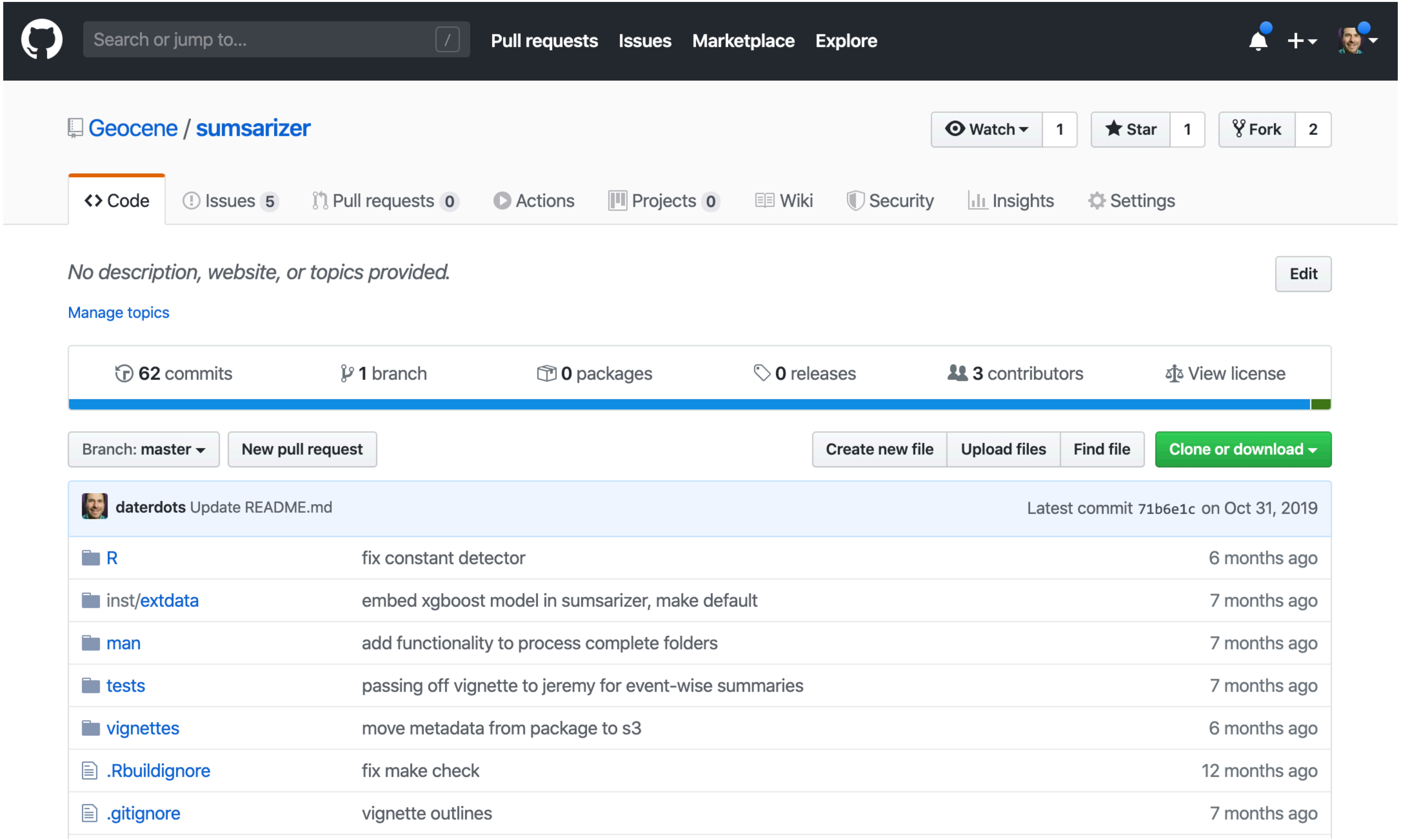

Geocene Studies
studies.geocene.com/missions/5df8a5b8-0000-0000-0000-CC78AB77FAAD/
GEOCENE
admin
Log out
Missions
Processors
Study
Dashboards
Export Data
Import Data
Mission Dot-3809 | 2019-12-17 11:53
Edit
Archive
Mission ID: 5df8a5b8-0000-0000-0000-CC78AB77FAAD
Created At (UTC): 01/10/2020 08:33 am
Meter Name: Dot-3809
Meter Kind: Geocene Temperature Logger, 2nd Generation
Sample Interval (sec): 300
Campaign: HAPIN-Rwanda
Tags: intervention_household:no
stove_type:rondereza
household_id:23753
Notes:
Link data from another mission
Raw Data
HAPIN Cooking Event Processor (Cooking)
Temperature (°C)
200
150
100
50
Jan 1
2020
Jan 2
Jan 3
Jan 4
Jan 5
Jan 6

Common Algorithms

Ilse Ruiz Mercado's algorithm (deterministic)

Geocene FireFinder (deterministic but tunable)
- Available in SUMSarizer and Geocene
- Tuning: primary threshold, minimum event length, minimum inter-event gap, minimum event temperature.
- Currently in-use on about 7000 cookstoves worldwide ranging from rocket, TLUD, TSF, charcoal, chulha, and plancha.

