



Team 7

**Project Title: Local Governments Survey (Harmonizing ARISE City Survey +
CivicPulse Kansas Sample)**

Course Number:CS896

Team Names: Anudeepthi Chirumamilla- Y828R524

Rupali Avhad - d798z925

Teja Sree Mukkapati-Q389W584

1.Introduction

Objective: This project aims to enable comparable measurement of workforce challenges, fiscal condition, and capacity/service preparedness indicators across local governments by harmonizing overlapping survey constructs across two instruments: Civic Pulse (Kansas sample from a national survey) and ARISE (Kansas city survey).

Why harmonize? Although the ideas measured by both surveys are similar, their question formats, coding systems, and item coverage differ. In order to define harmonized variables for analysis and align constructs, a crosswalk is necessary.

2. Data Source

The dataset used in this analysis is:

Rachel_Krause_ARISEData_Combined General Dataset.csv

This dataset contains survey responses from local governments and includes:

- 1.Workforce challenge (Q8)
2. Fiscal condition (Q119)
3. Grant capacity (Q5)
4. Preparedness program indicators (Q9 series)
- 5.Vulnerable population support indicators (Q10 series)

Each row represents a respondent, and each column represents a survey item.

3.Final Crosswalk Table(deliverable)

In order to ensure conceptual consistency and facilitate cross-source comparison, the Final Crosswalk Table was created to standardize important variables across the ARISE and CivicPulse datasets. In order to improve the dependability of cross-dataset interpretations, the harmonization procedure concentrated on bringing variable definitions, response scales, and measurement goals into alignment.

Four primary concepts were harmonized:

Concept	Harmonization Summary	Purpose / Interpretation
1.Workforce Challenge	1–5 scale retained (higher = more challenging) across ARISE Q8 and CivicPulse workforce challenge question.	measures staffing capacity and service strain.
2.Fiscal Condition:	1–5 scale (Very Poor–Excellent) harmonized between ARISE Q119 and CivicPulse fiscal condition rating.	Indicator of local government financial health.
3.Emergency Preparedness Capacity	Converted preparedness items to binary indicators (1 = present, 0 = not present); summed for total capacity score.	Measures disaster readiness and response capability.
4.Protection of Vulnerable Populations	Aligned actions for elderly, low-income, and disabled groups with CivicPulse vulnerability items to form support score.	Measures social resilience and equity.

4. Data Cleaning Methodology (Implemented in Code)

Missing Value Handling

The ARISE dataset contained missing values represented in multiple inconsistent formats.

The cleaning pipeline standardized missing data by converting:

- Blank cells and whitespace → NaN
- Textual missing indicators (e.g., NA, N/A, Don't know, Refused) → NaN
- Numeric coded missing values (e.g., 99, 999, -99) → NaN
- This step ensures that all missing values are uniformly treated during analysis.

Additionally, a missingness report was generated to identify:

- Columns with highest missing percentage
- Rows with high levels of non-response

```
j: print("Loaded:", arise.shape)
   arise.head()

Loaded: (309, 91)

j: 
```

	Unnamed: 0	ResponseId	RecipientLastName	ExternalReference	1st Dis	2nd Dis	3rd Dis	Q3_10_Agri	Q3_9_Cyber	Q3_19_Dam	...	Q13_Reduce staff	Q13_Defer capital projects
0	1	R_2sQnn7lItA9jdBn	chief administrator	2079950	3.0	21.0	1.0	NaN	NaN	NaN	...	1	1
1	2	R_2Qyp1YFLCR17M9C	chief administrator	2061250	11.0	3.0	8.0	NaN	NaN	NaN	...	1	1
2	3	R_2aQAihYkimvMFey	chief administrator	2005600	11.0	21.0	1.0	NaN	NaN	NaN	...	1	1
3	4	R_6sTVkGKJ3zy6aPL	chief administrator	2034300	21.0	1.0	2.0	NaN	NaN	NaN	...	0	1
4	5	R_3sddVtnZ4CSdNk0	chief administrator	2053225	3.0	1.0	23.0	NaN	NaN	NaN	...	1	1

```
5 rows x 91 columns

j: if "Unnamed: 0" in arise.columns:
    arise.drop(columns=["Unnamed: 0"], inplace=True)
    print("Removed extra column: Unnamed: 0")

Removed extra column: Unnamed: 0

j: # 1) Blank spaces -> NaN
   arise = arise.replace(r"^\s+$", np.nan, regex=True)

   # 2) Common missing text -> NaN
   arise = arise.replace(
       ["NA", "N/A", "na", "n/a", "NULL", "null", "None", "none", "Don't know", "Dont know", "DK", "Refused", "Prefer not to say"],
       np.nan
   )

   # 3) Common numeric missing codes -> NaN
   arise = arise.replace({99: np.nan, 999: np.nan, 9999: np.nan, -99: np.nan, -999: np.nan})

   print("Missing values standardized (blank/text/codes).")

Missing values standardized (blank/text/codes).

: missing_percent = (arise.isna().mean() * 100).sort_values(ascending=False)
  missing_count = arise.isna().sum().sort_values(ascending=False)

  missing_report = pd.DataFrame({
      "missing_count": missing_count,
      "missing_percent": missing_percent.round(2)
  })

  print("Top 15 columns with highest missing %:")
  display(missing_report.head(15))

Top 15 columns with highest missing %:
```

