Identification of Anomalous Data Entries in Repeated Surveys

Luca Sartore
National Institute of Statistical Sciences (NISS)
United States Department of Agriculture (USDA) National Agricultural Statistics Service (NASS)

Lu Chen (NISS/USDA NASS), Justin van Wart (USDA NASS)
Andrew Dau (USDA NASS), Valbona Bejleri (USDA NASS)

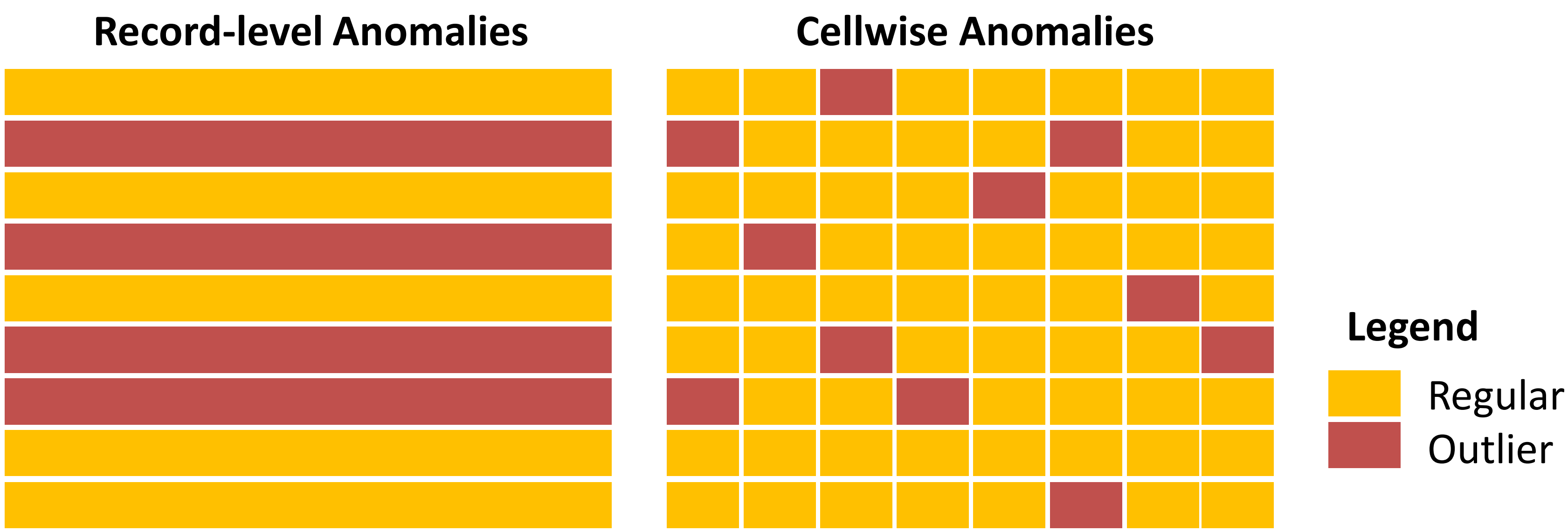
Government Advances in Statistical Programming 2023

NASS Editing System

- The USDA’s NASS conducts hundreds of surveys every year and prepares reports covering virtually every aspect of U.S. agriculture
- Data are acquired through repeated surveys
- Reviewing and vetting processes are time consuming
- **A semi-automated system** for editing
- The automation is **based on traditional** anomaly detection algorithms (**univariate outliers, edit limit thresholds**)
- There is a need for improved algorithms

A More Informative Approach

- Identifying anomalous data entries (**cells**) of a record (**row**)



Existing Approaches for Cellwise Outliers

- An R package (Raymaekers et al., 2022) that implements Agostinelli et al. (2015) and Rousseeuw et al. (2018)
- Not suitable for
  - Sparse datasets
  - Usage of previously reported data
  - Non-gaussian distributions
  - Stratified samples

Types of cellwise anomalies

- Four types of cellwise anomalies are identified cellwise anomalies
    1. Format anomalies
    2. Historical anomalies
    3. Tail anomalies
    4. Relational anomalies
- Score values are statistics obtained through transformations of the original data

A Distribution-free Approach

- Anomaly scores are based on Chebyshev-like inequalities proposed by Chepulis and Shevlyakov (2020)

$$\Pr(|X - \mu_\delta| \geq |\varepsilon|) \leq \min\left\{1, \frac{\sigma^\delta}{|\varepsilon|^\delta}\right\}$$

where X is a random variable, ε denotes the residuals, and

	$\delta = 1$	$\delta = 2$
μ_δ	Median	Mean
σ^δ	Mean absolute error	Variance

Combining Scores via Fuzzy Logic

- The scores are computed using
$$S_t = \min\left\{1, \frac{\sigma^\delta}{|\varepsilon|^\delta}\right\}$$
for any $t = 1,2,3,4$ type of anomaly
- The product t-norm (or triangular norm; Gupta 1991) defined as
$$S^* = \prod_t S_t$$
is used to identify anomalous entries
- These are detected when S^* is lower than the $100\tau\%$ quantile, where τ is predetermined by the user