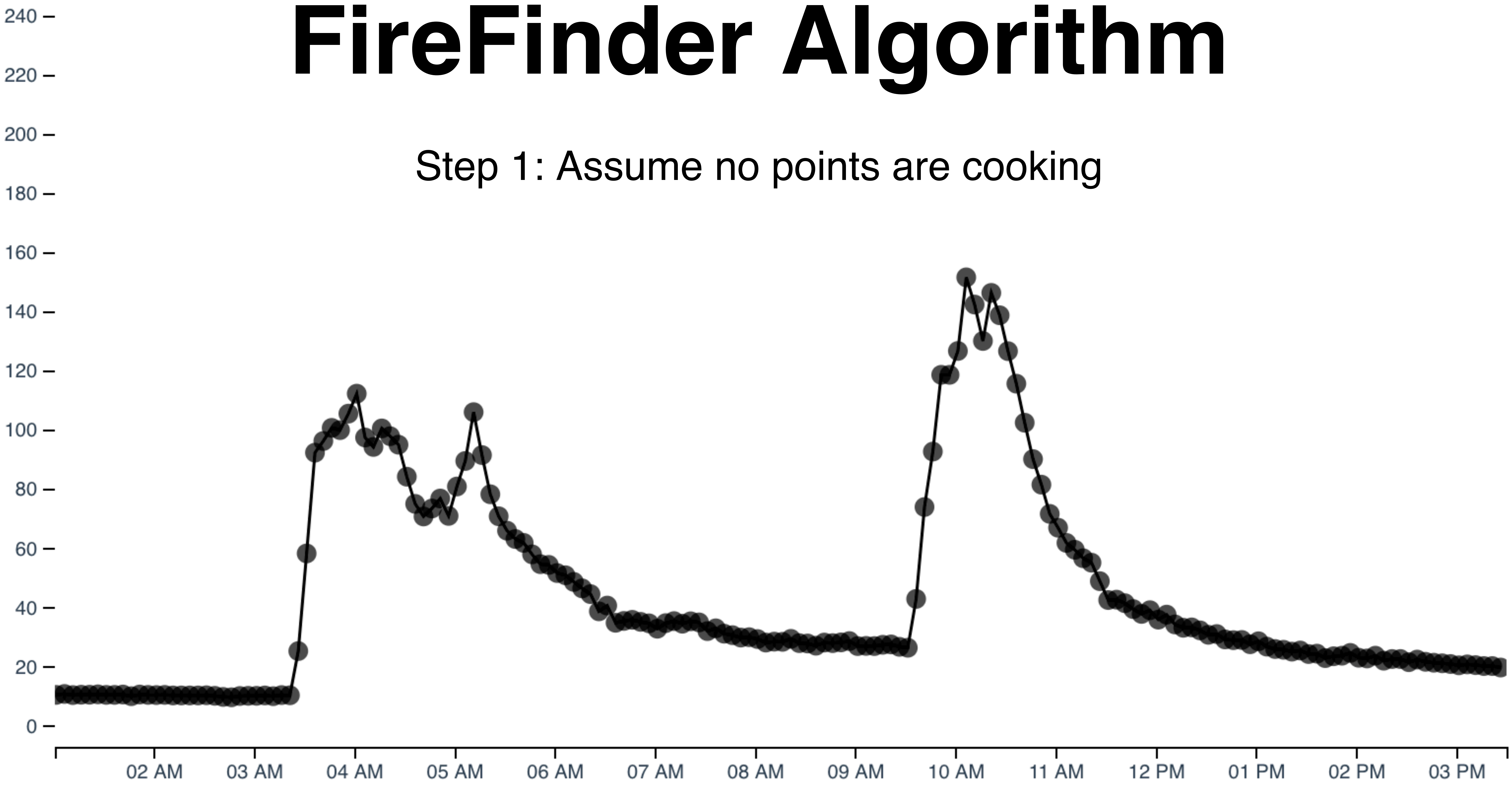
Machine Learning via SUMSarizer
- Available via TRAINSET and SUMSarizer
- Tuning: use graphical tools to build a training set, then train an ML model in R.


