Toward automated quality control for hydro-meteorological weather station data

Tom Dietterich Tadesse Zemicheal

Oregon State UNIVERSITY

Download the Python Notebook

- https://github.com/tadeze/dsa2018

Outline

- TAHMO Project
- Sensor Network Quality Control
  - Rule-based methods
  - Probabilistic methods
  - SENSOR-DX approach
- Exercises
  - Anomaly detection for temperature, relative humidity, and atmospheric pressure
  - Mixture regression model for precipitation

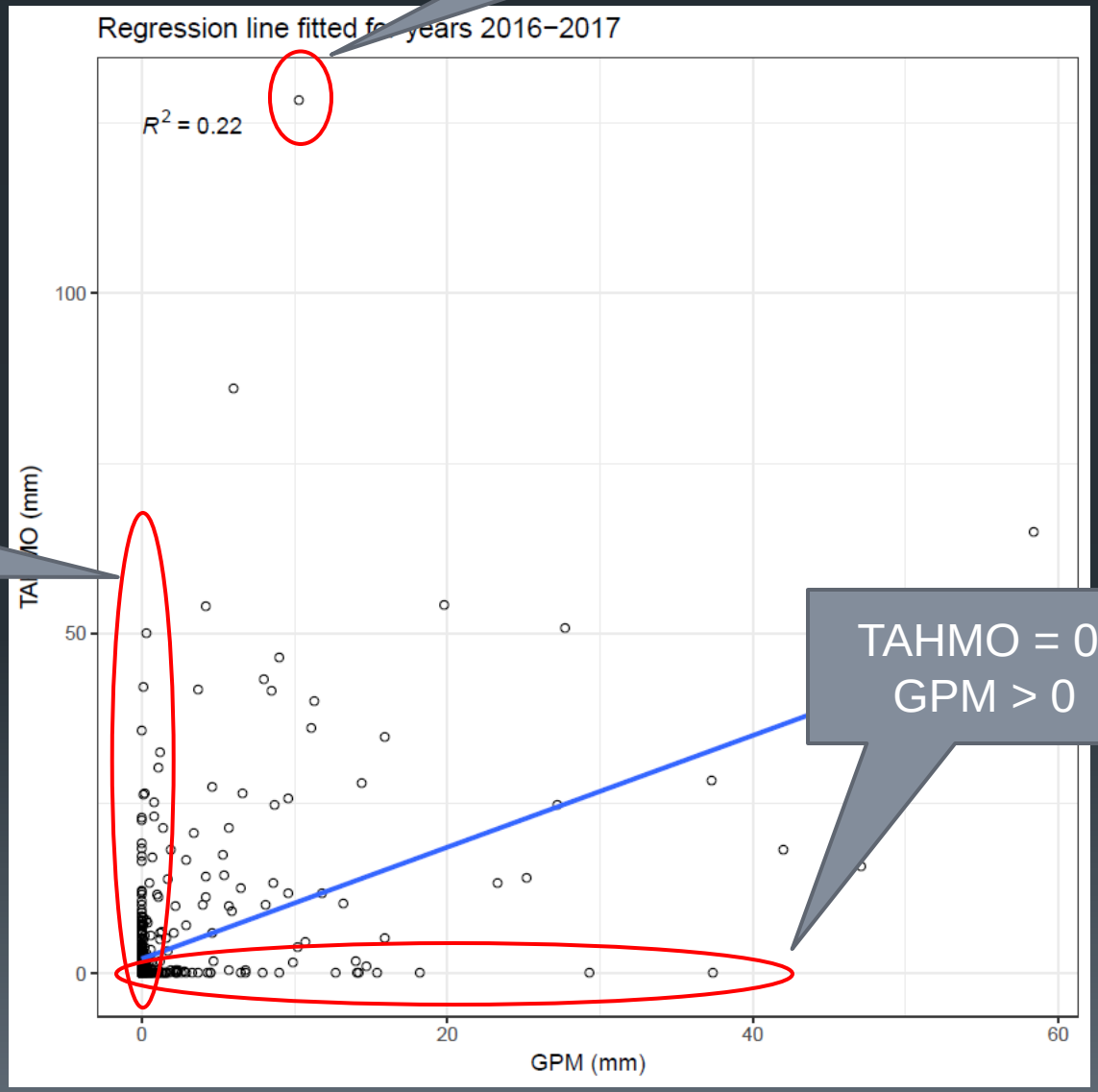
TAHMO: Motivation

- Africa is very poorly sensed
  - Only a few weather stations reliably report data to WMO (blue points in map)
  - Poor sensing ➔No crop insurance ➔Low agricultural productivity

- TAHMO Goal:
  - Make Africa the best-sensed continent & improve agriculture
  - Self-sustaining non-profit company

Do we need ground sta

- Scatterplot of precipitation estimate from satellite (NASA GPM) versus TAHMO station at South Tetu Girls High School



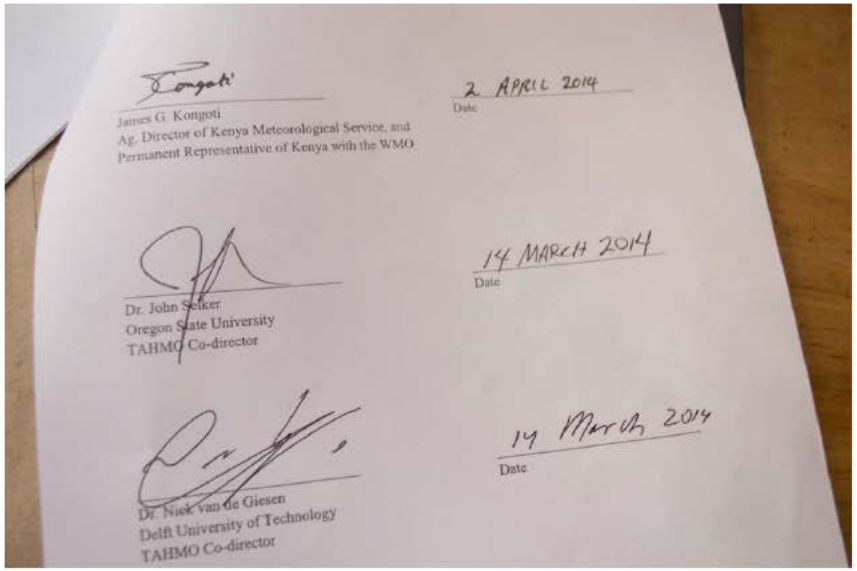
Business Plan

- Negotiate Memoranda of Understanding (MOUs) with each country in Sub-Saharan Africa
- Raise funds (gifts and grants) to develop and deploy weather stations
- Operating funds provided by selling the data
  - Free access for
    - The meteorological agency in each country
    - Education
    - Research

Memoranda of Understanding (MoUs)

MoU's
Kenya
Ghana
Malawi
Benin
Togo
Mali
Burkina Faso
Uganda
Ethiopia
Tanzania
Nigeria
South Africa

Close to complete
Rwanda
Ivory Coast
Cameroon
Zambia
Senegal



Finances

- Deployment cost
  - 20,000 stations x $2000 per station = $40M
- Operating cost
  - $600/stations/year = $12M
- Weather data market
  - Estimate $40,000M/year

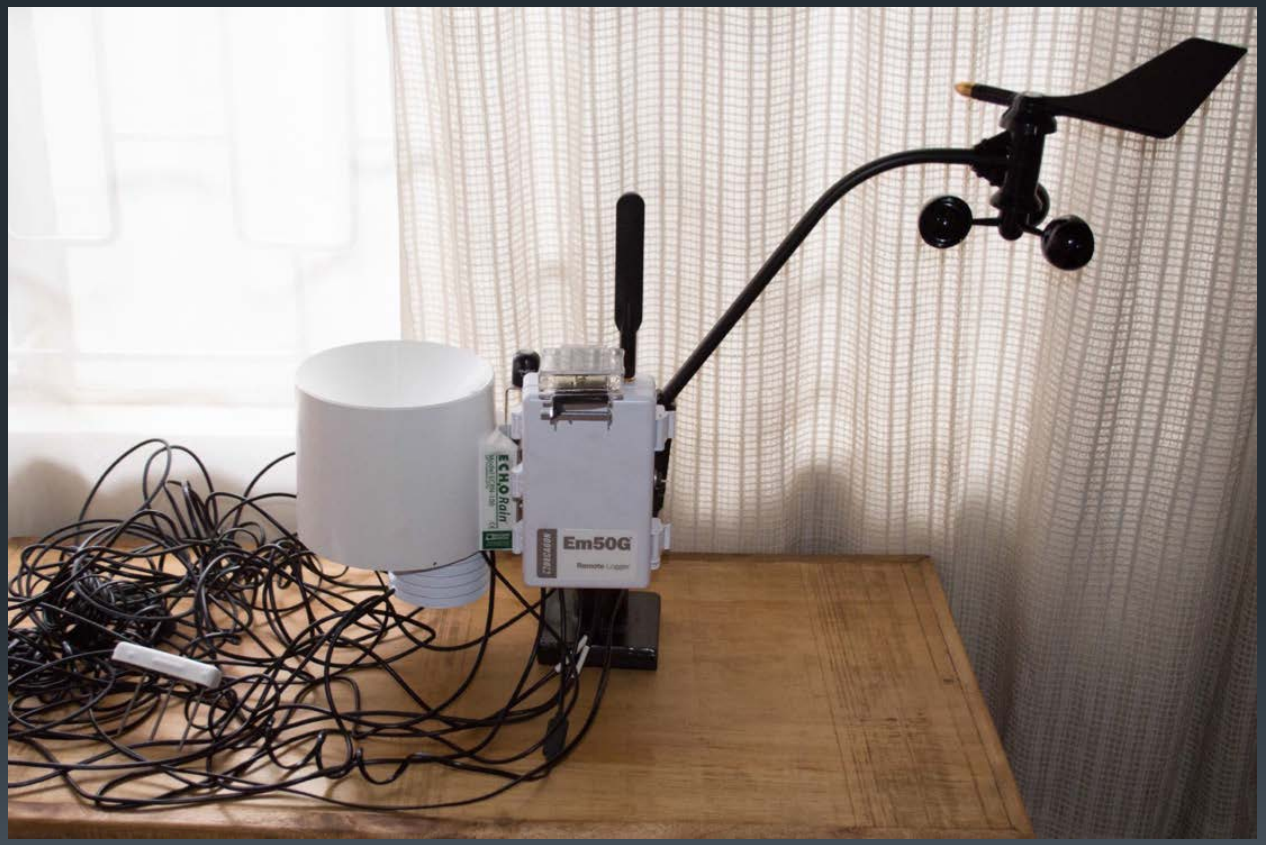
- Status: >500 stations deployed
  - Funding from USAID, UN, EU, IBM
  - School2School program

Technology

- Weather Stations
- Automated Quality Control

Generation 1 Weather Station

- cables
- 3 moving parts
- 5 components



Generation 3 station

- No moving parts
- No cables
- Two components

Generation 3 Features

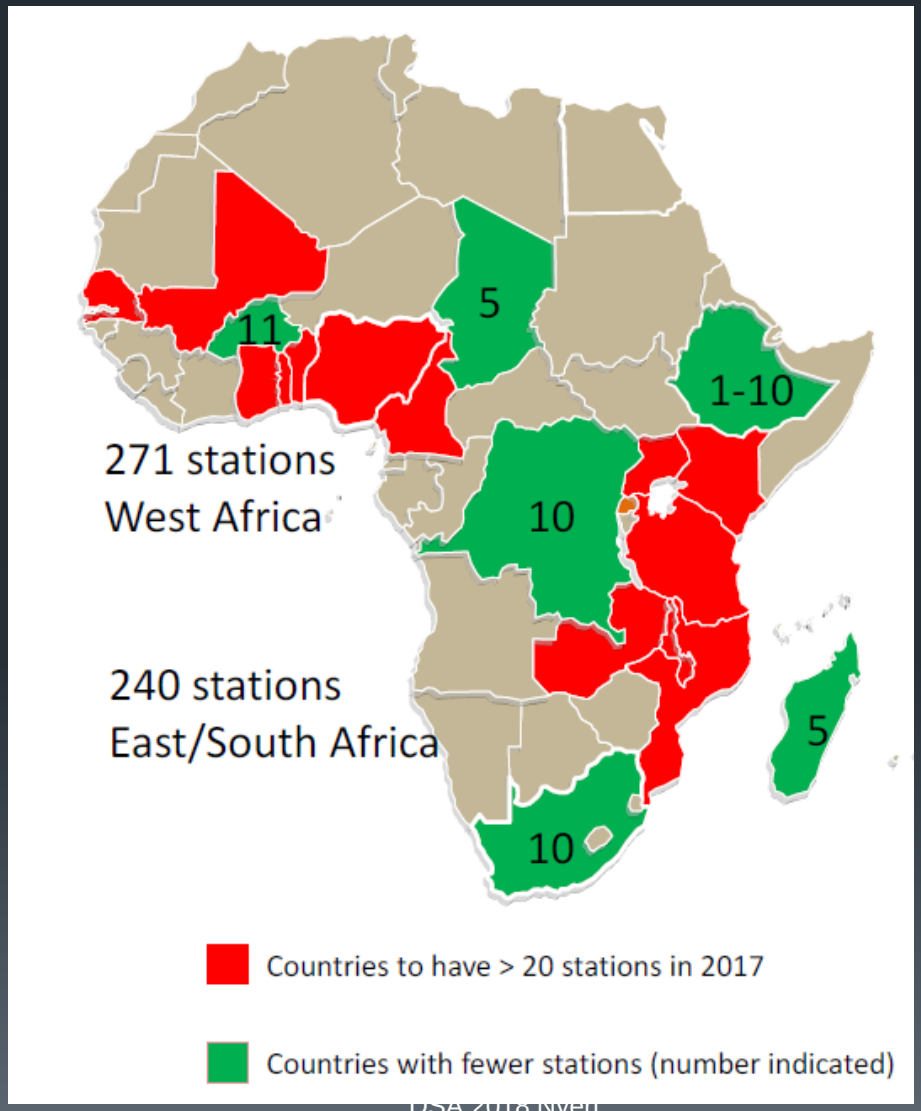
- Solar power
- 6-month reserve battery
- GSM/GPRS radio
- GPS & Compass
- Temperature (3 ways)
- Relative Humidity
- Accelerometer
- Sonic wind
- Drip-count rain
- Shortwave solar radiation
- Barometer
- Lightning detector
- 5 open sensor ports: soil moisture etc.

