Week 3 Slide Deck

Wrangling, Filtering; Formats

Jack Bandy 2026

Data Formats

How Do We Store This Data?

Spreadsheet??

- You've probably seen this
- Grid with cells, rows, columns (and sheets/tabs)
- Excel, Google Sheets, Numbers, and LibreOffice Calc
- Convenient for manual entry, review, sharing
- But don't keep your data in a spreadsheet...

	A	B	C	D	E
1	name	company	street	city	phone
2	Tyler Durden	Paper Street Soap Co.	537 Paper Street	Bradford	(288) 555-0153

CSV

- Comma-separated values
- Plain-text table: one row per record, one delimiter between fields
- Often used for spreadsheets, exports, and simple datasets
- Easy to inspect
- types and hierarchy are usually implicit

```
1 name,company,product,street,city,postalCode,phone
2 Tyler Durden,Paper Street Soap Co.,All Natural Handmade,537 Paper Stre
```

TSV

- Tab-separated values
- Plain-text table like CSV, but fields are separated by tabs
- Still flat: hierarchy and data types need outside context

```
name    company product street  city    postalCode  phone
Tyler Durden    Paper Street Soap Co.   All Natural Handmade    537 Pa
```

JSON

- JavaScript Object Notation
- Text format for data interchange
- Built from objects, arrays, strings, numbers, booleans, and null
- Common for web APIs and configuration files

```json
{
  "name": "Tyler Durden",
  "company": "Paper Street Soap Co.",
  "product": "All Natural Handmade",
  "address": {
    "street": "537 Paper Street",
    "city": "Bradford",
    "postalCode": "19808"
  },
  "phone": "(288) 555-0153"
}
```

XML

- Extensible Markup Language
- Nested tags represent elements and attributes
- Verbose, but widely used by older document systems

```xml
<?xml version="1.0" encoding="UTF-8"?>
<person>
  <name>Tyler Durden</name>
  <company>Paper Street Soap Co.</company>
  <product>All Natural Handmade</product>
  <address>
    <street>537 Paper Street</street>
    <city>Bradford</city>
    <postalCode>19808</postalCode>
  </address>
  <phone>(288) 555-0153</phone>
</person>
```

YAML

- YAML Ain't Markup Language
- Human-readable format based on indentation
- Supports mappings, lists, scalars, and comments
- Common for configuration files and data pipelines

```yaml
name: Tyler Durden
company: Paper Street Soap Co.
product: All Natural Handmade
address:
  street: 537 Paper Street
  city: Bradford
  postalCode: 19808
phone: (288) 555-0153
```

Parquet

- Binary columnar storage format
- Efficient for large datasets
- Stores schema and data types with the data
- Common in Spark, DuckDB, Polars, and “data lakes”

```
message business_card {
  required binary name (STRING);
  required binary company (STRING);
  required binary product (STRING);
  required binary street (STRING);
  required binary city (STRING);
  required binary postalCode (STRING);
  required binary phone (STRING);
}

row 1:
Tyler Durden | Paper Street Soap Co. | All Natural Handmade |
537 Paper Street | Bradford | 19808 | (288) 555-0153
```

Format Summary

Format	Best fit	Short example
CSV	Flat tables, spreadsheet exports, simple datasets	`name,company,phone` `Tyler Durden,Paper Street Soap Co.,(288) 555-0153`
TSV	Flat text tables where commas may appear in fields	`name company phone`
JSON	APIs, nested records, web data	`{"name":"Tyler Durden","city":"Bradford"}`
XML	Document-like data with tags and attributes	`<name>Tyler Durden</name>`
YAML	Human-edited configuration and pipeline settings	`name: Tyler Durden` `city: Bradford`
Parquet	Typed, compressed analytics data	`name: STRING` `city: STRING`

Sources

1. GitHub source: https://github.com/jackbandy/data-science-fun/blob/main/docs/slides/week3.md.
2. Slide deck built with Quarto revealjs.
3. Format examples adapted from Wikipedia and project documentation: JSON, XML, Comma-separated values, YAML, Tab-separated values, Spreadsheet, and Apache Parquet.
4. Tyler Durden business card image: Wikimedia Commons remake by Michaelpreid, modified, CC BY-SA 4.0, https://commons.wikimedia.org/wiki/File:Tyler_Durden_Business_Card.png.
5. Title font is Big Shoulders; Body font is Libre Franklin.