Search or jump to...
Pull requests
Issues
Marketplace
Explore
Geocene / sumsarizer
Watch 1
Star 1
Fork 2
Code
Issues 5
Pull requests 0
Actions
Projects 0
Wiki
Security
Insights
Settings
No description, website, or topics provided.
Edit
Manage topics
62 commits
1 branch
0 packages
0 releases
3 contributors
View license
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
daterdots Update README.md
Latest commit 71b6e1c on Oct 31, 2019
R
fix constant detector
6 months ago
inst/extdata
embed xgboost model in sumsarizer, make default
7 months ago
man
add functionality to process complete folders
7 months ago
tests
passing off vignette to jeremy for event-wise summaries
7 months ago
vignettes
move metadata from package to s3
6 months ago
.Rbuildignore
fix make check
12 months ago
.gitignore
vignette outlines
7 months ago

```r
firefinder_detector = function(data,
                               primary_threshold = 75,
                               min_event_temp = NULL,
                               min_event_sec = 5*60,
                               min_break_sec = 30*60,
                               ...) {

  primary_threshold <- as.numeric(primary_threshold)
  min_event_temp <- as.numeric(min_event_temp)
  min_event_sec <- as.numeric(min_event_sec)
  min_break_sec <- as.numeric(min_break_sec)
  setDT(data)
  data <- copy(data)
  max_run_length <- 100

  #CALCULATE FEATURES
  sample_interval <- get_sample_interval(data)
  sample_interval_mins <- sample_interval/60

  #make a column of 1st derivative (degC/minute)
  data[, difftemps := c(0, diff(value) / sample_interval_mins)]

  #make a column of delta timestamps
  data[, difftimes := c(as.numeric(diff(data$timestamp), units = "secs"), 0)]

  #look at whether or not most of the data coming up in the next
  #hour is negative slope or 100 data points, whichever is lower
  if (nrow(data) > 1) {
    data$quantile_difftemps = runquantile(data$difftemps,
                                          min(max_run_length,
                                              min(round(60/sample_interval_mins),
                                                  nrow(data))),
                                          probs = 0.8,
                                          align = 'right')
  } else {
    data$quantile_difftemps = NA
  }

  #RUN THE DECISION TREE

  #just assume there is no cooking to start
  data$event_raw = FALSE

  #define points that are likely to be cooking
  data[value > primary_threshold, event_raw := TRUE]

  #get rid of long runs of negative slopes
  data[quantile_difftemps < 0, event_raw := FALSE]

  #assume cooking for highly positive slopes
  data[difftemps > 2, event_raw := TRUE]

  #get rid of highly negative slopes
  data[difftemps < -1 * value / 500, "event_raw":= FALSE]

  #remove places with gaps longer than the sample interval
  data[difftimes > sample_interval, "event_raw":= FALSE]

  data[,"event_raw":=smooth_events(event_raw, sample_interval, min_event_sec, min_break_sec)]

  #remove events with very low cooking temps
  if(!is.null(min_event_temp)){
    data[,"event_num":=number_events(event_raw)]
    data[,"event_max":=max(value),by=list(event_num)]
    data[event_max<min_event_temp,"event_raw":=FALSE]
  }
  #remove events for data that is out of range and is probably an error
  data[!(data$value < 1000 & data$value > -50),"event_raw":=FALSE]

  return(data$event_raw)
}
```

