Research Data Services

Data Literacy Series, n.3/2025

Decoding Character Encoding Issues

Character encoding is the system of assigning numeric correspondence to characters, such as letters and symbols, so computers can store, process, and share them. Encoding issues have become less common in data handling with the widespread adoption of the UTF-8 standard. Still, some researchers may experience problems when working with data from legacy systems or old databases. Here, we cover some of the basics of character encoding standards and tips for researchers to avoid potential problems.

Comparing Common Character Encoding Standards

Feature	UTF-8	ISO-8859-1 (or Latin-1)	ASCII
Supported Character Set	Universal; supports all Unicode characters.	Supports Western European characters and some symbols.	Limited to 128 characters (basic English letters, digits, punctuation).
Accented Characters	Fully supported (e.g., é, ñ, ü, etc.)	Supported for many Western European characters (e.g., é, ñ), but not all.	Not supported; will display as ? or missing.
Special Symbols	Fully supported (e.g., ©, €, ±, ∑)	Some supported (e.g., ©, ±) but others (e.g., €) will display incorrectly.	Not supported (e.g., €, ∑) will display as ? or missing.
Emojis	Fully supported	Not supported; will display as ? or missing.	Not supported; will display as ? or missing.
Mathematical Symbols	Fully supported (e.g., ±, ∑)	Some supported (e.g., ±), others not (e.g., ∑).	Not supported; will display as ? or missing.
Languages	Fully supports characters from all languages (e.g., 中文, привет).	Supports many European languages (e.g., Spanish, French), but not Asian scripts or Cyrillic.	Only supports basic English characters.
Error Handling	Robust error handling; corrupt or unsupported characters are easy to identify.	Unsupported characters often display as ?, leading to partial data loss or corruption.	Errors in unsupported characters result in ? or loss of information.

While GoogleSheets and Numbers use UTF-8 by default, Microsoft Excel for Windows uses a machine-specific ANSI encoding (e.g., Windows-1252 for English), which may not handle all characters correctly. To be safe, always save your spreadsheets as CSV UTF-8 before sharing them with others.

Example

The example below illustrates how UTF-8 encoded text, including symbols, emojis, and accented characters, is misinterpreted when viewed in encodings like ISO-8859-1 or ASCII:

UTF-8	"Using characters such as ©, ñ, €, ¥, €, emojis 😊, mathematical symbols like ± and ∑, and foreign characters like 中文 (Chinese), along with accented words such as déjà vu, in the wrong encoding can cause data corruption or display errors."
ISO-8859-1	"Using characters such as �, �, ?, �, ?, emojis ?, mathematical symbols like � and ?, and foreign characters like ?? (Chinese), along with accented words such as d�j� vu, in the wrong encoding can cause data corruption or display errors."
ASCII	"Using characters such as Â©, Ã±, â□¬, Â¥, â□¬, emojis ð□□□, mathematical symbols like Â± and â□□, and foreign characters like ä¸ æ□□ (Chinese), along with accented words such as dÃ©jÃ vu, in the wrong encoding can cause data corruption or display errors."

Recommendations

- ✓ Adopt UTF-8 as the default encoding whenever possible
- ✓ Specify encoding explicitly when reading and writing files to ensure proper handling
- ✓ Avoid mixing encodings. If the file is in ASCII, keep it in that format unless conversion is necessary
- ✓ Always check the final file to ensure characters are displaying correctly before sharing or uploading it to a repository

Check & Convert

You can use the following commands in Terminal or Git Bash to check the encoding of your file and convert it if necessary:

```
#Check which encoding is currently in use

% file -I myfile.csv

#Output demonstrating iso-8859-1 as the current method

myfile.csv: text/plain; charset=iso-8859-1

#Convert and save it as a new file in UTF-8

iconv -f ISO-8859-1 -t UTF-8 myfile.csv > myfile-utf8.csv
```

UC **SANTA BARBARA**
Library

Need help? Contact us:
rds@library.ucsb.edu
www.library.ucsb.edu

Data Literacy Series, n.3/2025

Decoding Character Encoding Issues

Character encoding is the system of assigning numeric correspondence to characters, such as letters and symbols, so computers can store, process, and share them. Encoding issues have become less common in data handling with the widespread adoption of the UTF-8 standard. Still, some researchers may experience problems when working with data from legacy systems or old databases. Here, we cover some of the basics of character encoding standards and tips for researchers to avoid potential problems.