	missing_count	missing_percent
Q3_17_Terrorism	309	100.00
Q3_12_Earthquake	309	100.00
Q3_19_Dam	306	99.03
Q3_7_waste spill	305	98.71
Q3_8_Industrial fire	305	98.71
Q3_10_Agri	302	97.73
Q3_23_Others	301	97.41
Q3_23_TEXT	296	95.79
Q3_22_Soil erosion	291	94.17
Q3_5_Tornados	289	93.53
Q3_9_Cyber	289	93.53
Q6_7_TEXT	285	92.23
Q3_6_Wildfires	276	89.32
Q3_3_Floods	245	79.29
Q3_2_heat	236	76.38

```


j: # These are the key ARISE variables from our crosswalk.
   for col in ["Q8", "Q119", "Q5"]:
       print(col, "found" if col in arise.columns else "NOT found")

Q8 found
Q119 found
Q5 found
```

This improves transparency and reproducibility of the cleaning process.

```
: # If a value is not between 1–5, we treat it as invalid and convert it to missing (NaN).
```

```
for col in ["Q8", "Q119", "Q5"]:  
    if col in arise.columns:  
        x = pd.to_numeric(arise[col], errors="coerce")  
        arise[col] = x.where((x >= 1) & (x <= 5), np.nan)
```

```
print("Fixed: Q8, Q119, Q5 now contain only valid 1–5 values (or NaN).")
```

Fixed: Q8, Q119, Q5 now contain only valid 1–5 values (or NaN).

```
: # If a value is not between 1–5, we treat it as invalid and convert it to missing (NaN).
```

```
for col in ["Q8", "Q119", "Q5"]:  
    if col in arise.columns:  
        x = pd.to_numeric(arise[col], errors="coerce")  
        arise[col] = x.where((x >= 1) & (x <= 5), np.nan)
```

```
print("Fixed: Q8, Q119, Q5 now contain only valid 1–5 values (or NaN).")
```

Fixed: Q8, Q119, Q5 now contain only valid 1–5 values (or NaN).

5. Harmonized Variable Construction

Core Harmonized Variables (Likert 1–5)

Based on the crosswalk and cleaned dataset, the following harmonized variables were created:

workforce_challenge_1to5 ← Derived from Q8

fiscal_condition_1to5 ← Derived from Q119

grant_capacity_1to5 ← Derived from Q5

survey_source = "ARISE"

These variables preserve the original 1–5 scale and maintain directional consistency (higher values indicate stronger magnitude of the construct).

```
# These are the cleaned harmonized variables we will use in analysis.
```

```
arise["workforce_challenge_1to5"] = arise["Q8"] if "Q8" in arise.columns else np.nan  
arise["fiscal_condition_1to5"] = arise["Q119"] if "Q119" in arise.columns else np.nan  
arise["grant_capacity_1to5"] = arise["Q5"] if "Q5" in arise.columns else np.nan
```

```
arise["survey_source"] = "ARISE"
```

```
arise[["workforce_challenge_1to5", "fiscal_condition_1to5", "grant_capacity_1to5", "survey_source"]].head()
```

	workforce_challenge_1to5	fiscal_condition_1to5	grant_capacity_1to5	survey_source
0	3.0	3.0	3.0	ARISE
1	4.0	4.0	2.0	ARISE
2	5.0	4.0	3.0	ARISE
3	3.0	3.0	2.0	ARISE
4	3.0	3.0	3.0	ARISE

Preparedness Capacity Score (Q9 Composite)

Preparedness capacity was operationalized using Q9 program indicators (e.g., early warning systems, evacuation plans, backup infrastructure).