Station Placement and Security

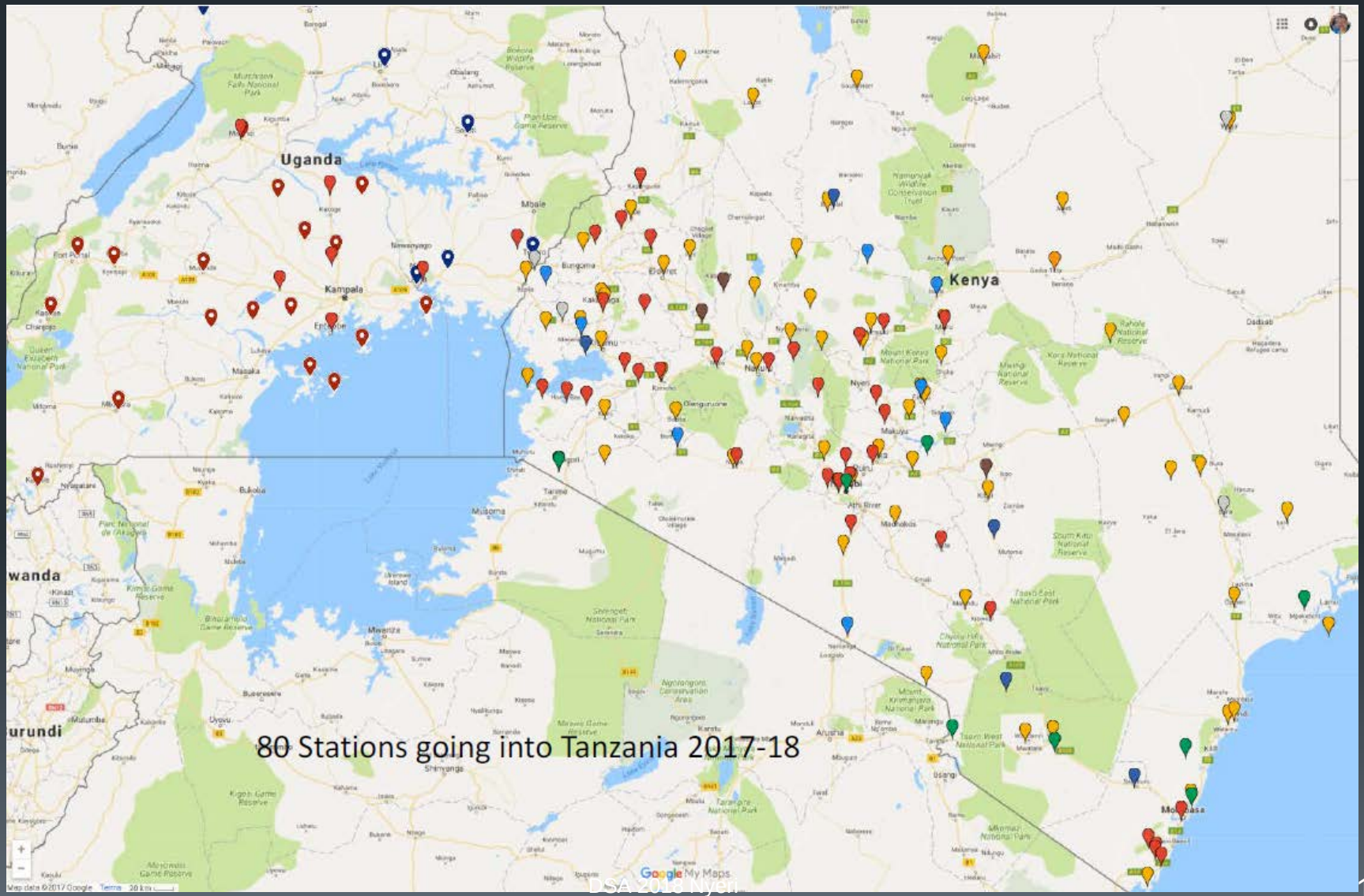
- General strategy: Place stations at schools
  - Teacher monitors the station and clean it regularly
  - Use the station as an educational resource
    - TAHMO provides educational materials and lesson plans
    - Students can download data and analyze it

- School2School Program
  - Schools in US and Canada can purchase two stations
    - One for their school
    - One for a school in Africa
    - Students learn about their partner school starting with the weather

Current Status



Uganda and Kenya



Quality Control

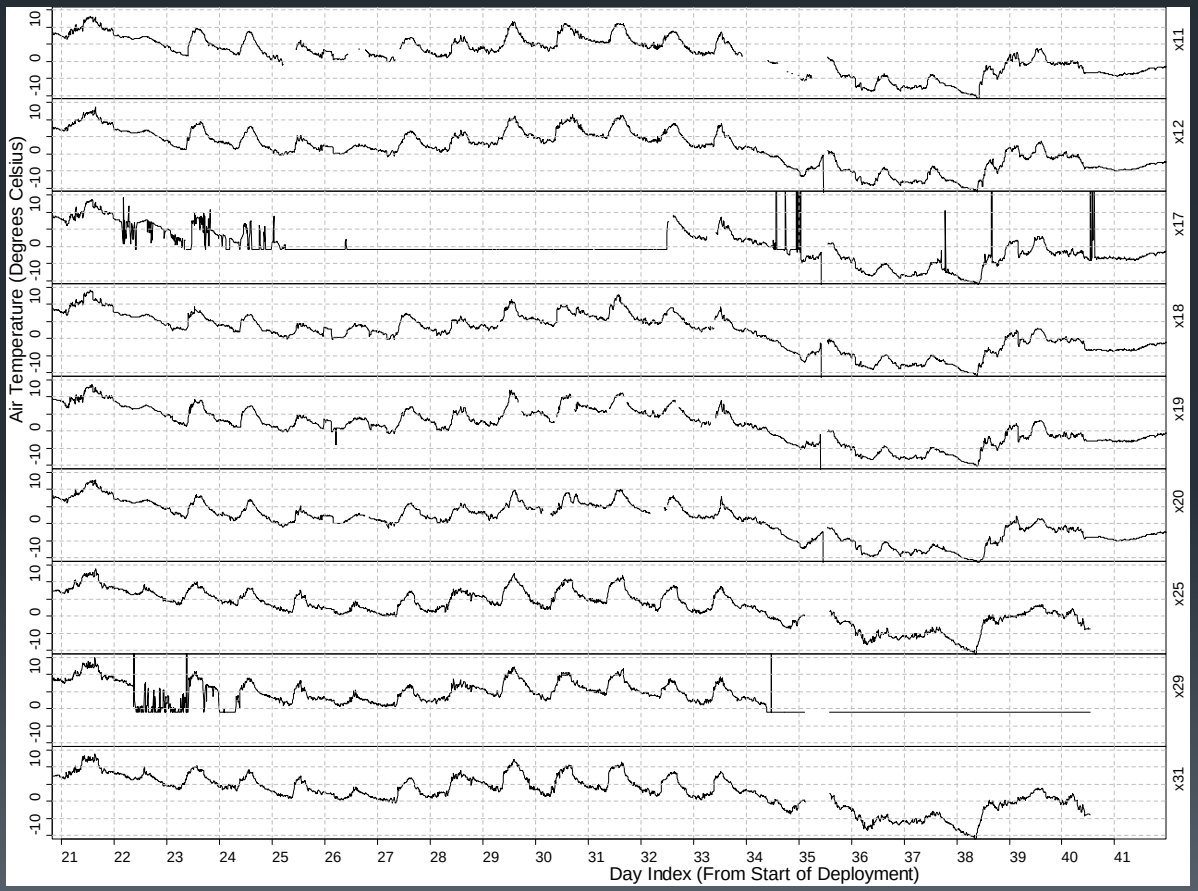
- Weather Sensors Fail
  - Solar radiation sensor gets dirty
  - Wind sensors (anemometers) get dirty or blocked
  - Rain gauge becomes obstructed
  - Novel failures occur often
- Battery Failure
  - Poor cellular telephone connectivity

Ant Infestation

Wasps in the Anemometer

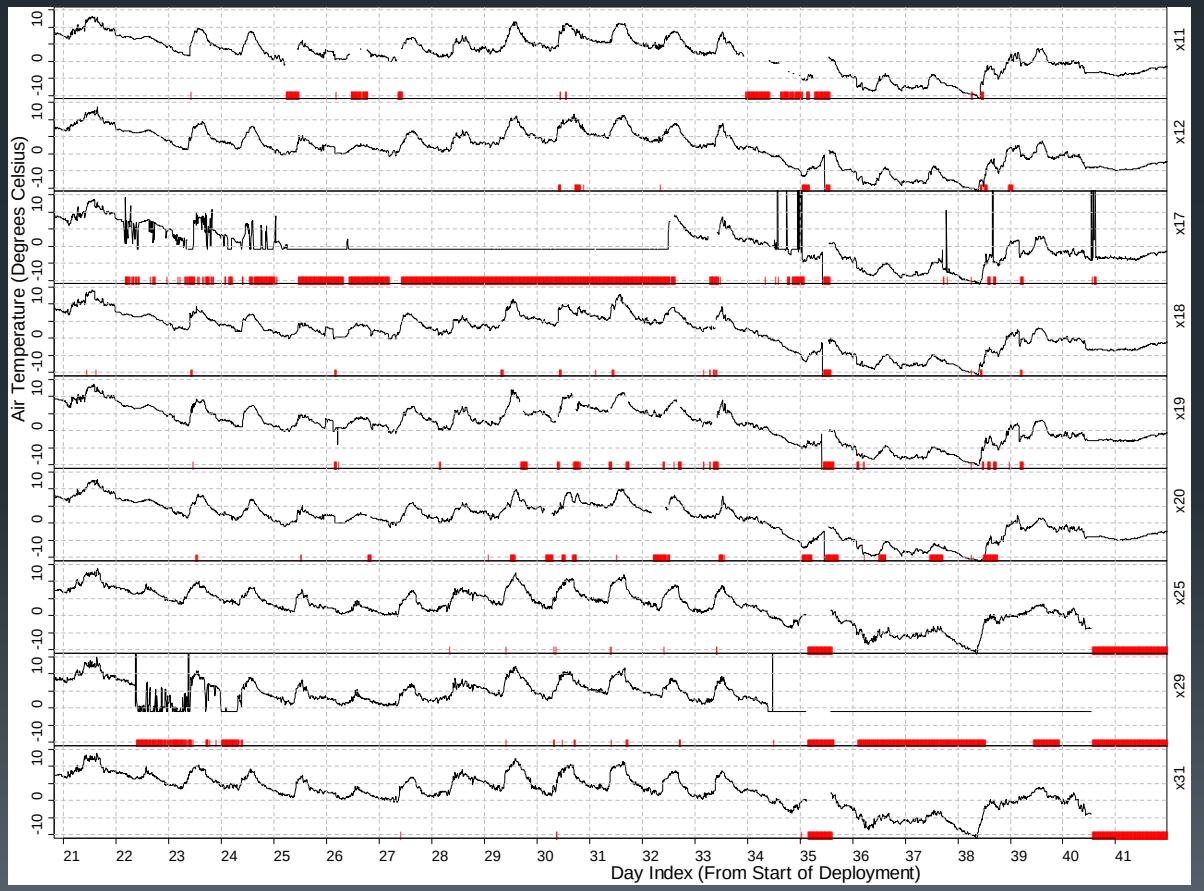
Data Quality Control

- An ideal method should produce two things given raw data:



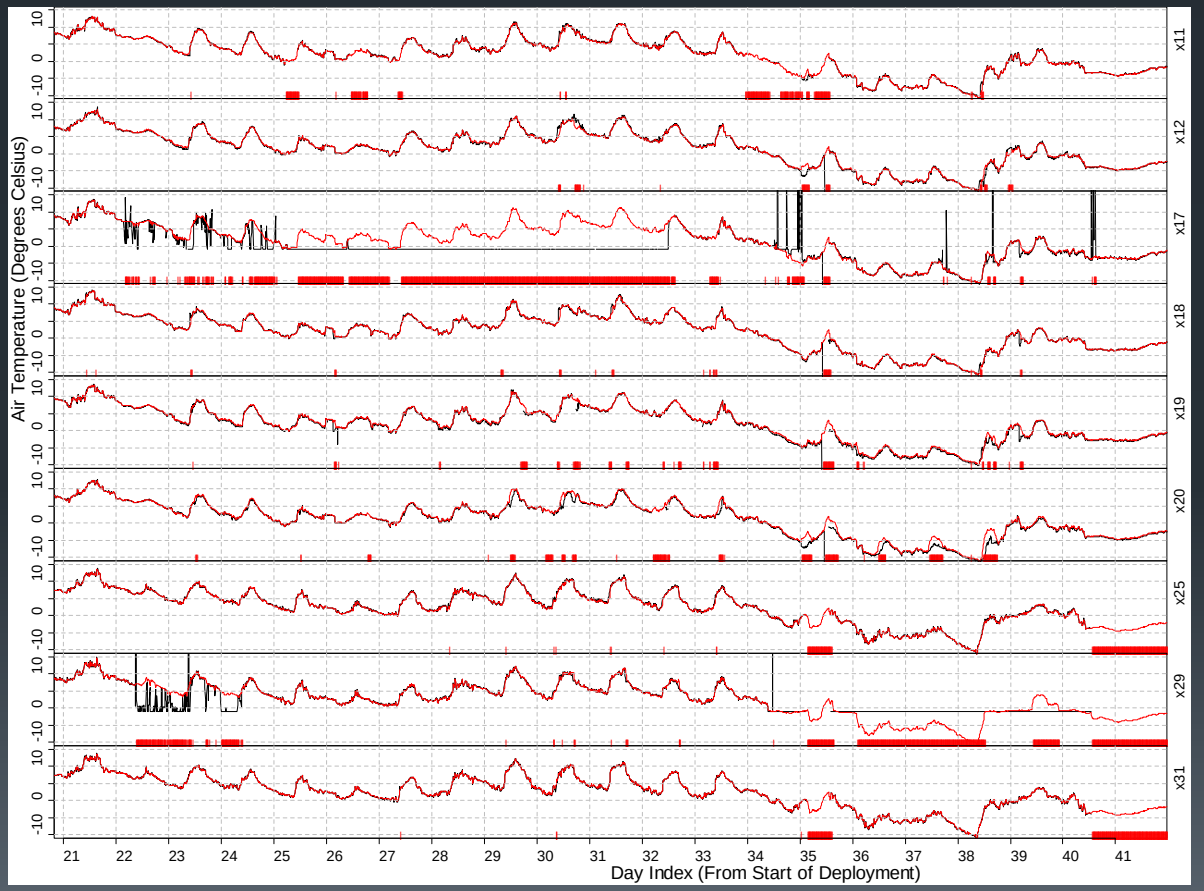
Data Quality Control

- An ideal method should produce two things given raw data:
  - A label that marks anomalies



Data Quality Control

- An ideal method should produce two things given raw data:
  - A label that marks anomalies
  - An imputation of the true value (with some confidence measure)



Dereszynski &, Dietterich, ACM TOS 2011.