Implementation & Testing

- FUZZY HRT was tested also with other datasets of varying sizes and different complexities
  - 5 to 50 variables of interest
  - 218 to 21,154 records
  - varying from higher to lower incidence of anomalies
- To improve calculation time, algorithm was developed based on SIMD instructions and the OpenMP library in C
- C code was interfaced in R to run the algorithm, using the `.C()` function (R Core Team, 2023)

Code for Format Anomalies

```
/**
 * @brief Checking format inconsistencies in the data vector
 * @param zScore pointer to an empty vector (for the output)
 * @param x pointer to the vector of data
 * @param n pointer to the size of the data vector
 */
void format_check(double *zScore, double *x, int *n) {
    int i;
    #pragma omp parallel for simd
    for (i = 0; i < *n; i++) {
        zScore[i] = (double) (x[i] > 0.0);
    }
}
```

Code for Historical Anomalies

```
...
vr = 0.0; nn = 0;
#pragma omp parallel for simd reduction(+ : vr, nn)
for (i = 0; i < *n; i++) {
    hdta[i] = log(current_data[i] / previous_data[i]);
    hdta[i] = isfinite(hdta[i]) ? fabs(hdta[i]) : 0.0;
    vr += hdta[i]; nn += (int) (hdta[i] > 0.0);
}
vr /= (double) nn;
#pragma omp parallel for simd private(tmp)
for (i = 0; i < *n; i++) {
    tmp = vr / hdta[i];
    hScore[i] = tmp < 1.0 ? tmp : 1.0;
} ...
```

Code for Tail Anomalies

```
/**
 * @brief Checking distribution tails of each columns by group
 * @param dta Pointer to the input dataset
 * @param dim Pointer to the size of the input dataset
 * @param gr  Pointer to the integer vector with group IDs
 * @param ng  Pointer to total number of groups
 * @param tScore Pointer to the scoring vector for tail outliers
 */
void tail_check(double *dta, int *dim, int *gr, int *ng, double *tScore) {
    int i, g;
    #pragma omp parallel for default(shared) private(i, g) collapse(2)
    for (i = 0; i < dim[1]; i++) {
        for (g = 1; g <= *ng; g++) {
            group_tail(&tScore[*dim * i], &dta[*dim * i], dim, gr, g);
        }
    } ...
```

Further Details on Tail Anomalies

```
...
m = median(preprocess_data, n_valid_samples);
for (i = 0; i < n_valid_samples; i++) {
    preprocess_data [i] -= m;
    preprocess_data [i] = fabs(preprocess_data[i]);
}
v = median(preprocess_data, n_valid_samples);
v = (double) (v <= 0.0) + (double) (v > 0.0) * v;
v = 1.0 / v;
for (i = 0; i < nn; i++) {
    res[i] += (double) (gr[i] == g) * (data[i] - m) * v;
}
...
```