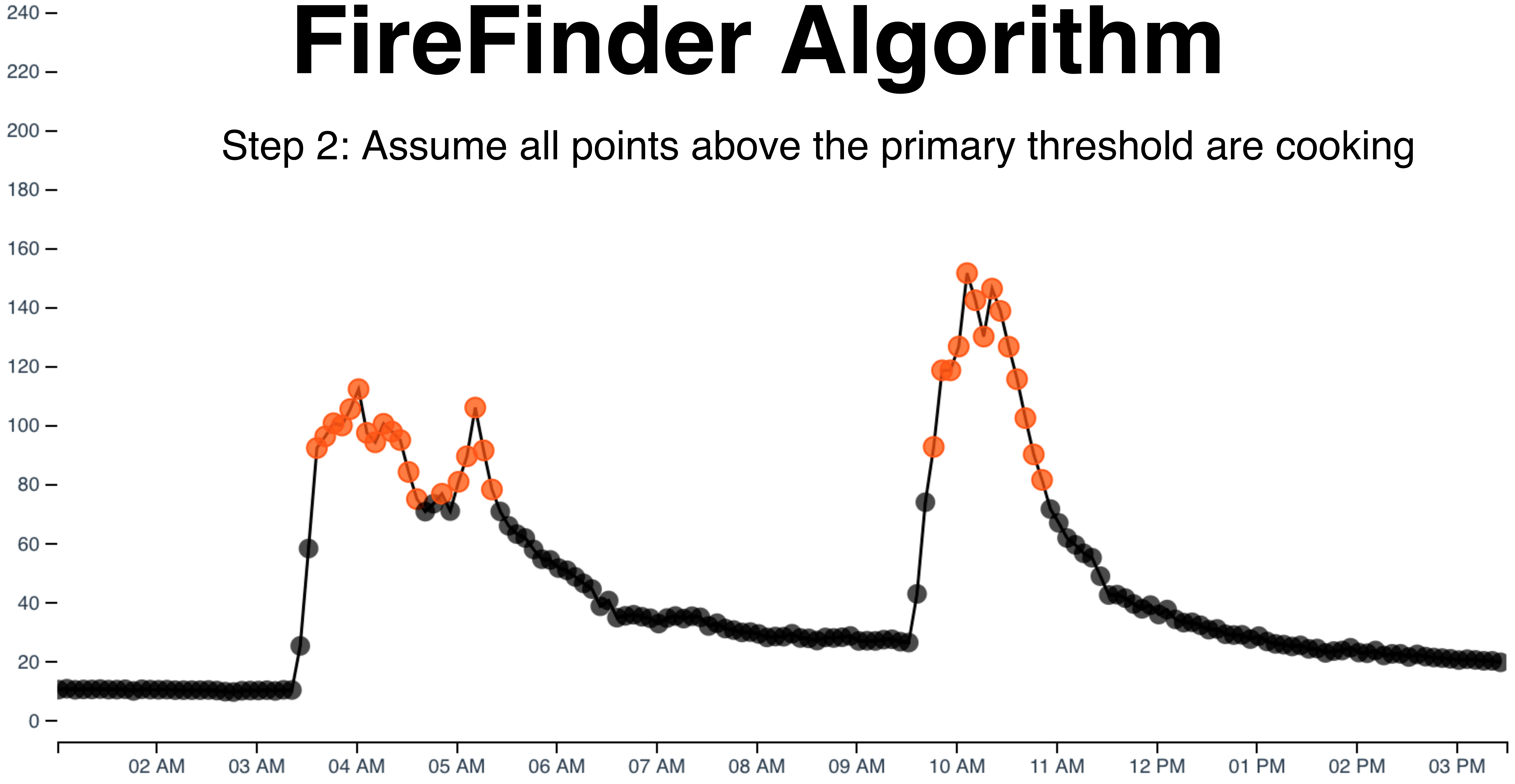
FireFinder Algorithm

Step 1: Assume no points are cooking



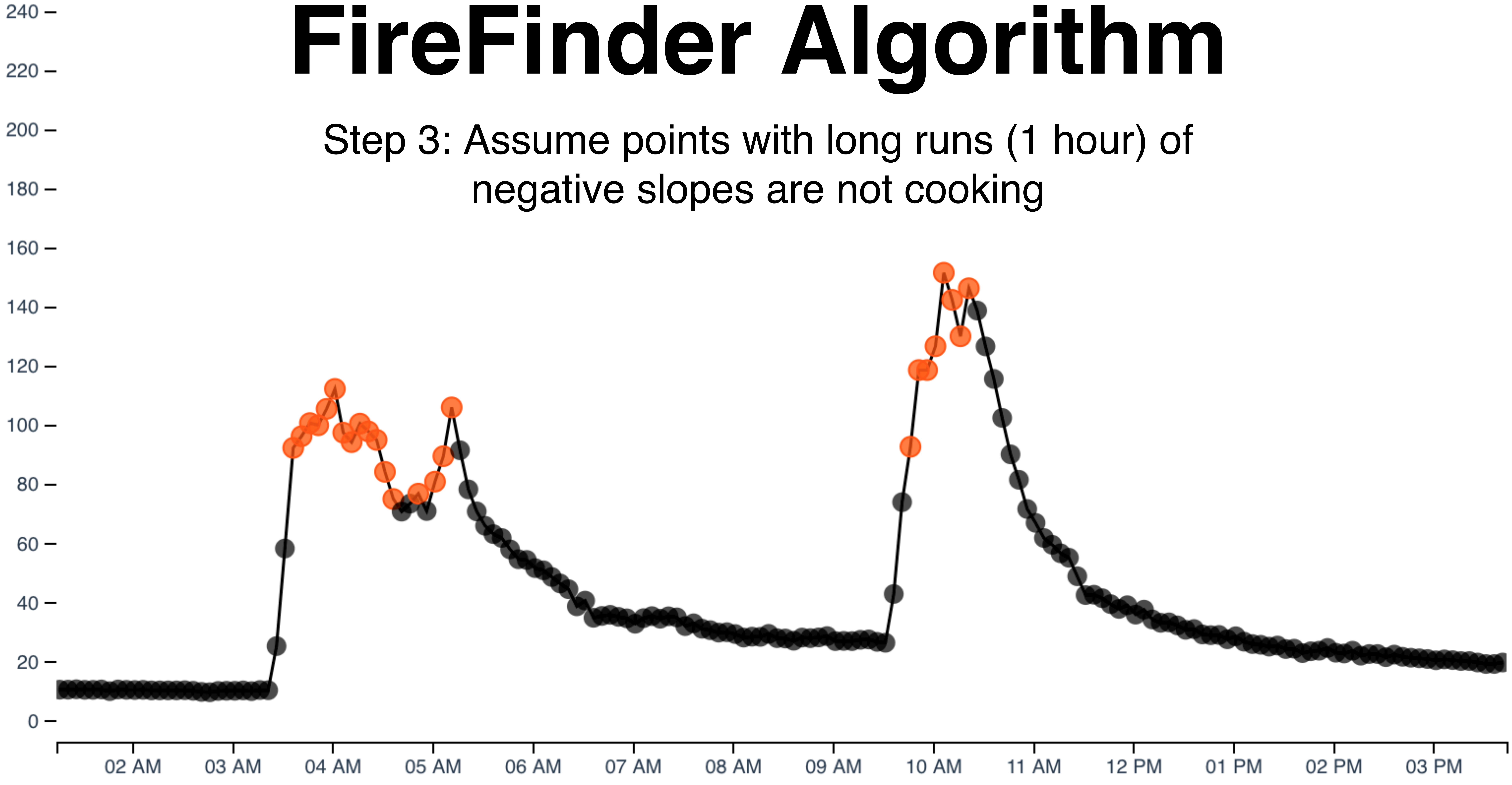
FireFinder Algorithm

Step 2: Assume all points above the primary threshold are cooking



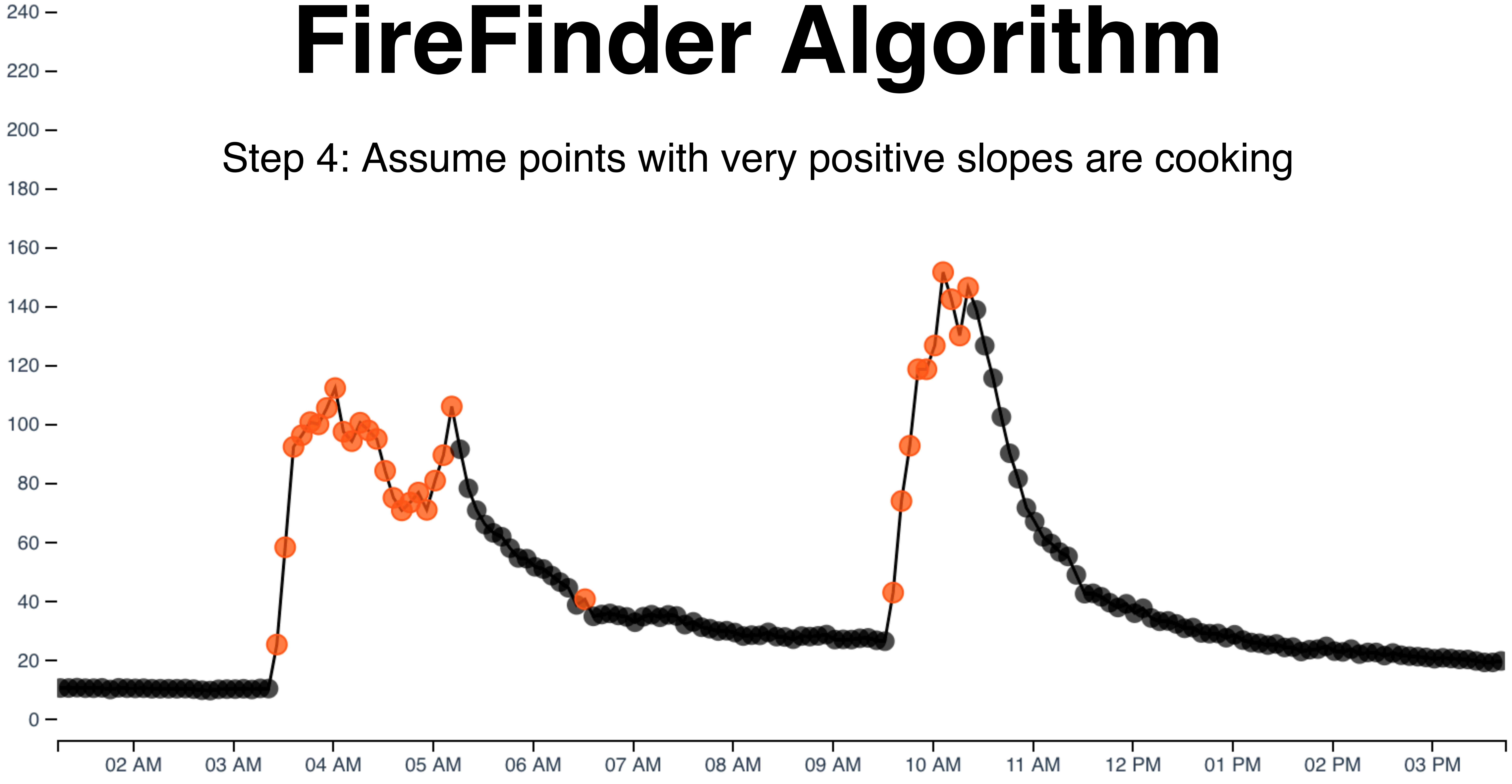
FireFinder Algorithm

Step 3: Assume points with long runs (1 hour) of negative slopes are not cooking