Existing Approaches to Quality Control

- Manual Inspection (used at H J Andrews LTER)
- Complex Quality Control (OK Mesonet)
- Probabilistic Quality Control (Rawinsonde Network)

- All of these require large amounts of expert time
- TAHMO is much larger than these networks
- TAHMO will be larger than the networks used by the US National Weather Service
- We need a fully-automated QC method

Existing Methods 1: Complex Quality Control

- Rule-based approach that raises an alarm if a rule is violated
  - Step test: $x_{t+1} - x_t < \theta_1$
  - Flatline test: # of consecutive steps where $x_{t+1} = x_t$ must be $< \theta_2$
  - Buddy test: $|x_t - y_t| < \theta_3$ for two identical sensors x and y
  - ...

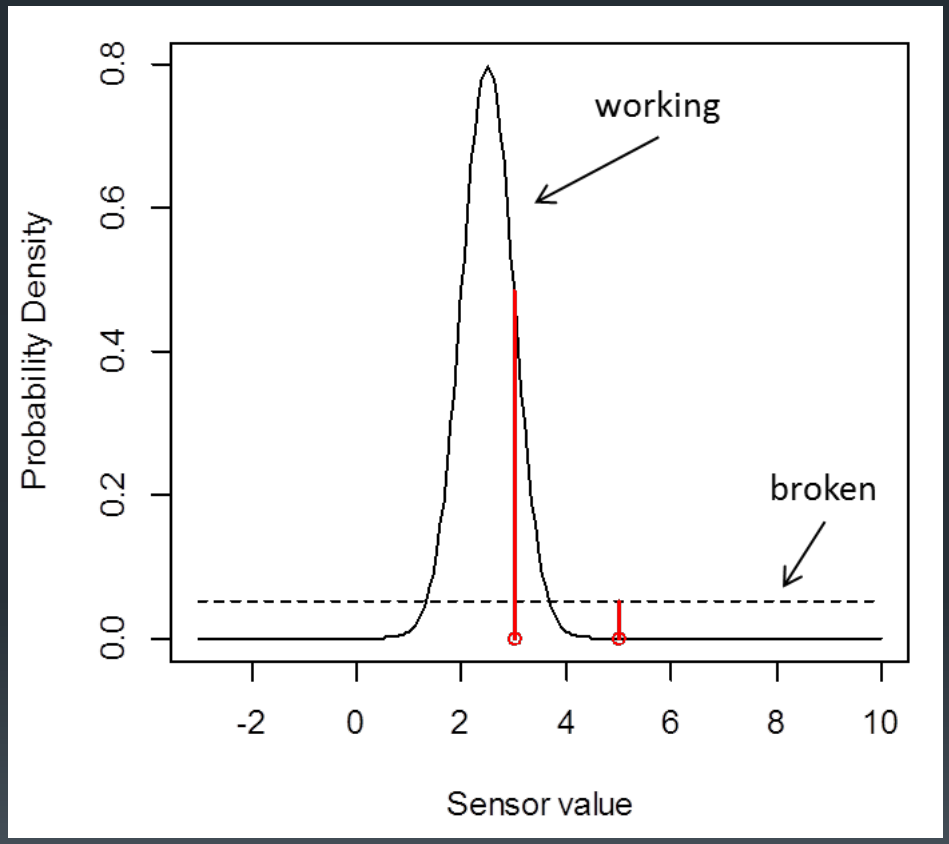
Complex Quality Control

- Problems:
  - No unifying principles
  - Considers each variable separately
  - Hard to maintain
- Advantages:
  - Practical
  - Easily extended by adding new rules
  - Does not require a model of the signals

Probabilistic Quality Control

- Define s_t to be the state of the sensor at time t

  $s_t \in \{0,1\}$ where 0 = OK and 1 = Broken
- $P(x_t|s_t = 0)$ is the "normal" probability density for the sensor
- $P(x_t|s_t = 1)$ is the "broken" probability density for the sensor
- $P(s_t)$ is the prior over sensor states
- Query: $P(s_t|x_t) = \frac{P(s_t)P(x_t|s_t)}{P(x_t)}$

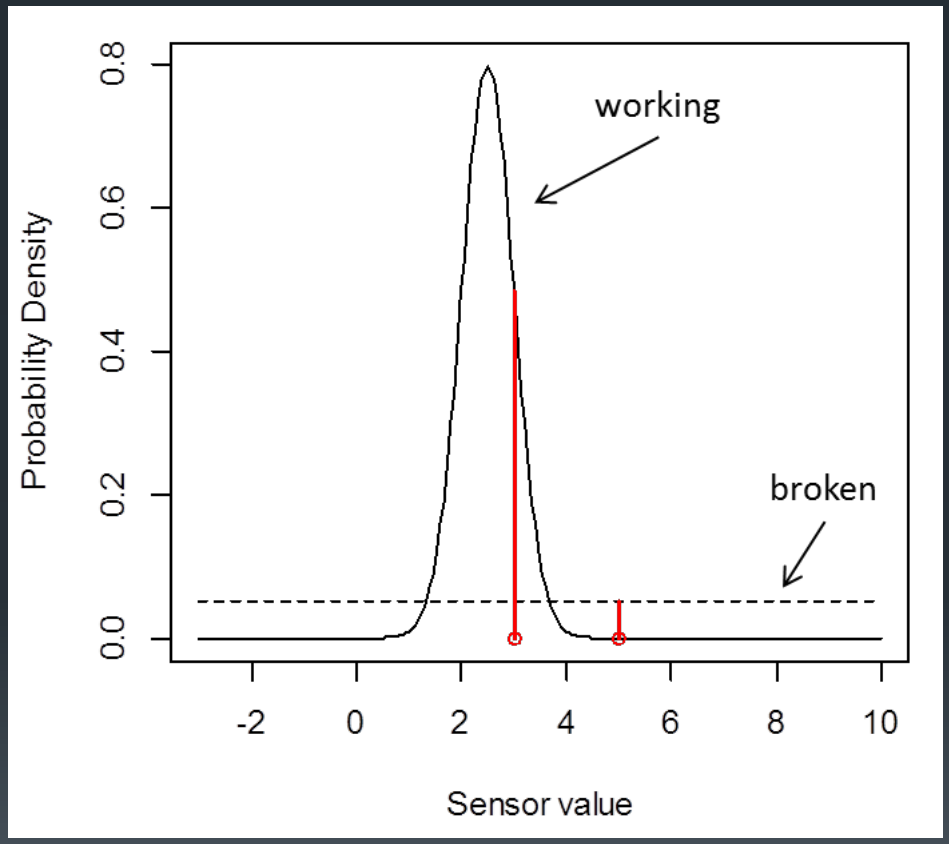


Challenge: Modeling the Broken distribution

- Modeling $P(x|s=0)$
  - Lots of data; virtually all data points are from this case
  - However, the distribution may still be complex
- Modeling $P(x|s=1)$ is very difficult
  - Bad sensor values are rare, so little data
  - Sensors break in novel ways, so hard to predict the sensor readings

Hack: “Junk Bucket” Distribution

- Assume $P(x_t|s_t = 1)$ is the uniform distribution
- This is equivalent to setting a threshold on $P(x_t|s_t = 0)$
- Hard to do this well
- Hard to model multiple sensors



Our Idea:
Apply Anomaly Detection Methods

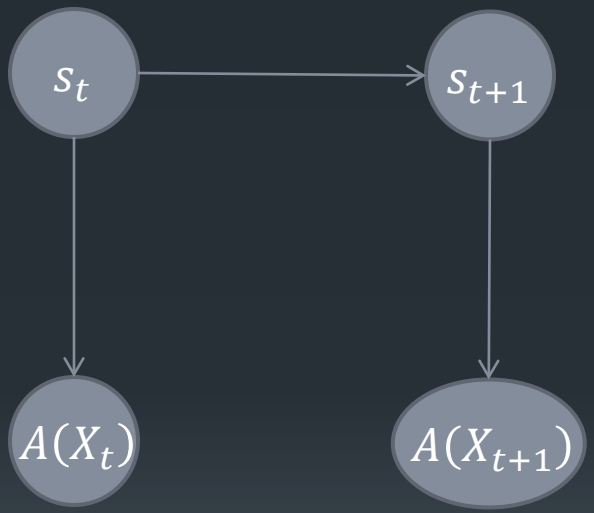
- Suppose we could assign an anomaly score $A(x_t)$ to each observation x_t
  - Scores near 0 are “normal”
  - Scores > 0.5 are “anomalous”
- Learn a probabilistic model of the anomaly scores instead of the raw signals

$$P(A(x_t)|s_t)$$

Basic Configuration

Observe X_t
Compute $A(X_t)$
Compute $\arg\max_{s_t} P(s_t)P(A(X_t)|s_t)$

Cool Things We Can Do: Model Persistence of Sensor State



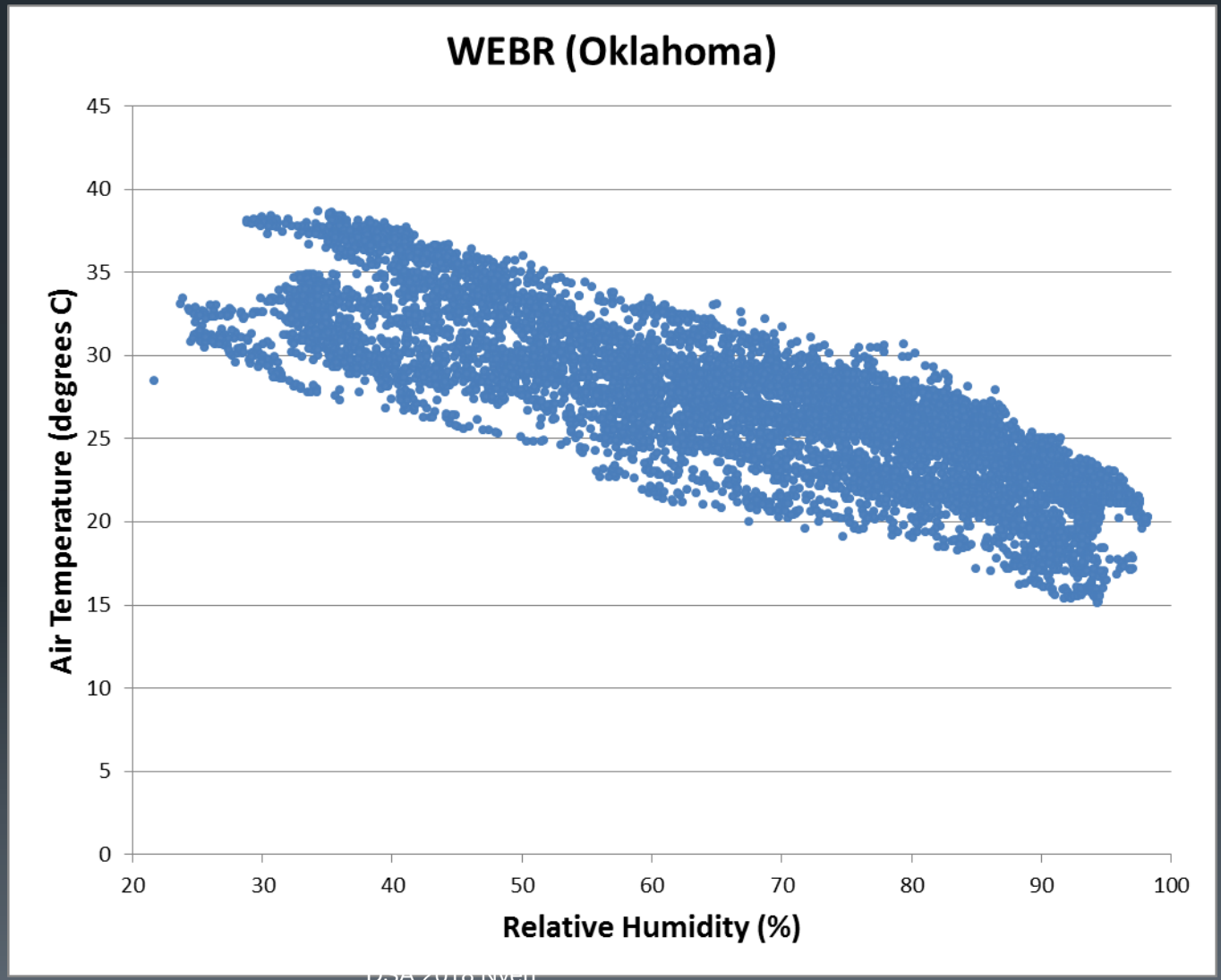
$P(s_{t+1}|s_t)$ encodes persistence of sensor state

- Sensors that are working usually continue working
- Sensors that are broken usually stay broken (until cleaned/repaired)