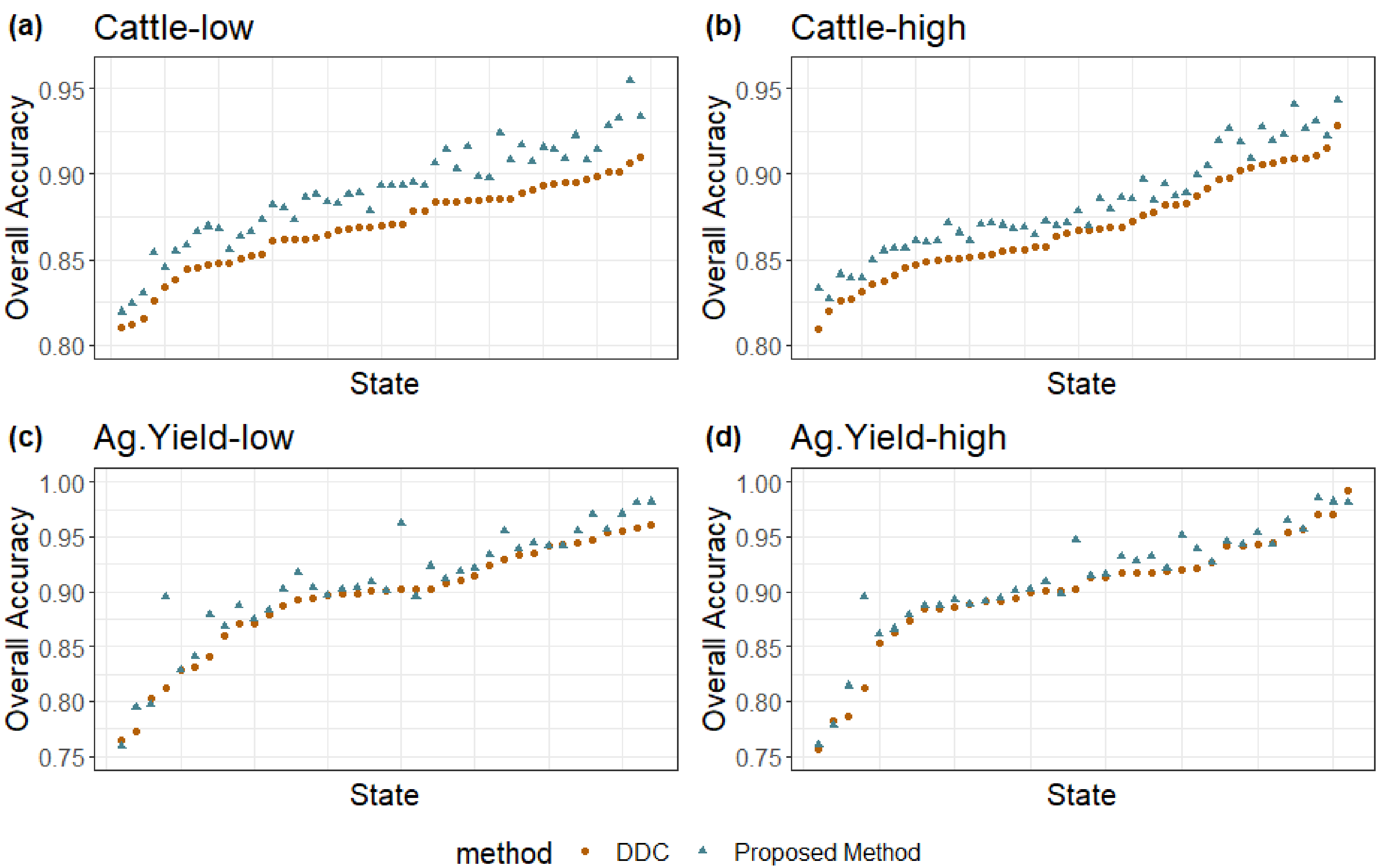
Code for Relational Anomalies

```
...
    * @param A   Pointer to input data matrix
    * @param dim Pointer to size of input matrix
...
void relat_check(double *A, int *dim) {
    int i; double tmp, v, *Q, *E, *qty; ...
    E = (double *) malloc(dim[0] * dim[1] * sizeof(double));
    ...
    #pragma omp parallel for default(shared) private(i)
    for (i = 0; i < dim[1]; i++) col_check(E, A, dim, i);
    #pragma omp parallel for simd
    for (i = 0; i < dim[0] * dim[1]; i++) A[i] = E[i];
    ...
```

Further Details on Relational Anomalies

```
... /** @param s skipping index */ ...
void col_check(double *E, double *A, int *dim, int s) {
    int i, j, k; double tmp, v, *Q, *qty;
    Q = (double *) malloc(dim[0] * (dim[1] - 1) * \
                              sizeof(double));
    qty = (double *) malloc((dim[1] - 1) * sizeof(double));
    if (Q && qty) {
    /* Compute only the matrix Q of the QR-decomposition */...
    /* Computing Q^t y */...
    /* Computing the residuals (i.e., y - Q(Q^t y)) */...
    }
    free(Q); free(qty);
}
```

Application and Results



Luca Sartore (NISS / USDA NASS)

Conclusions

- Modern editing systems must identify cellwise outliers not just anomalous records
- Our algorithm has been more effective than existing approaches when detecting cellwise outliers
- High-performance computing in R can be achieved via using SIMD instructions and the OpenMP library in C
- More research is needed for generalizing our approach to discrete distributions

References

1. Agostinelli C., Leung A., Yohai V.J., and Zamar R.H. Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. Test. 2015 Sep;24(3):441-61
2. Rousseeuw P.J., and Van Den Bossche W. Detecting deviating data cells. Technometrics 60, no. 2 (2018): 135-145
3. Raymaekers J., Rousseeuw P., Van den Bossche W., and Hubert M. cellWise: Analyzing data with cellwise outliers. CRAN, R package version 2.5.0 (2022): 467
4. Chepulis MA, Shevlyakov GL. On outlier detection with the Chebyshev type inequalities. Журнал Белорусского государственного университета. Математика. Информатика. 2020;3(0):28-35
5. Gupta MM, Qi J. Theory of T-norms and fuzzy inference methods. Fuzzy sets and systems. 1991 Apr 15;40(3):431-50
6. R Core Team. Writing R Extensions. R Foundation for Statistical Computing, Vienna, Austria. 2023.

Thank you!

Questions?

Luca Sartore, PhD luca.sartore@usda.gov
Lu Chen, PhD lu.chen@usda.gov