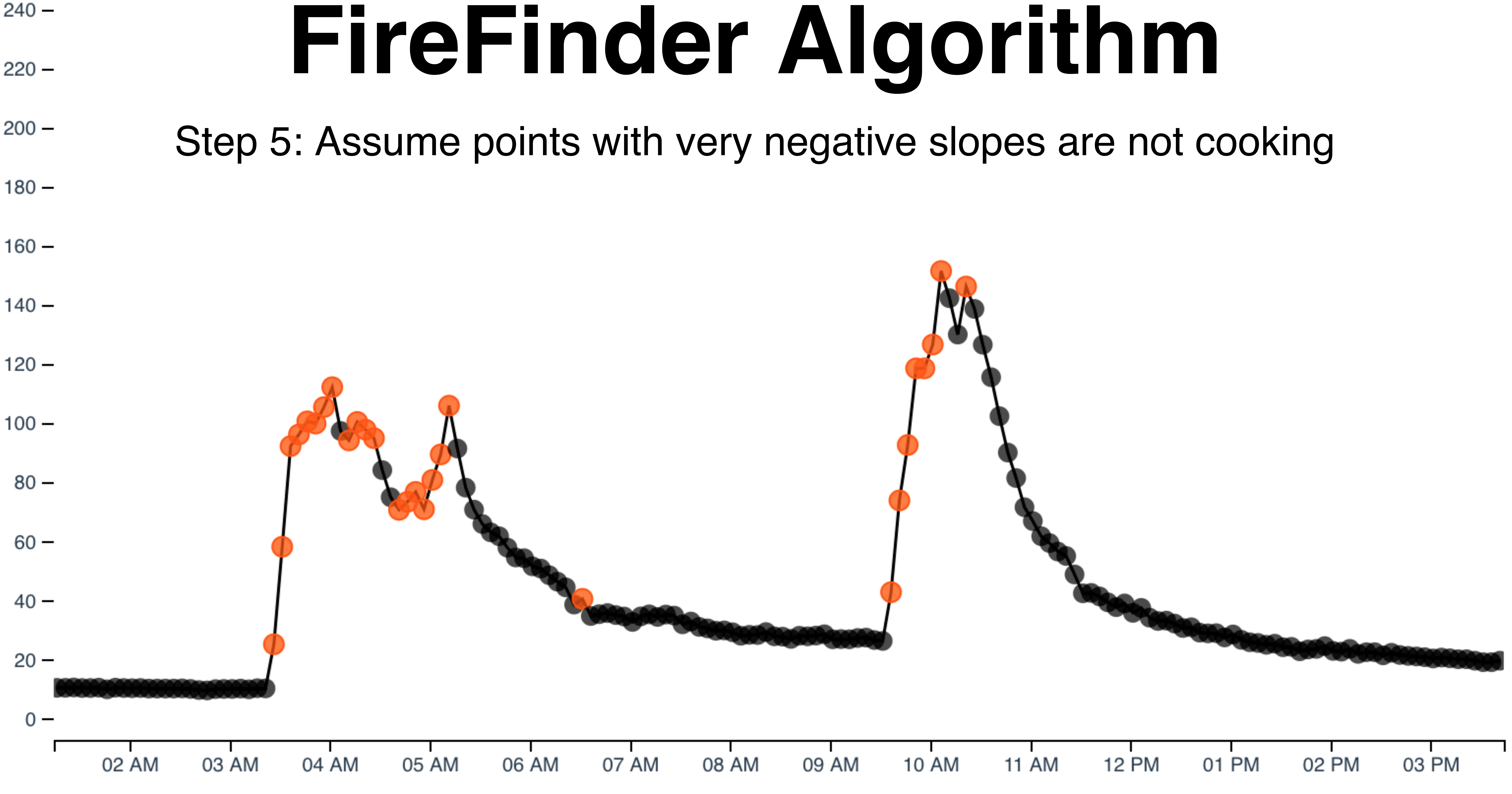
FireFinder Algorithm

Step 4: Assume points with very positive slopes are cooking



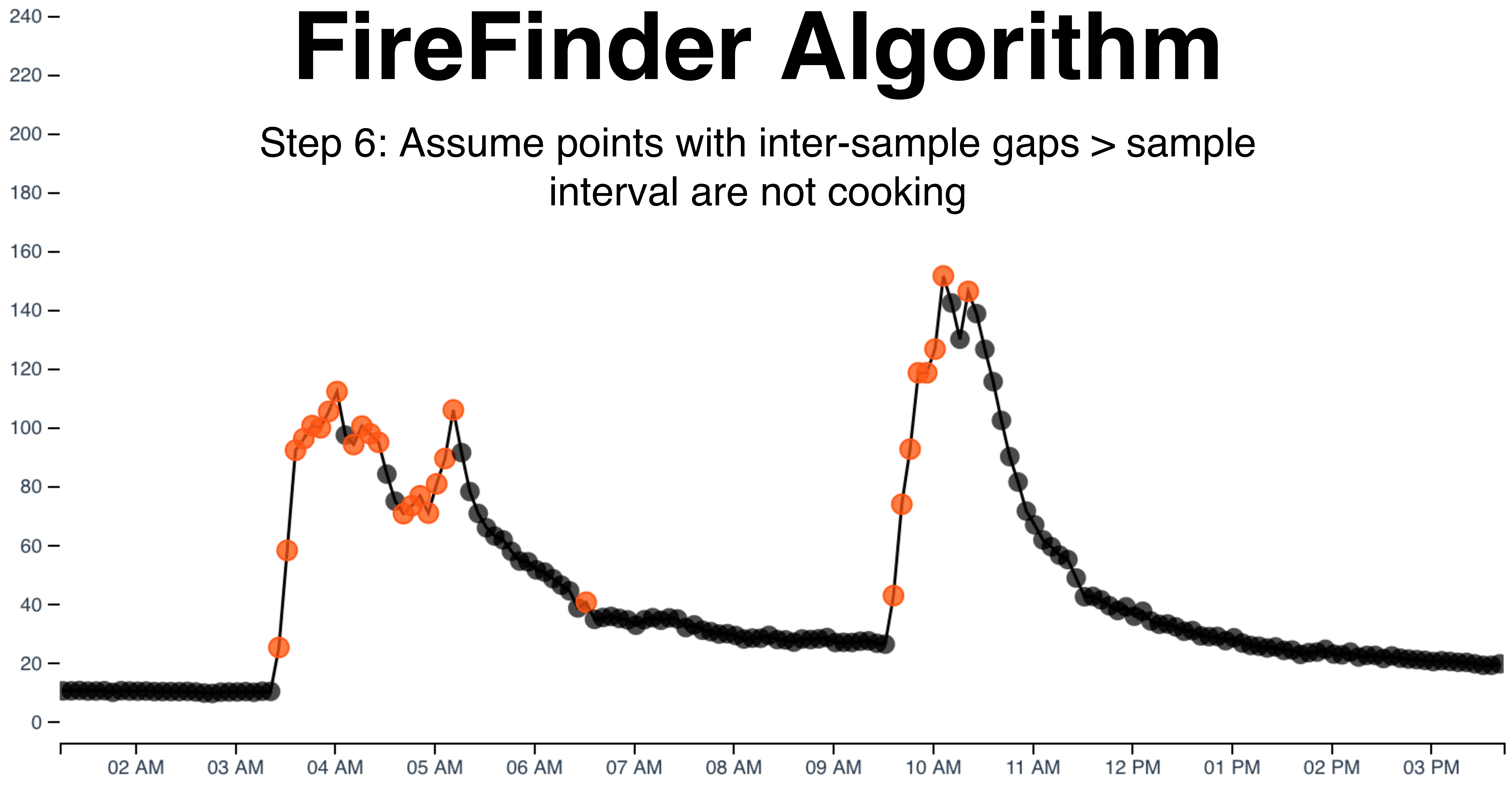
FireFinder Algorithm

Step 5: Assume points with very negative slopes are not cooking



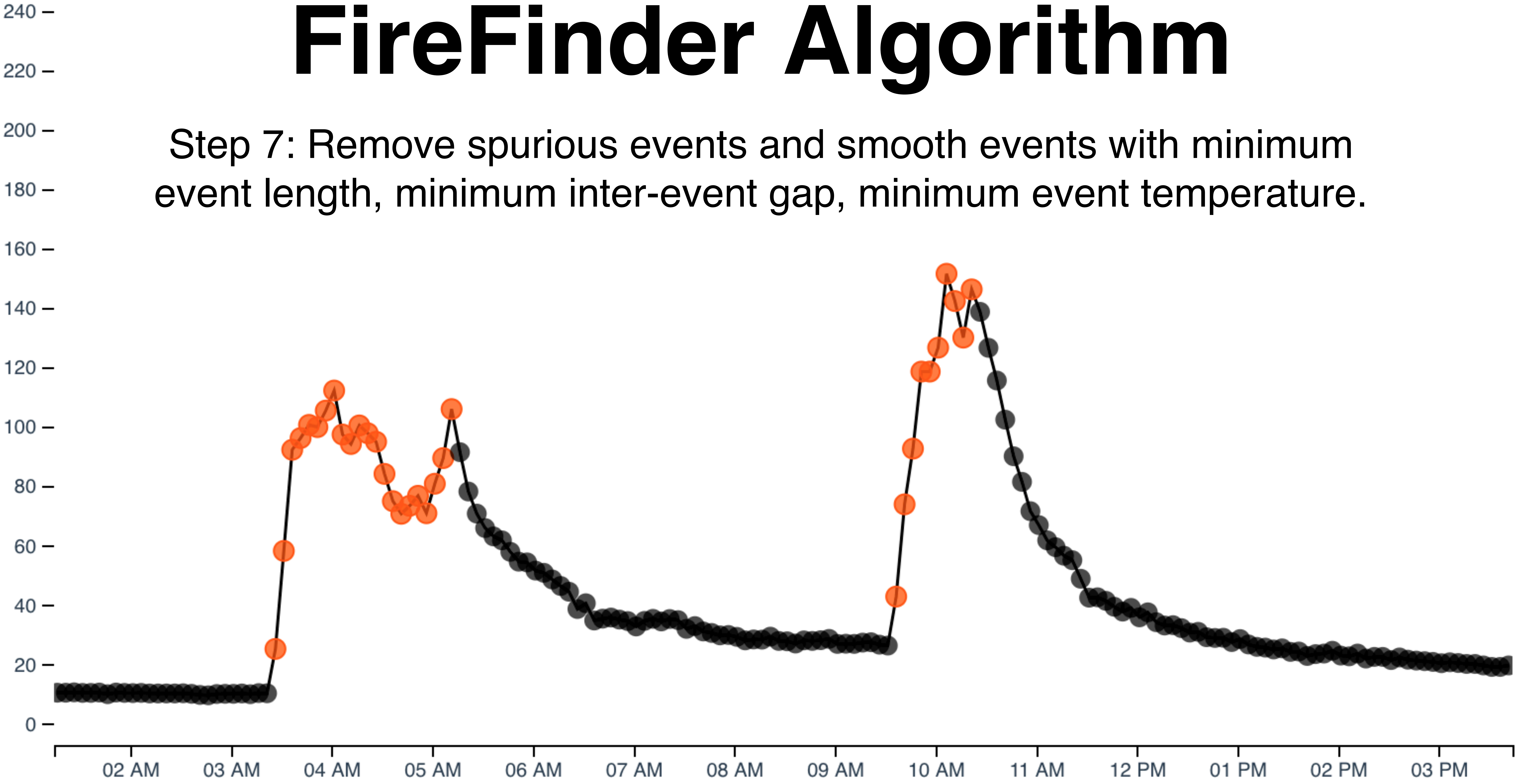
FireFinder Algorithm

Step 6: Assume points with inter-sample gaps > sample interval are not cooking



FireFinder Algorithm

Step 7: Remove spurious events and smooth events with minimum event length, minimum inter-event gap, minimum event temperature.



Tradeoffs

Better algorithms typically have better sensitivity and specificity, but not necessarily better bias!

Better algorithms are much more computationally demanding and require specialized skillsets.

Keep in mind: it’s not about whether a cooking event started at 12:00 or 12:10, it’s about what important insights your dataset reveals.

Understand your question: do you *really* need to know exactly how many times per day someone cooked, or do you actually need to know which cookstove is used the least/most?

Conclusion

FireFinder is a simple deterministic algorithm for detecting cooking events.

You can find the code at github.com/geocene/sumsarizer.

FireFinder has been demonstrated to work well on a wide variety of cookstove types.

You can use FireFinder for free at studies.geocene.com.