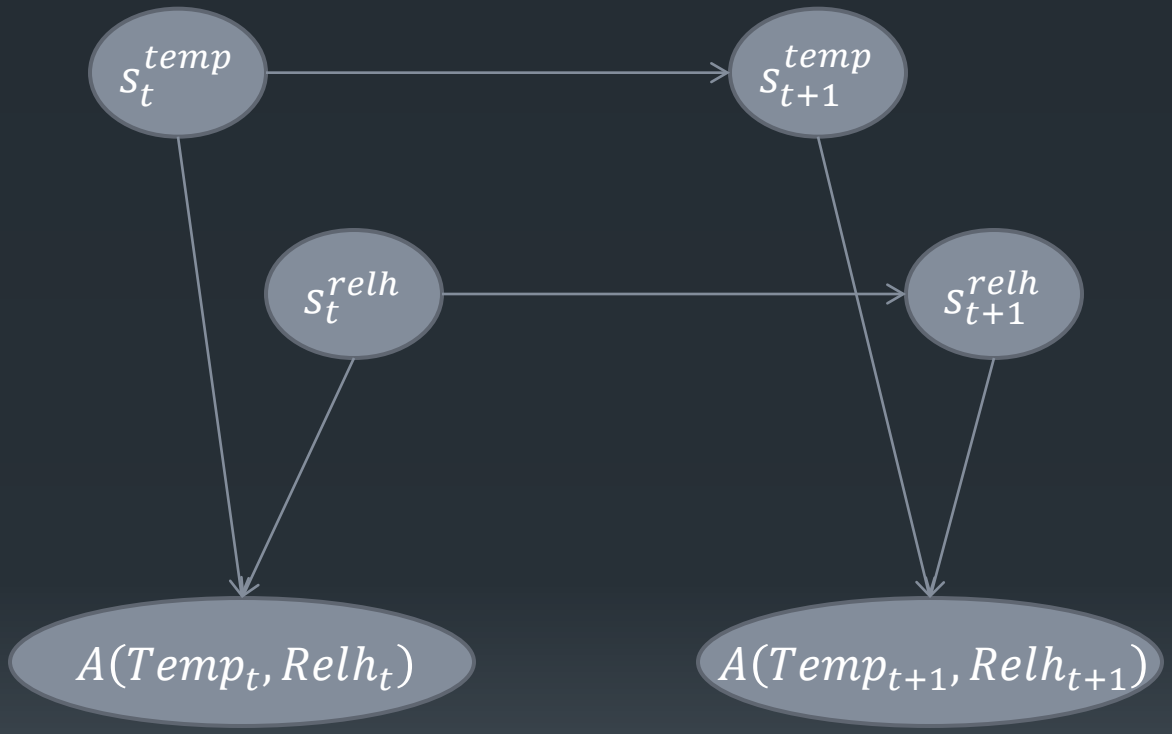
Cool Things We Can Do #2: Model the Joint Distribution of Sensors

Example: Temperature and Relative Humidity are strongly (negatively) correlated

July 2009



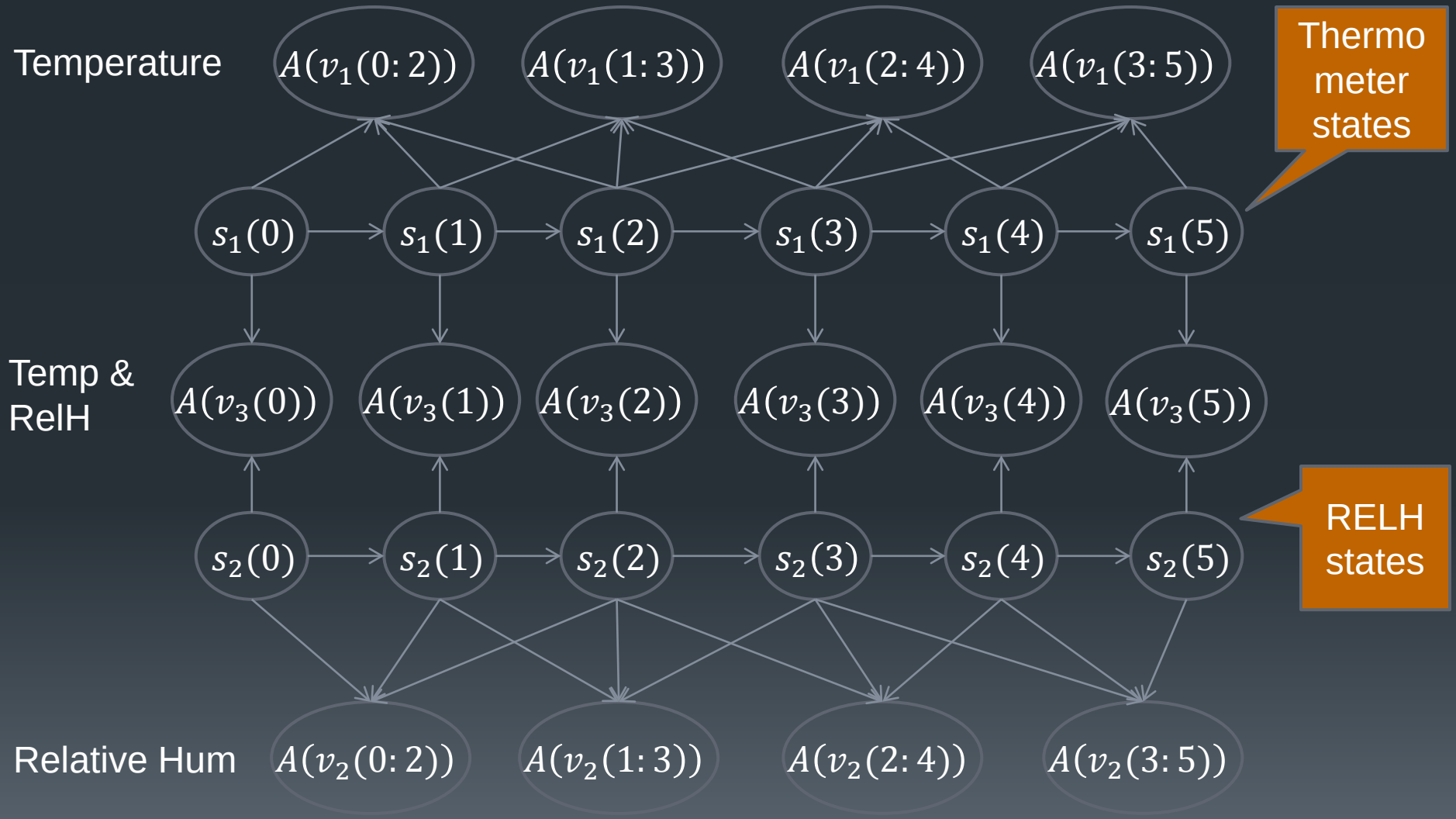
Joint Anomaly Detection



SENSOR-DX: Multiple View Approach

- Capture joint distribution over time and space
  - Single sensor over K time steps
    - $A(x_{t-K+1}, x_{t-K+1}, \dots, x_{t-1}, x_t)$ captures this distribution
  - Rate of change of sensor signals
    - $A(X_t - X_{t-1})$ is like a "step test" in CQC
  - Differences between nearby weather stations
    - $A(X_t(\ell_1) - X_t(\ell_2))$
  - Difference between current value and value predicted from spatial neighbors
    - $A(x_t(\ell) - f(x_t(\ell'_1), \dots, x_t(\ell'_k)))$

Diagnostic Model



Run Time Quality Control

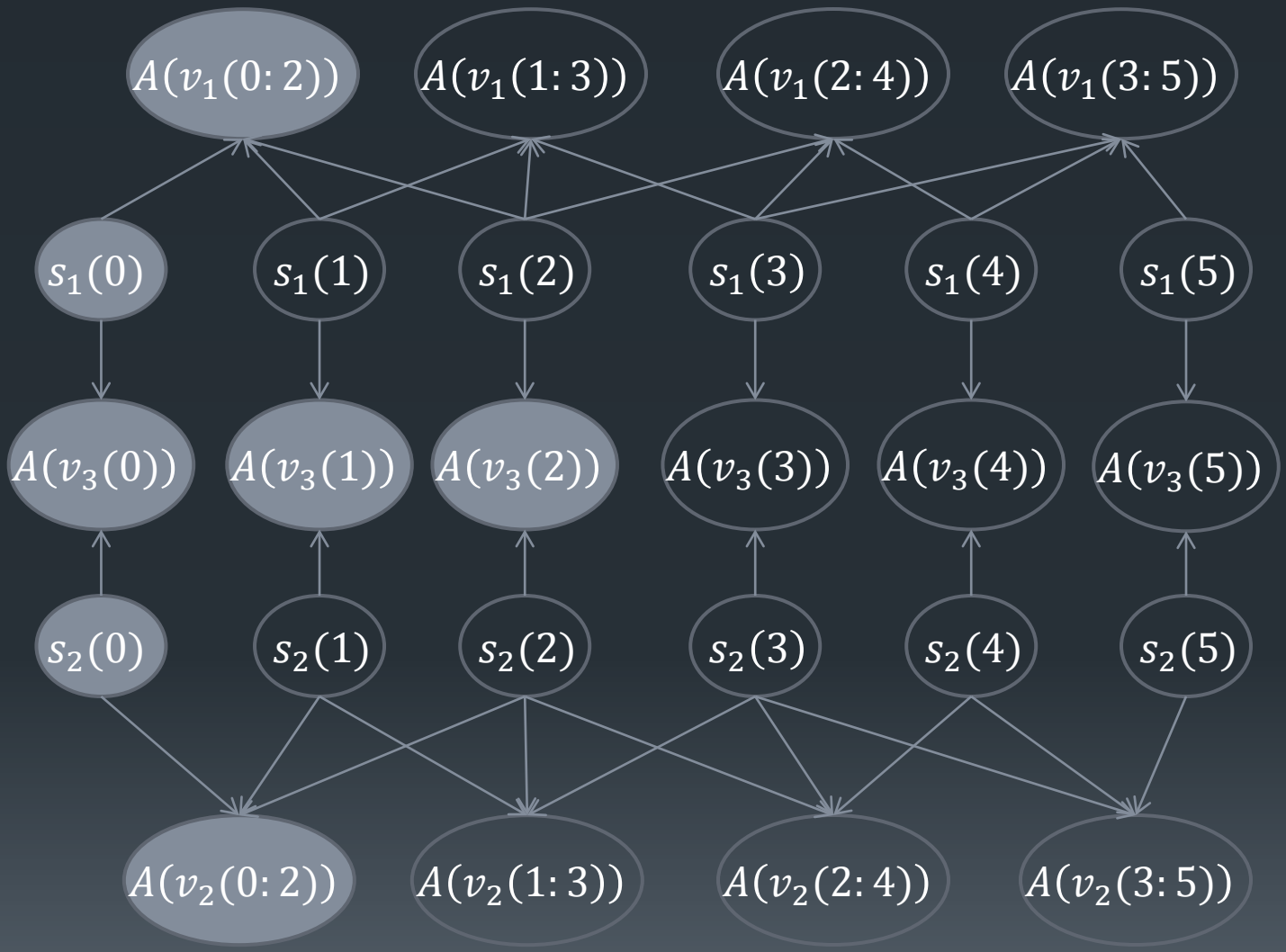
- Assemble incoming data into view tuples
- Compute anomaly score for each view tuple
- Perform probabilistic inference to determine which sensor states best explain the observed anomaly scores:

$$\arg\max_{S} P(S|A(V))$$

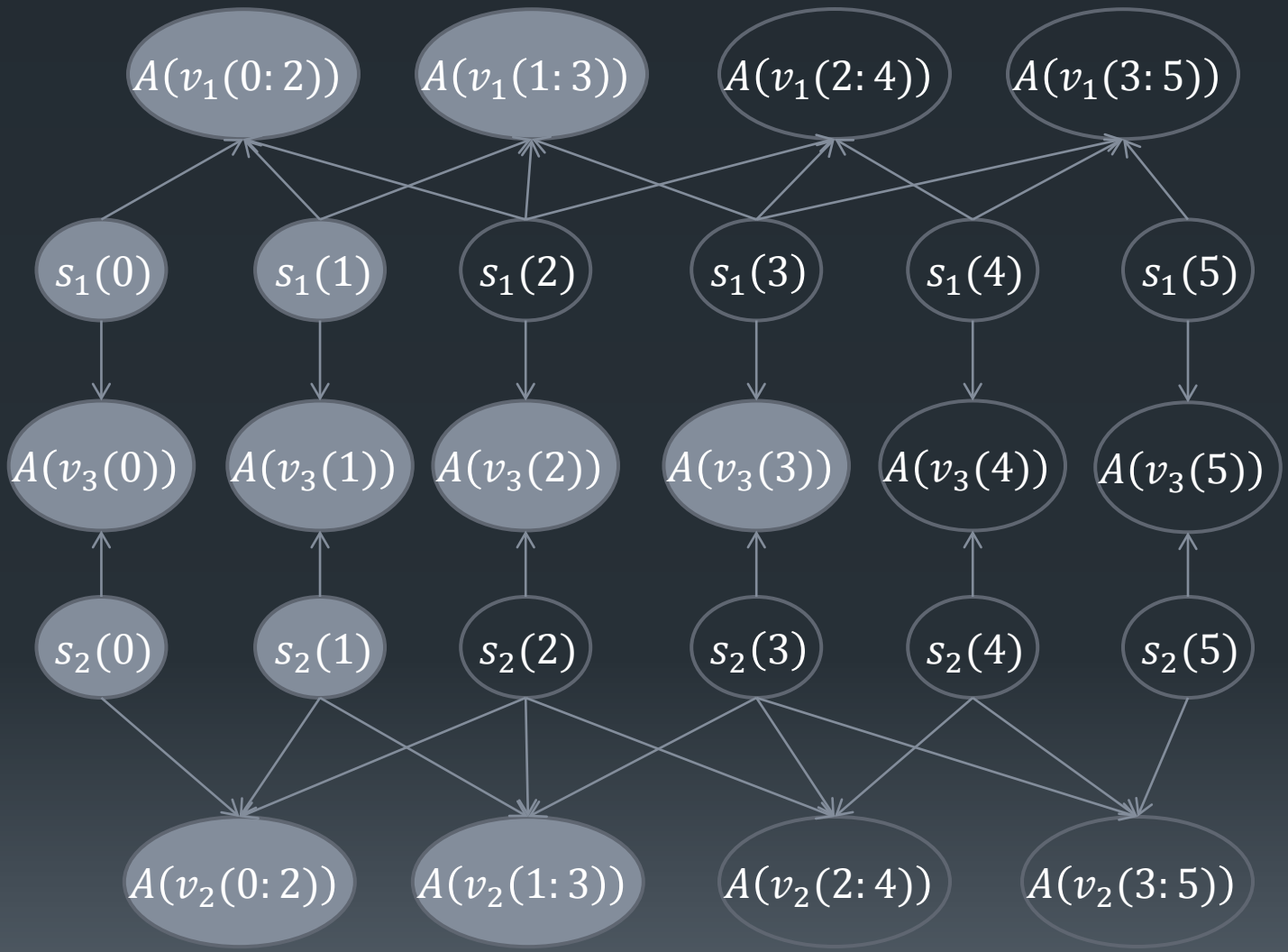
Online Probabilistic Inference

- We can’t wait for a whole year of observations before detecting broken sensors
- We have developed an incremental probabilistic inference approach