Comparing Common Character Encoding Standards

Feature	UTF-8	ISO-8859-1 (or Latin-1)	ASCII
Supported Character Set	Universal; supports all Unicode characters.	Supports Western European characters and some symbols.	Limited to 128 characters (basic English letters, digits, punctuation).
Accented Characters	Fully supported (e.g., é, ñ, ü, etc.)	Supported for many Western European characters (e.g., é, ñ), but not all.	Not supported; will display as ? or missing.
Special Symbols	Fully supported (e.g., ©, €, ±, ∑)	Some supported (e.g., ©, ±) but others (e.g., €) will display incorrectly.	Not supported (e.g., €, ∑) will display as ? or missing.
Emojis	Fully supported	Not supported; will display as ? or missing.	Not supported; will display as ? or missing.
Mathematical Symbols	Fully supported (e.g., ±, ∑)	Some supported (e.g., ±), others not (e.g., ∑).	Not supported; will display as ? or missing.
Languages	Fully supports characters from all languages (e.g., 中文, привет).	Supports many European languages (e.g., Spanish, French), but not Asian scripts or Cyrillic.	Only supports basic English characters.
Error Handling	Robust error handling; corrupt or unsupported characters are easy to identify.	Unsupported characters often display as ?, leading to partial data loss or corruption.	Errors in unsupported characters result in ? or loss of information.

While GoogleSheets and Numbers use UTF-8 by default, Microsoft Excel for Windows uses a machine-specific ANSI encoding (e.g., Windows-1252 for English), which may not handle all characters correctly. To be safe, always save your spreadsheets as CSV UTF-8 before sharing them with others.

Example

The example below illustrates how UTF-8 encoded text, including symbols, emojis, and accented characters, is misinterpreted when viewed in encodings like ISO-8859-1 or ASCII:

UTF-8	"Using characters such as ©, ñ, €, ¥, €, emojis 😊, mathematical symbols like ± and ∑, and foreign characters like 中文 (Chinese), along with accented words such as déjà vu, in the wrong encoding can cause data corruption or display errors."
ISO-8859-1	"Using characters such as �, �, ?, �, ?, emojis ?, mathematical symbols like � and ?, and foreign characters like ?? (Chinese), along with accented words such as d�j� vu, in the wrong encoding can cause data corruption or display errors."
ASCII	"Using characters such as Â©, Ã±, â□¬, Â¥, â□¬, emojis ð□□□, mathematical symbols like Â± and â□□, and foreign characters like ä¸ æ□□ (Chinese), along with accented words such as dÃ©jÃ vu, in the wrong encoding can cause data corruption or display errors."

Recommendations

- ✓ Adopt UTF-8 as the default encoding whenever possible
- ✓ Specify encoding explicitly when reading and writing files to ensure proper handling
- ✓ Avoid mixing encodings. If the file is in ASCII, keep it in that format unless conversion is necessary
- ✓ Always check the final file to ensure characters are displaying correctly before sharing or uploading it to a repository

Check & Convert

You can use the following commands in Terminal or Git Bash to check the encoding of your file and convert it if necessary:

```
#Check which encoding is currently in use

% file -I myfile.csv

#Output demonstrating iso-8859-1 as the current method

myfile.csv: text/plain; charset=iso-8859-1

#Convert and save it as a new file in UTF-8

iconv -f ISO-8859-1 -t UTF-8 myfile.csv > myfile-utf8.csv
```

UC SANTA BARBARA
Library

Need help? Contact us:
rds@library.ucsb.edu
www.library.ucsb.edu