Method used in code:

- Automatically detected all columns starting with “Q9_”
- Cleaned values to ensure valid binary representation (0/1)
- Preserved missing values (NaN) instead of converting them to 0

Computed:

preparedness_score = row-wise sum of Q9 items (min_count=1)

Vulnerable Population Support Score (Q10 Composite) Support for vulnerable groups (e.g., elderly, disabled, low-income populations) was captured using Q10 indicators.

Method used in code:

- Identified all Q10 columns dynamically
- Converted responses to numeric and preserved missing values

Computed:

- vulnerable_support_score = row-wise sum of Q10 items

```
# These are preparedness (Q9) and vulnerable support (Q10) items

q9_cols = [c for c in arise.columns if c.startswith("Q9_")]
q10_cols = [c for c in arise.columns if c.startswith("Q10_")]

print("Q9 columns found:", len(q9_cols))
print(q9_cols)

print("\nQ10 columns found:", len(q10_cols))
print(q10_cols)

Q9 columns found: 14
['Q9_Early warning', 'Q9_Evacuation plan', 'Q9_Financial assistance for low-income AC', 'Q9_Water conservation programs', 'Q9_Energy conservation programs', 'Q9_Zoning', 'Q9_Financial assistance for low-income shut-offs', 'Q9_Heating or cooling stations', 'Q9_Tornado shelter', 'Q9_Early warning_Lang', 'Q9_Code enforcement', 'Q9_Backup electric', 'Q9_Evacuation route or plan', 'Q9_None of the above']

Q10 columns found: 8
['Q10_Elderly people', 'Q10_Low income', 'Q10_Homeless', 'Q10_Non-English', 'Q10_Racial minorities', 'Q10_Disabled', 'Q10_Immigrants', 'Q10_One of the above']

# cleaning Q9 Q10

def clean_yes_no(series):
    x = pd.to_numeric(series, errors="coerce")
    # Keep only 0 and 1, everything else becomes NaN (safer for survey data)
    return x.where(x.isin([0, 1]), np.nan)

# Apply cleaning to Q9 and Q10 columns
for col in q9_cols:
    arise[col] = clean_yes_no(arise[col])

for col in q10_cols:
    arise[col] = clean_yes_no(arise[col])

print("Cleaned Q9 and Q10 values (missing preserved correctly).")

Cleaned Q9 and Q10 values (missing preserved correctly).
```

- This produces a robust composite index representing the breadth of vulnerable population support initiatives.

```
: arise["preparedness_score"] = arise[q9_cols].sum(axis=1, min_count=1)
  arise["vulnerable_support_score"] = arise[q10_cols].sum(axis=1, min_count=1)

# Quick preview
  arise[["preparedness_score", "vulnerable_support_score"]].head()
```

```
:   preparedness_score  vulnerable_support_score
```

	preparedness_score	vulnerable_support_score
0	3	1
1	5	1
2	4	1
3	9	3
4	6	1

6. Correlation and Diagnostic Analysis

Correlation analysis was conducted as an optional exploratory diagnostic step to examine relationships among:

- Workforce challenges
- Fiscal condition
- Grant capacity
- Preparedness score
- Vulnerable support score

This step was included for validation and exploratory insight, but it is not the primary focus of the data cleaning deliverable.

7. Limitations

CivicPulse response dataset is not currently available; therefore cross-survey harmonized comparison cannot yet be performed.

Survey items differ in wording and scope, requiring careful harmonization through the crosswalk framework.

Composite indices (Q9/Q10) depend on self-reported program availability and may contain inherent reporting bias.

8. Conclusion

This submission successfully implements a reproducible ARISE data cleaning and harmonization pipeline aligned with the finalized crosswalk table. The workflow standardizes missing values, corrects inconsistent entries, and constructs theoretically grounded composite indices (preparedness and vulnerable support).

9.Next Week Goal: Harmonization and Variable Alignment

Next week we will implement full harmonization using the survey codebooks as the primary reference for:

- 1.Variable definitions and construct meaning,
- 2.Response coding (scale values and missing codes),
- 3.Direction consistency (ensuring higher values represent the same interpretation across datasets),
- 4.Mapping equivalent items across ARISE and CivicPulse.