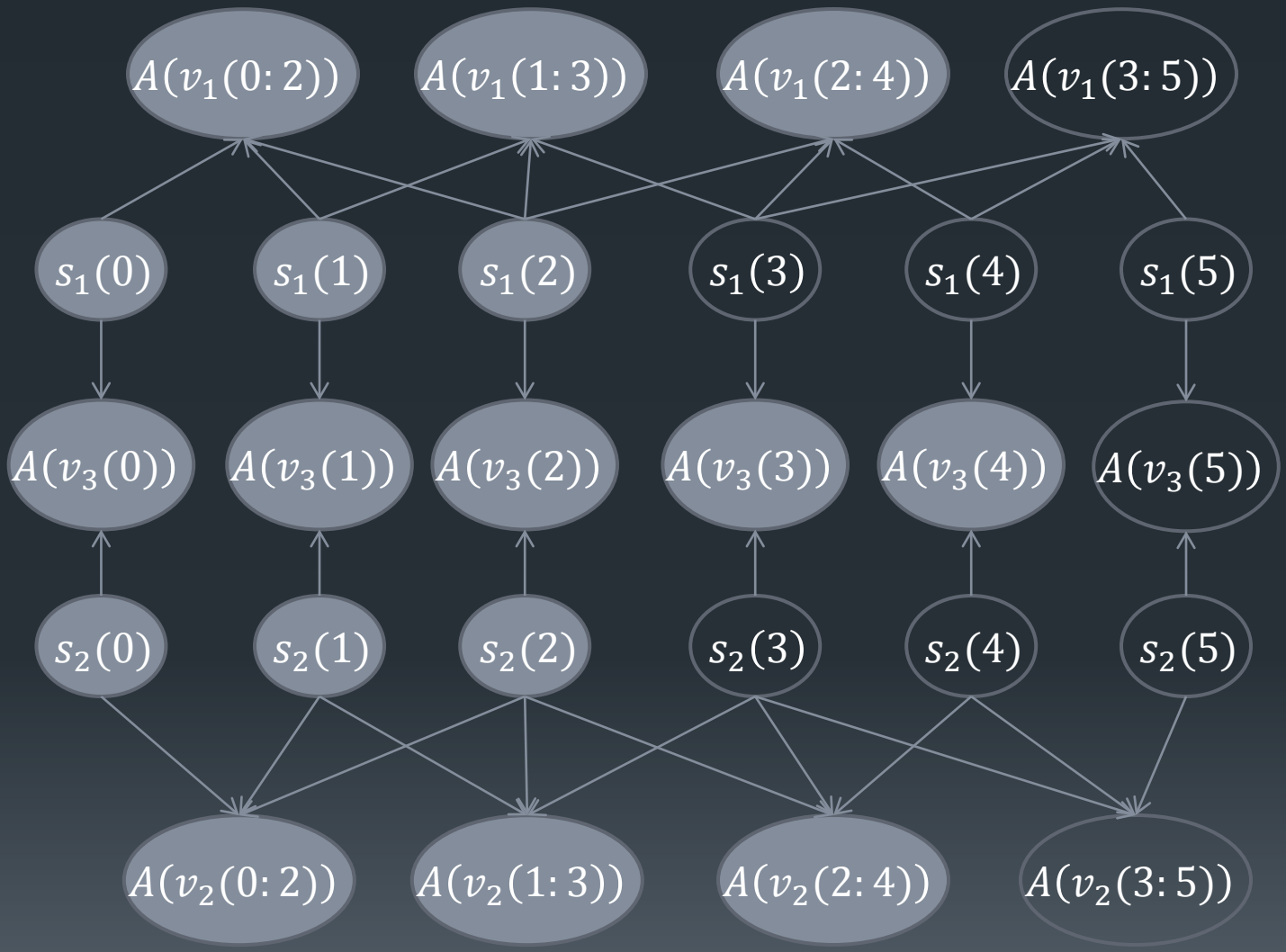
Data time: 2 Diagnosis time: 0



Data time: 3 Diagnosis time: 1



Data time: 4 Diagnosis time: 2



Computing Anomaly Scores

Many different possibilities

1. Joint Model $P(TEMP, RELH)$
   - Challenge: Joint relationship depends on day of year, amount of water in atmosphere
2. Time Series Model $P(TEMP_t | TEMP_{t-1}, TEMP_{t-2}, \ldots)$
   - Challenge: Seasonal variation, Daily variation, Weather system variation
3. Regression from Nearby Station
   - Because nearby weather station experiences the same dependencies on atmospheric water content, season, day, and weather system, it compensates for all of these sources of variation

Regression-Based Density Estimation

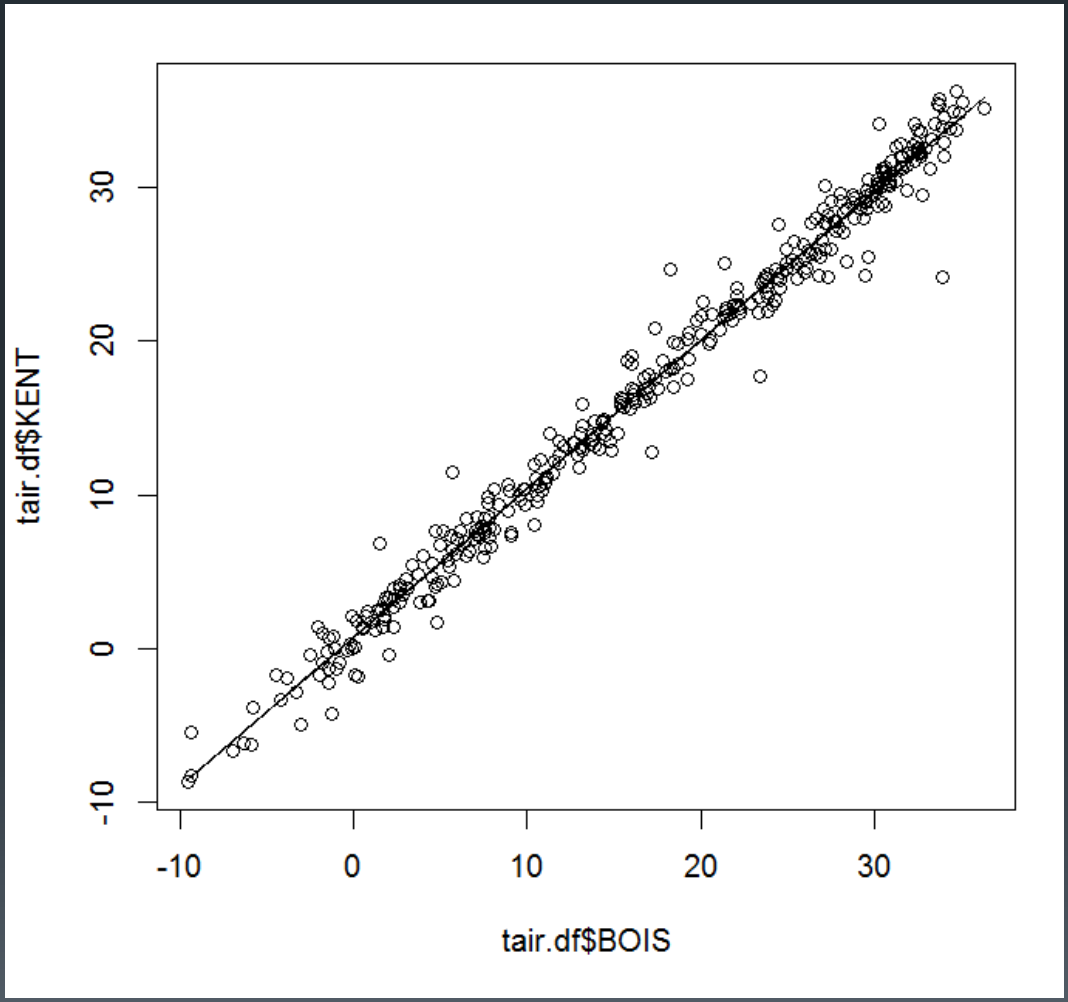
- Consider the view $\langle Temp(\ell, t), Temp(\ell', t)\rangle$ for two nearby weather stations ℓ and ℓ'
- We can fit a regression model

$$Temp(\ell, t) \approx \beta_0 + \beta_1 Temp(\ell', t)$$

- Ordinary least squares regression assumes that the response variable $Temp(\ell, t)$ has a normal distribution with
  - mean $\hat{\mu}_t = \beta_0 + \beta_1 Temp(\ell', t)$
  - variance $\hat{\sigma}_t^2 = \mathbb{E}[(Temp(\ell, t) - \hat{\mu}_t)^2]$
- We can compute the anomaly score as
  - $-\log Normal(Temp(\ell, t); \hat{\mu}_t, \hat{\sigma}_t^2)$

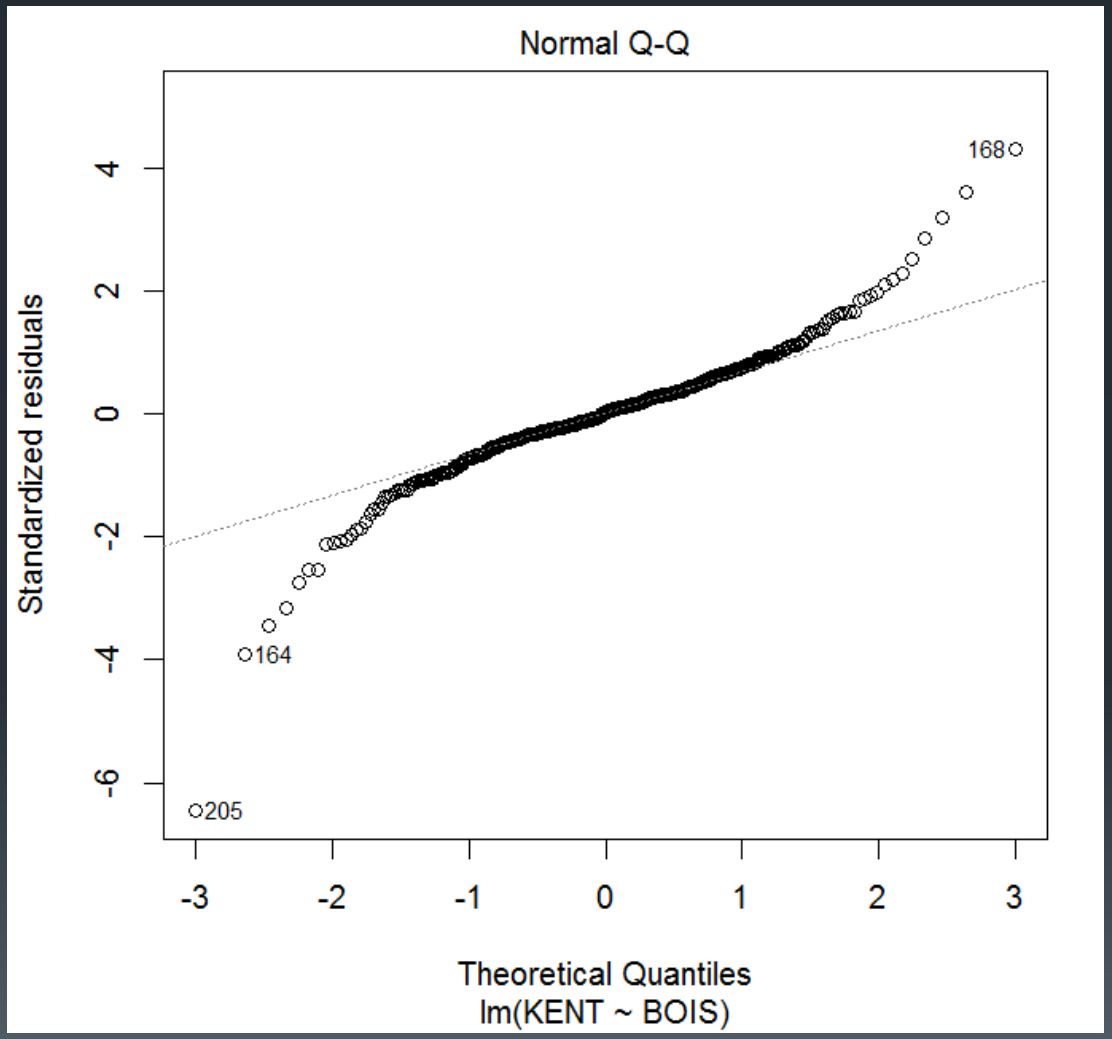
Linear Regression for Temperature

- Predicting temperature of KENT from temperature at BOIS (in Oklahoma, US)
- Temperature at 0:00UTC each day of 2009
- $KENT = 0.7344 + 0.9677\ BOIS$



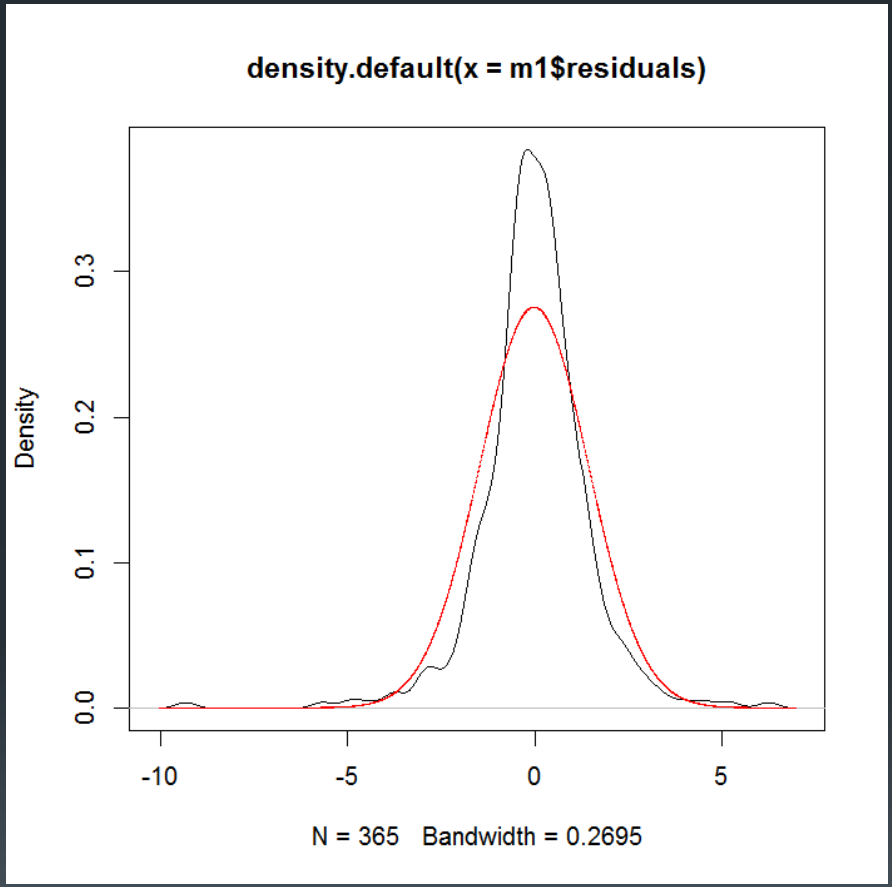
How Good is the Normal Distribution Assumption?

- This is a Q-Q plot
- X axis is the quantile of each residual according to the fitted Normal distribution
- Y axis is empirical quantile of each residual
- A perfect fit would have all points in the dotted line
- The residuals have heavier tails than the Gaussian

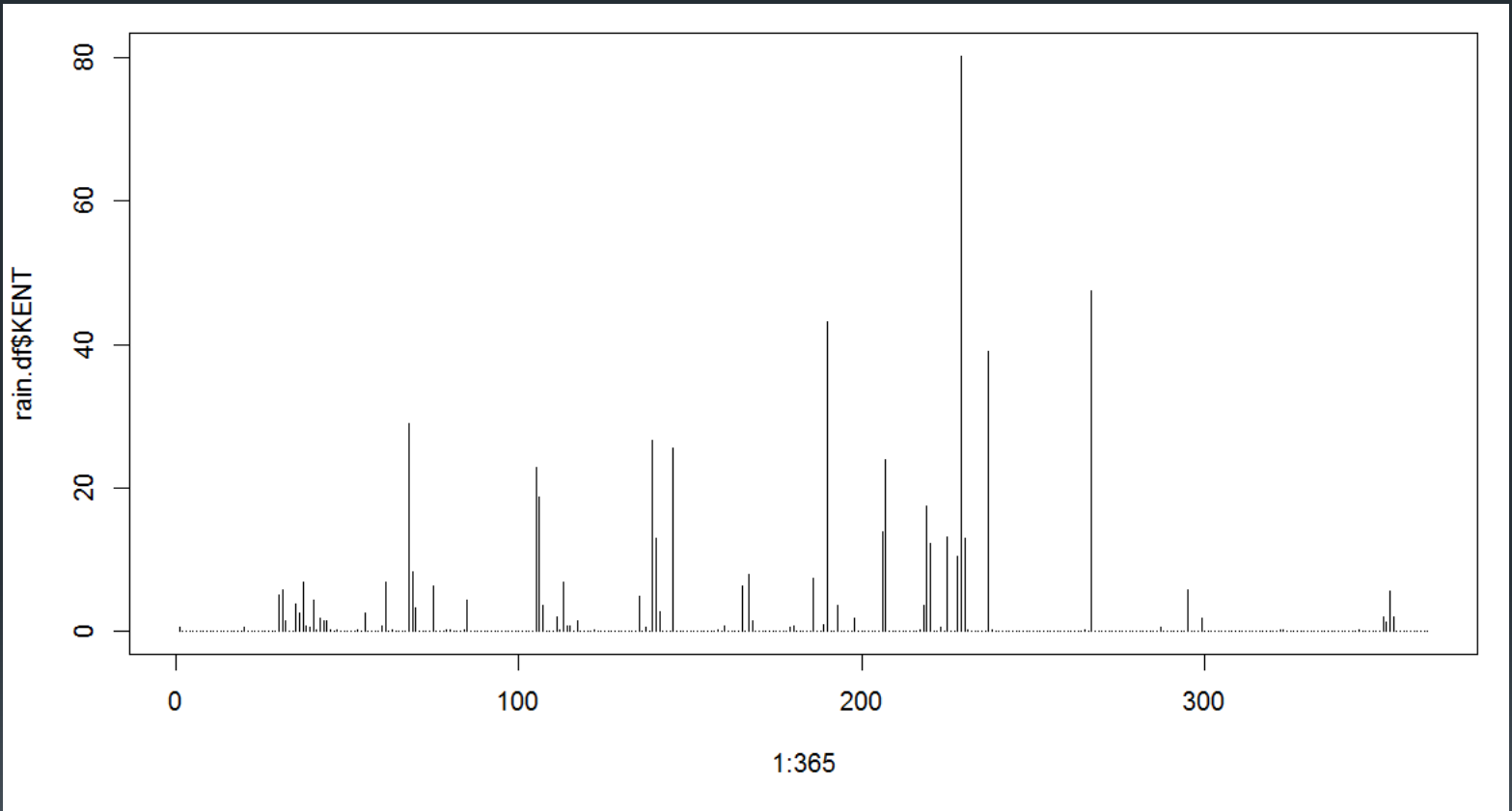


Kernel Density Estimate of the Residuals

- scikitlearn.neighbors.kde KernelDensity
- R: “density” automatically selects σ^2
- Replaces the assumption of a Gaussian distribution



Precipitation is very hard



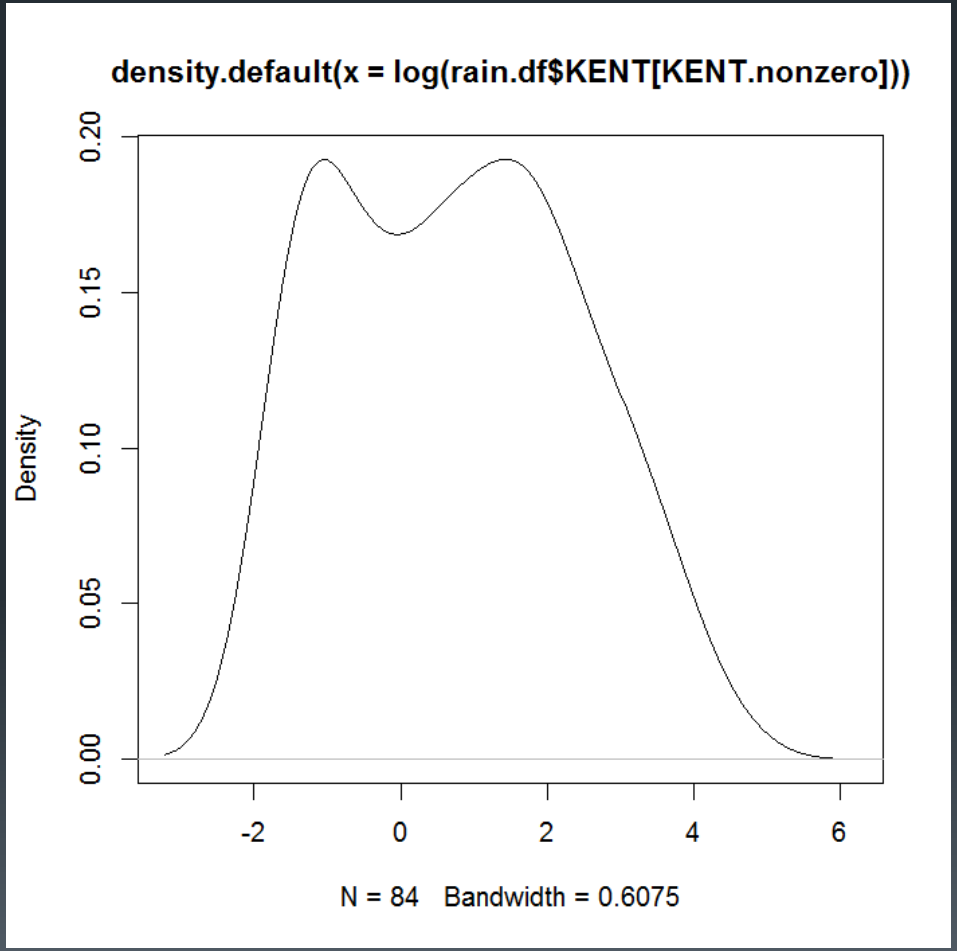
Precipitation

- Often exactly zero
- Very “bursty”; highly non-Gaussian

- We model the distribution as a mixture of two components
  - with probability p_0 we predict $RAIN = 0$
  - with probability $1 - p_0$ we draw an amount of rain according to $P(RAIN | RAIN > 0)$.
- The amount of rain should be positive, so we need to use a distribution over the positive real numbers
  - One solution is to predict $\log(RAIN)$

Kernel Density of log(RAIN)

$P(\log(RAIN)|RAIN > 0)$



Recall: Density Estimation under Transformations

- Let $g(\log x)$ be our density estimator for $\log x$
- To convert this to a density estimator $f(x)$ for x, we must divide by x:

$$f(x) = \frac{g(\ln x)}{x}$$

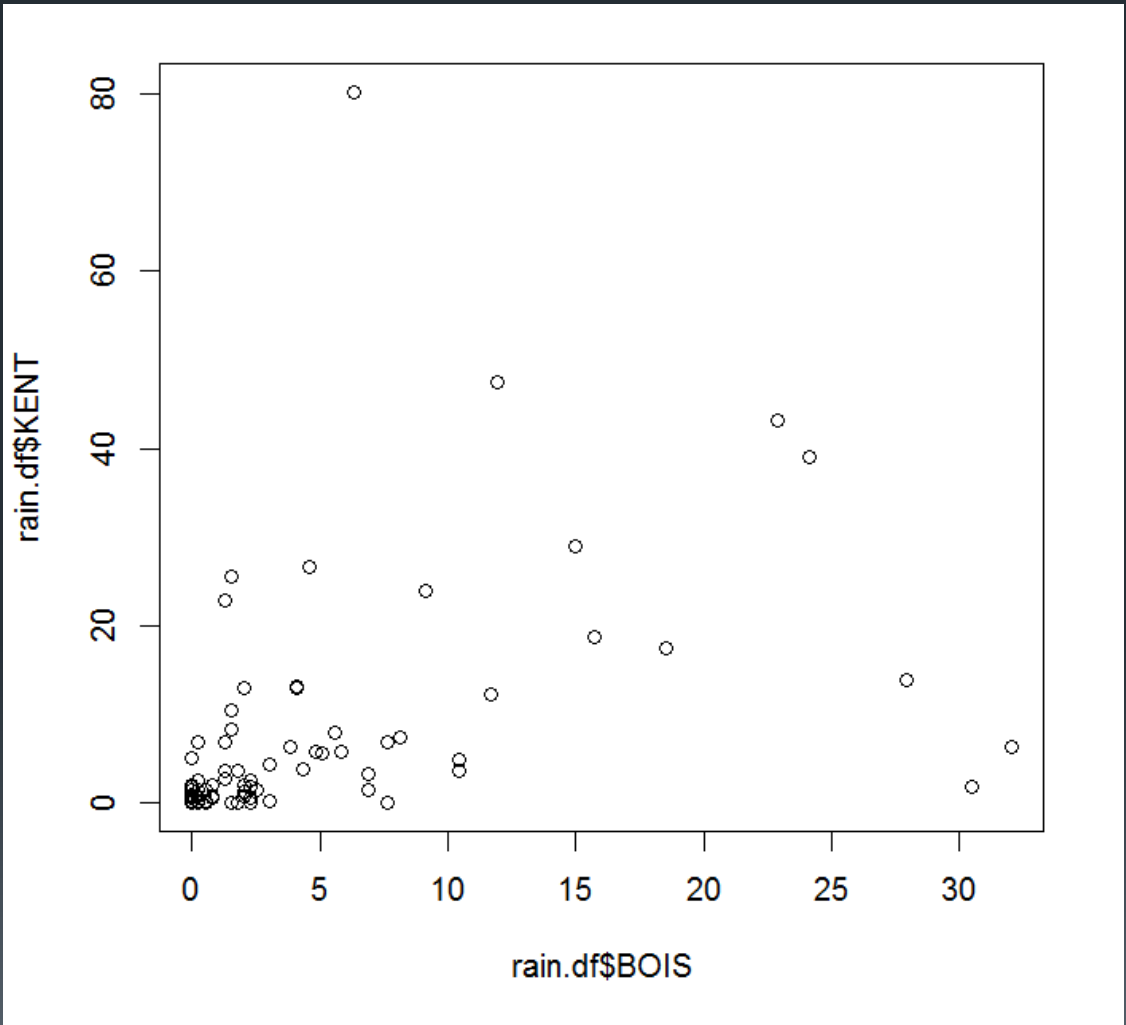
- The general rule is the following
- $f(x) = g\big(y(x)\big)\left|\frac{dy}{dx}\right|$
- In our case $y(x) = \ln x$ and $\frac{dy}{dx} = \frac{1}{x}$

Computing an Anomaly Score for RAIN

- If $RAIN = 0, -\log p_0$
- If $RAIN > 0, -\log\left[(1 - \mathrm{p}_0)\frac{\mathrm{P}(\log RAIN)}{RAIN}\right]$

Predicting RAIN at one station ℓ from a neighboring station ℓ'

- Does not look promising



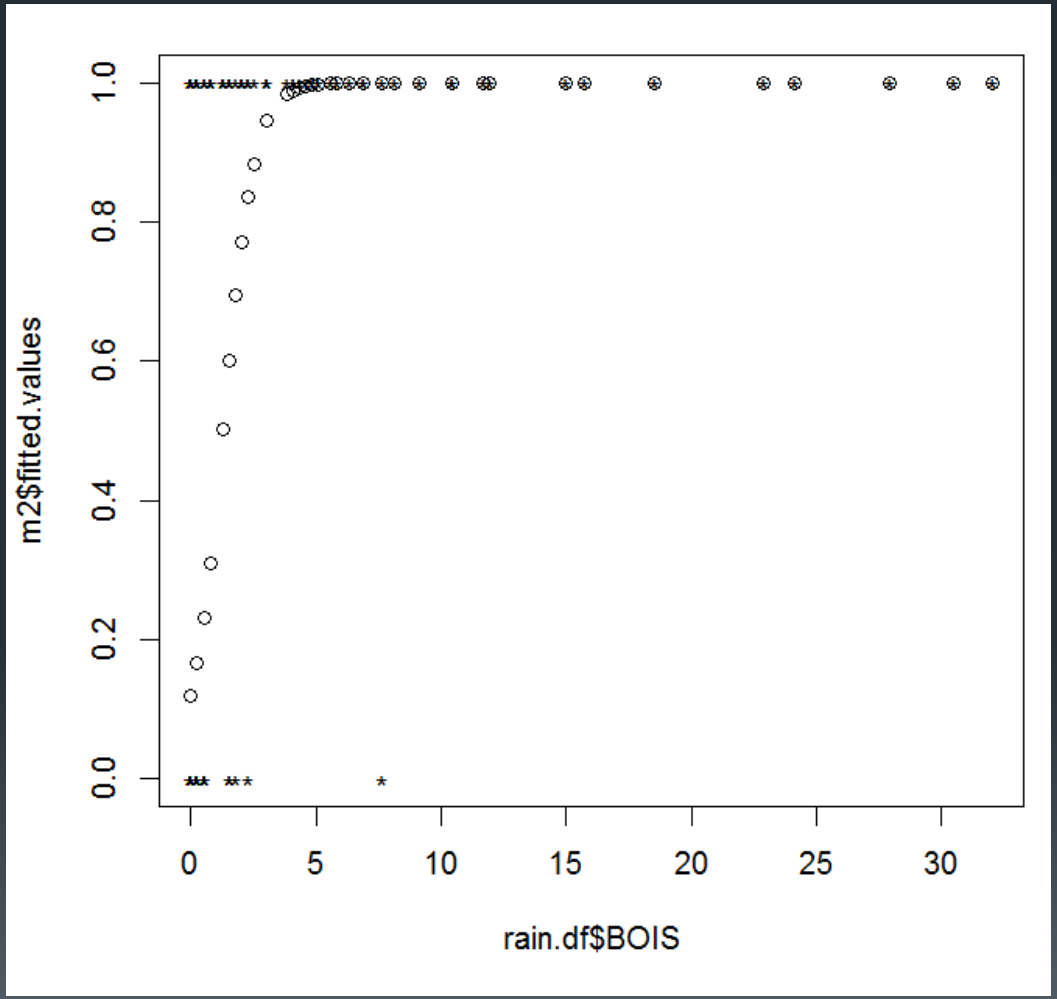
Conditional Mixture Model

- Condition each part of the Mixture Model on the neighboring station
- Linear Models:
  - Logistic regression model of $p_1 = 1 - p_0$:
    - $\log \frac{p_1}{1-p_1} = a + b\, Rain(\ell')$
  - Regression model for amount of rain
    - $\ln Rain(\ell) = c + d \ln Rain(\ell') + \epsilon$
    - Use KDE over the residuals ϵ

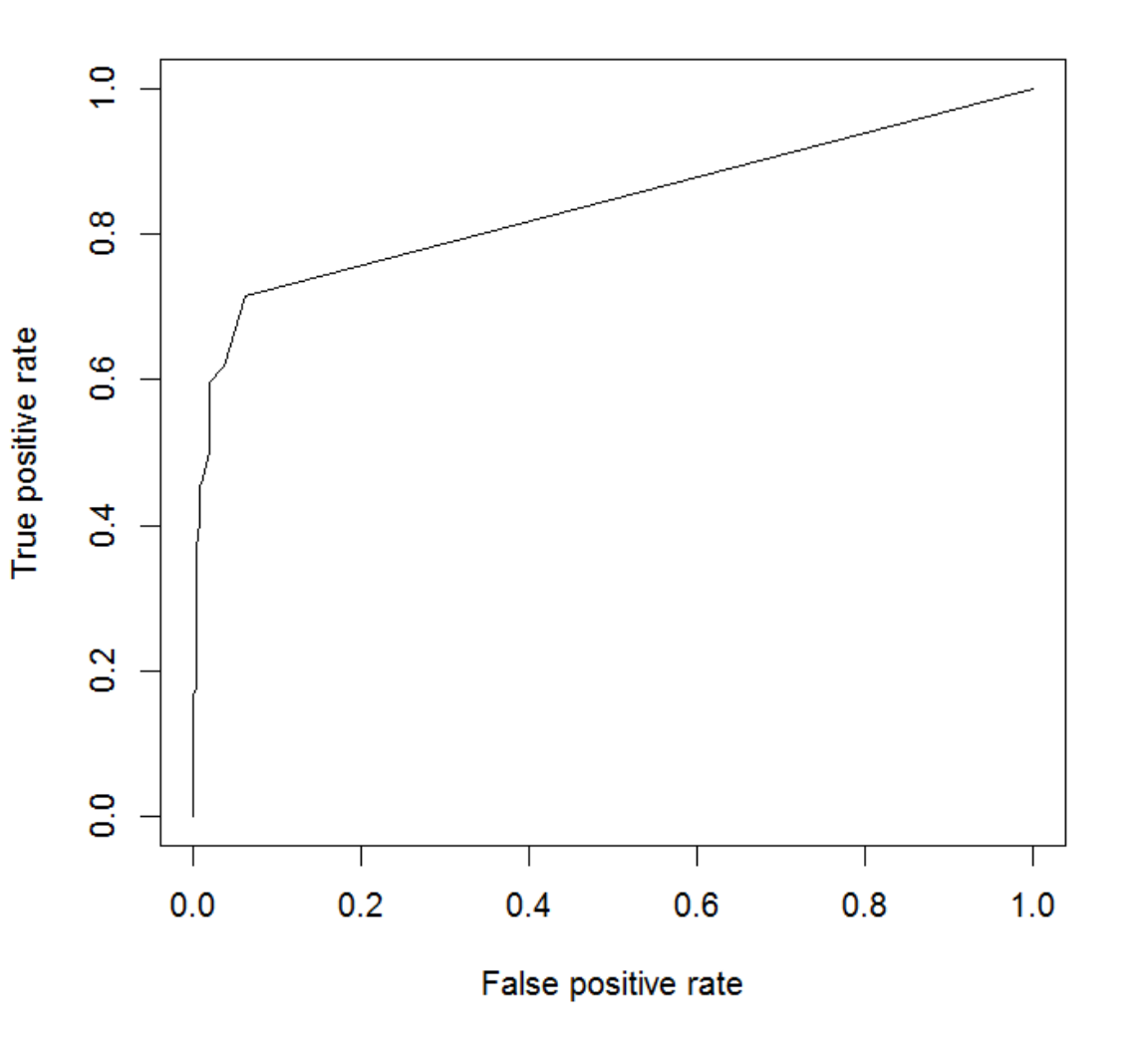
Predicting probability of rain

- Convert KENT rain to 0/1 variable “YES”
- Fit logistic regresion:

  logit(YES) = a + b BOIS



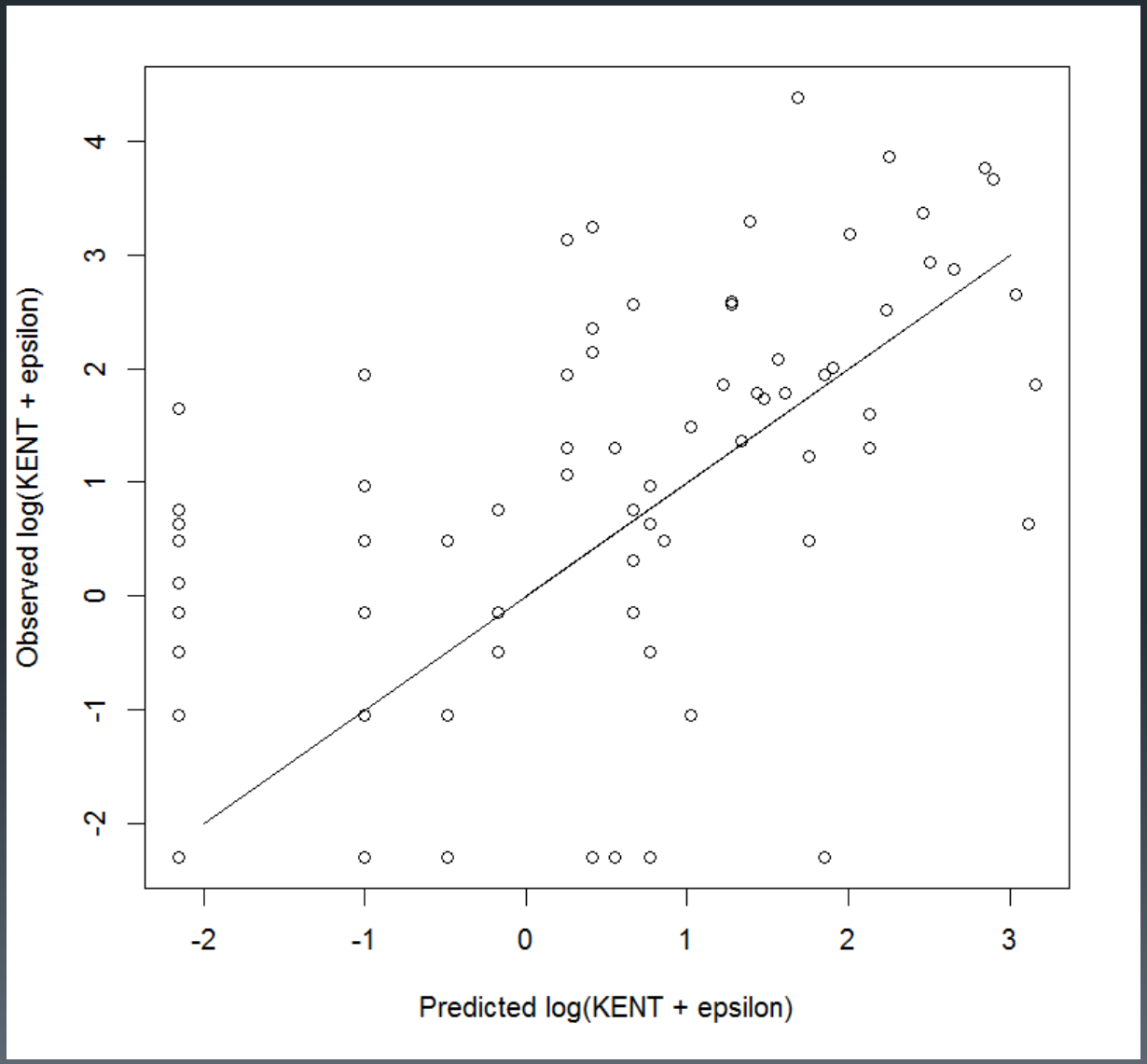
ROC Curve: AUC = 0.8397



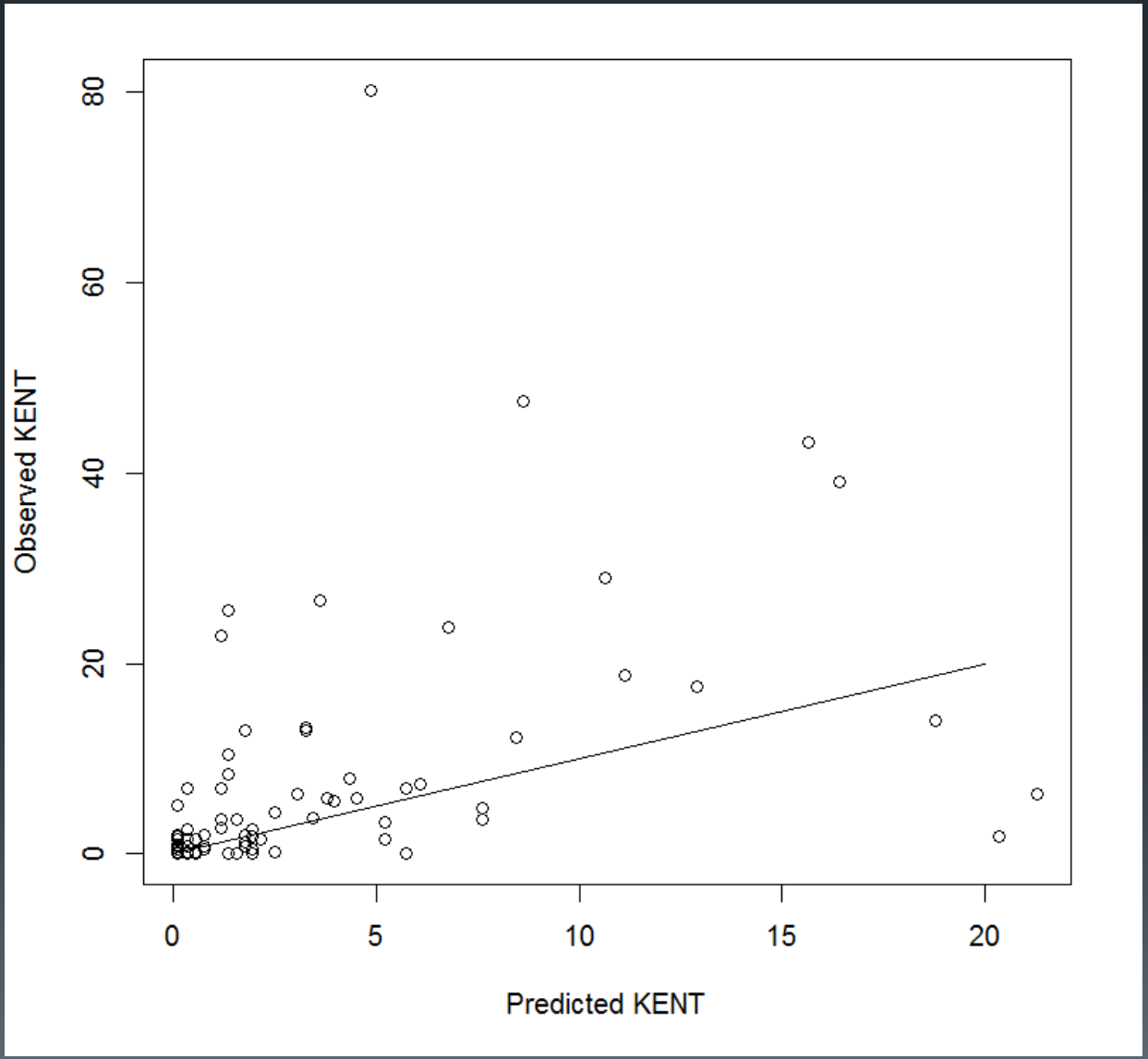
Challenge: Imperfect Rain Prediction

- The model for $p_1 = P\big(Rain(\ell) = 1 | P(rain(\ell'))\big)$ will not be perfect
- Therefore, we cannot train the model for $P\big(Rain(\ell)\big)$ using only non-zero values of $Rain$
- Solution: Add a small ϵ before taking the log
  - $\ln(Rain(\ell) + \epsilon) = c + d \ln(Rain(\ell') + \epsilon)$
  - For each training example $\big(Rain(\ell', t), Rain(\ell, t)\big)$, we employ a weight $w_t = P(Rain(\ell, t) = 1 | Rain(\ell', t))$
  - Find (c, d) to minimize the weighted squared error
  - $P\big(Rain(\ell)\big) = \frac{P\big(\ln\big(Rain(\ell')+\epsilon\big)\big)}{Rain(\ell)+\epsilon}$
  - I used $\epsilon = 0.1$

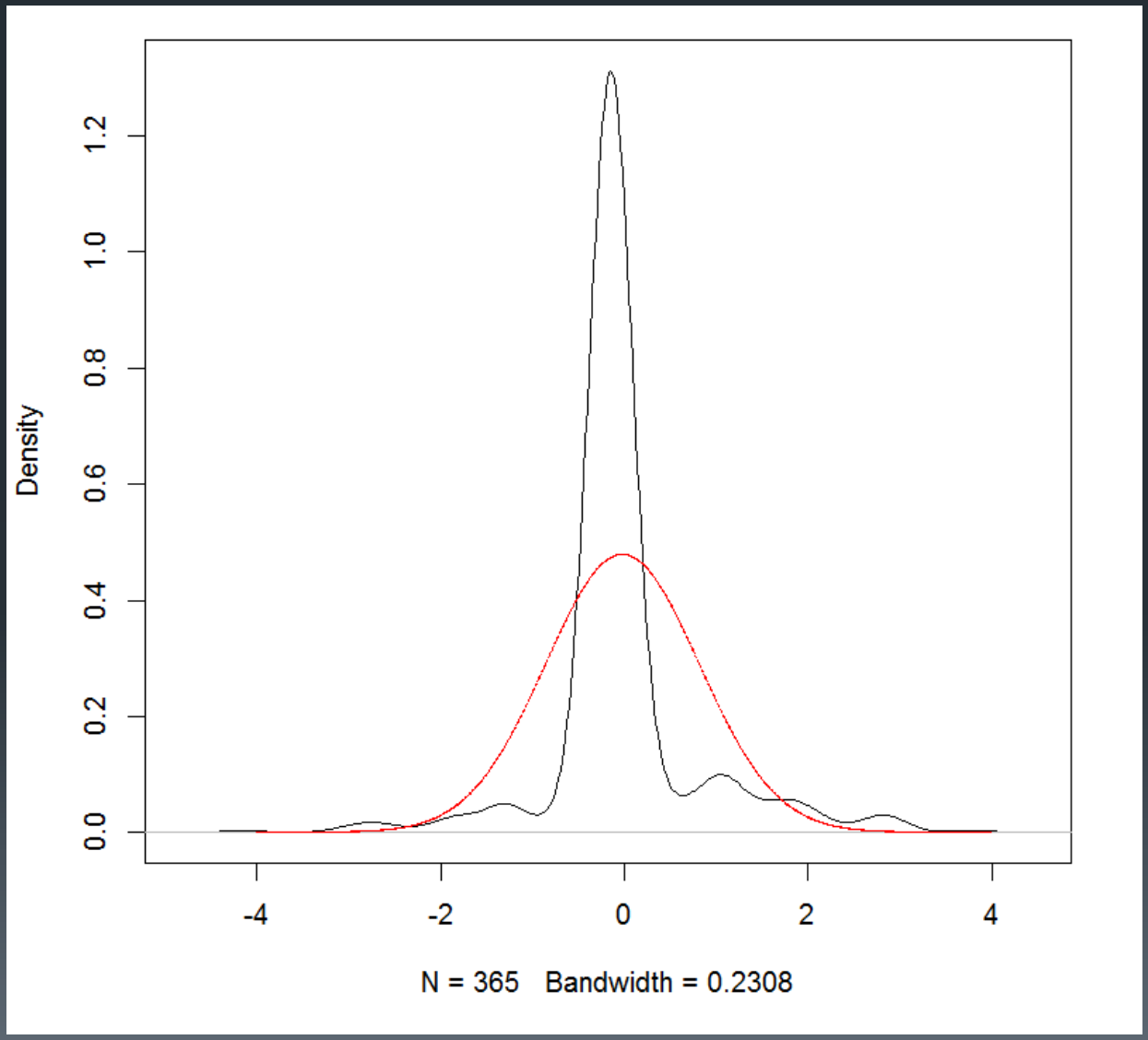
Quantitative Rain Prediction (log scale)



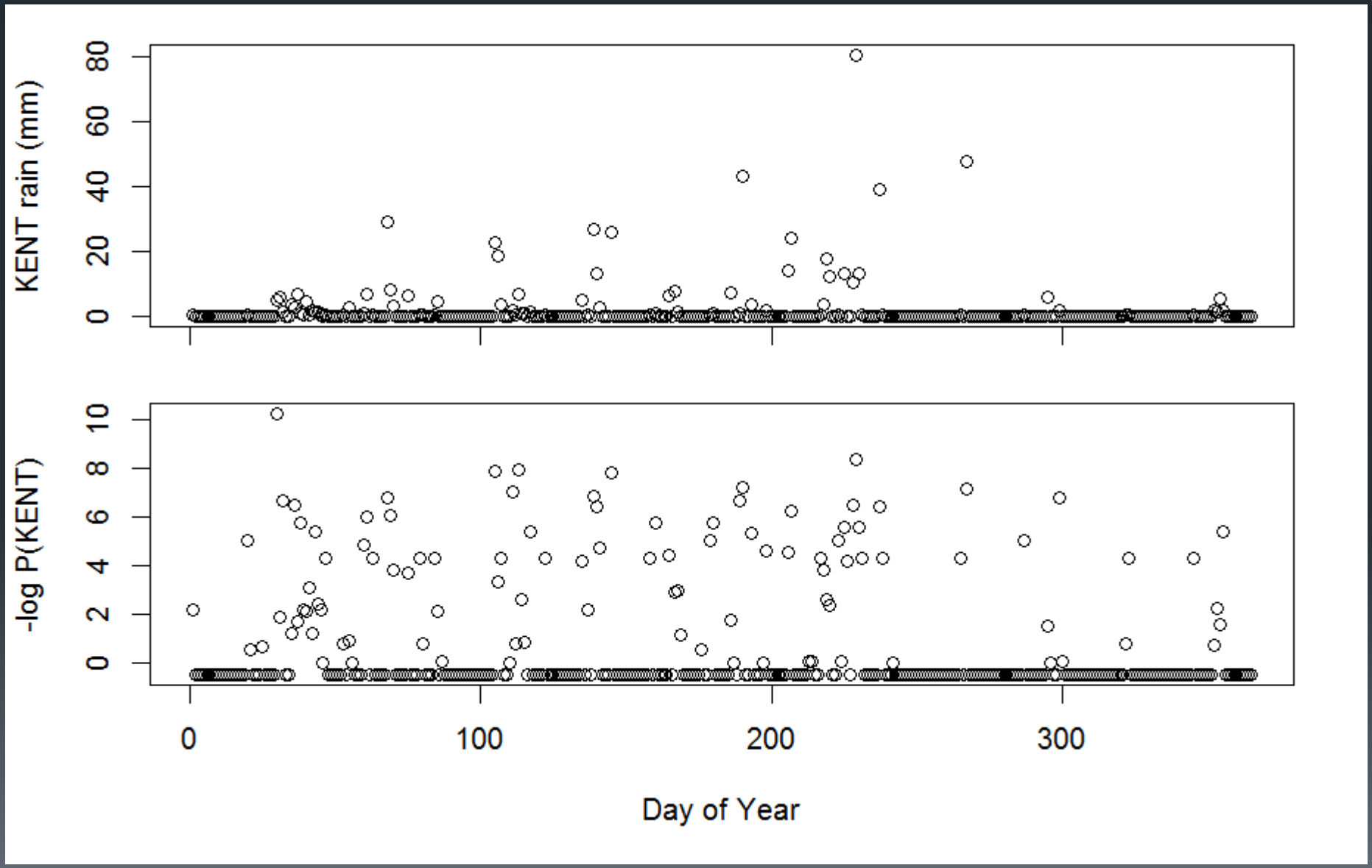
Quantitative Rain Prediction (in mm)



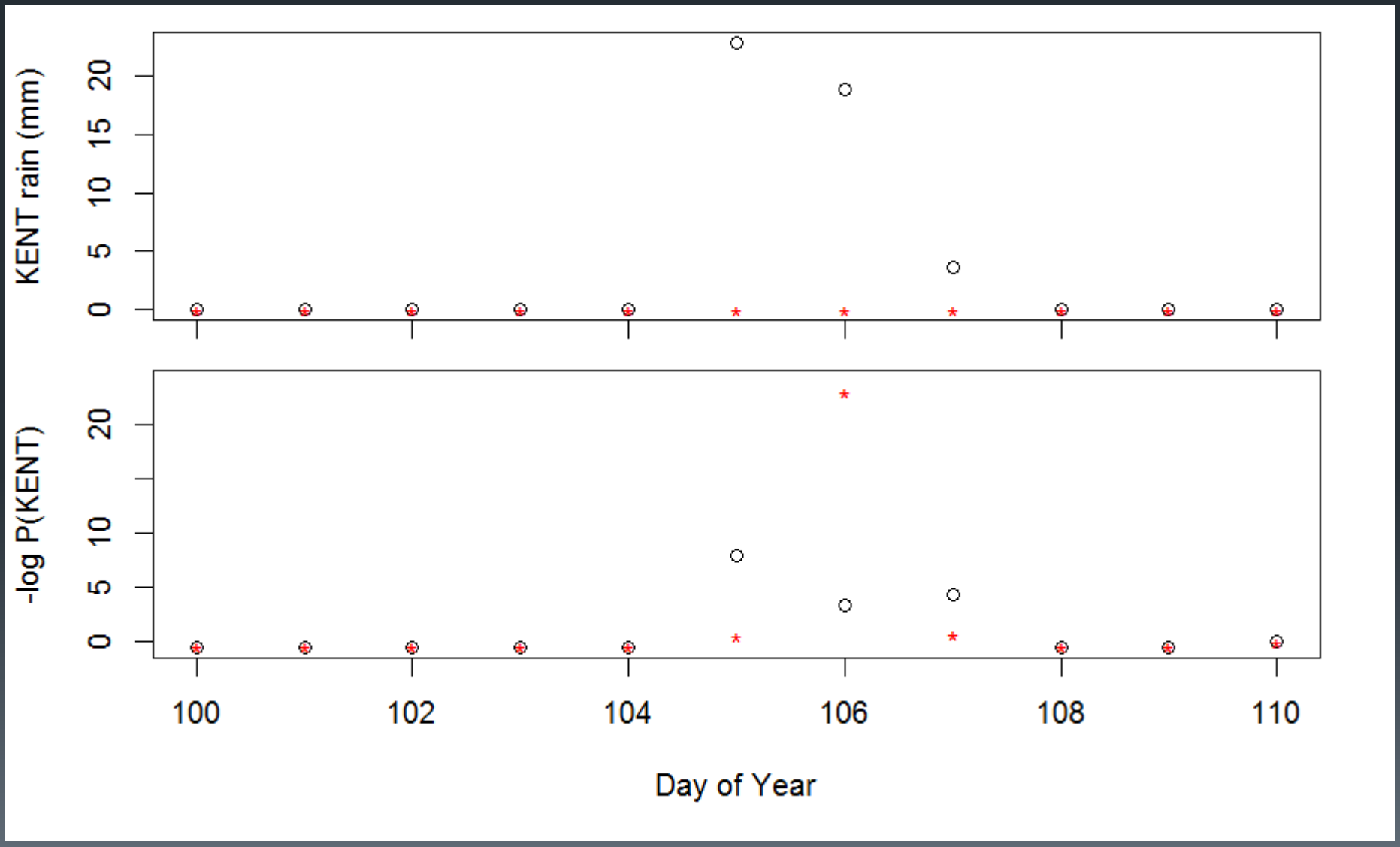
Residuals (log scale)



KENT anomaly score results



Fake failure for days 100-110



Summary

- TAHMO is creating a weather station network of unprecedented size
  - QC must be automated as much as possible
- Existing QC Methods
  - Rule-based (ad hoc)
  - Probabilistic (requires modeling the sensor values when the sensor is broken)
- SENSOR-DX Approach
  - Define multiple views
  - Fit an anomaly detector to each view
  - Probabilistic QC by modeling the anomaly scores of broken sensors
  - Diagnostic reasoning to infer which sensors are broken

Summary (2): Anomaly Detection Methods

- Predict sensor readings at station ℓ from a nearby station ℓ'
- For temperature, linear regression works well
  - But residuals are non-Gaussian, so we fit a kernel density estimator
- For Precipitation, we learn a mixture model
  - Logistic regression to predict $p_1 = P(RAIN(\ell) = 1)$
  - Weighted linear regression after transforming by $\log Rain(\ell) + \epsilon$
  - Again, residuals are non-Gaussian, so fit KDE

Exercise: https://github.com/tadeze/dsa2018

- Fit the Probabilistic Precipitation Model for Three Stations in Kenya
- Insert fake sensor failures
- Measure how well we can detect these sensor failures
  - Set a threshold: $-\log P\big(RAIN(\ell)\big) > \theta$
  - What value of θ can detect all of the fake failures but minimize false alarms?
    - Precision at 100% Recall
  - How bad are the false alarms?