Parsing Raw Data

Faculty of Information Technology
Monash University, Australia

FIT5196 week 3

Outline

1. Extracting Data From CSV files

2. Extracting Data From XML files

3. Extracting Data From JSON files

4. Extracting Data From PDF files

5. Summary

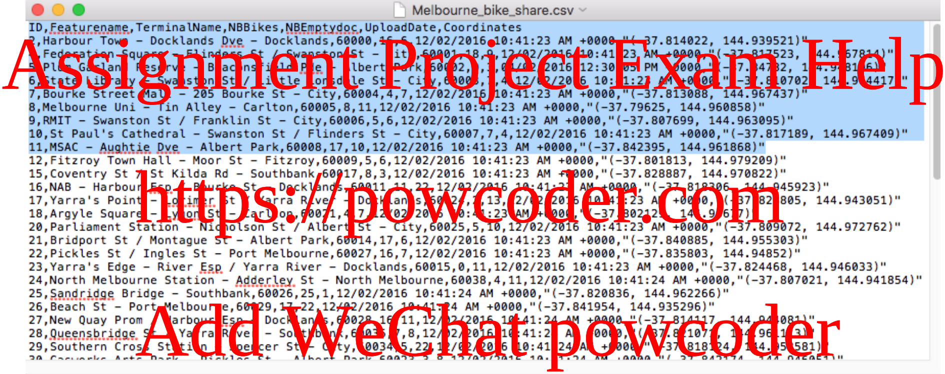
Data File Formats: Covered In This Unit

- Easy-to-parse (machine-readable) formats:
  - CSV: Comma Separated Values
  - JSON: JavaScript Object Notation
  - XML: eXtensible Markup Language
- Hard-to-parse formats:
  - Excel
  - PDF: Portable Document Format

Data File Formats: Not Covered In this Unit

- RDF: Resource Description Framework
  - A standard model for data interchange on the Web.
  - RDF has features that facilitate data merging even if the underlying schemas differ
  - RDF supports the evolution of schemas over time without requiring all the data consumers to be changed.
  - RDF is ideal for storing graph data, such as Knowledge Graphs.
  - Python libs
    - `http://rdflib.readthedocs.io/en/3.4.0/intro_to_graphs.html`
- HDF5 : Hierarchical Data Format
  - HDF5 contains an internal file system-like node structure
  - HDF5 can stores multiple datasets and supports metadata
  - HDF5 is a good choice for efficiently read and write large datasets.
  - Python libs
    - PyTables
    - h5py
    - pandas.HDFStore()

CSV: Comma Separated Values



Melbourne_bike_share.csv

```
ID,Featurename,TerminalName,NBBikes,NBEmptydoc,UploadDate,Coordinates
2,Harbour Town - Docklands Dve - Docklands,60000,16,6,12/02/2016 10:41:23 AM +0000,"(-37.814022, 144.939521)"
4,Federation Square - Flinders St / Swanston St - City,60001,18,9,12/02/2016 10:41:23 AM +0000,"(-37.817523, 144.967814)"
5,Plum Garland Reserve - Beaconsfield Pde - Albert Park,60002,[illegible],[illegible],[illegible]/2016 12:30:05 PM +0000,"(-37.84732, 144.9[illegible]816)"
6,State Library - Swanston St / Little Lonsdale St - City,60003,[illegible],4,12/02/2016 10:41:23 AM +0000,"(-37.810702, 144.9[illegible]4417)"
7,Bourke Street Mall - 205 Bourke St - City,60004,4,7,12/02/2016 10:41:23 AM +0000,"(-37.813088, 144.967437)"
8,Melbourne Uni - Tin Alley - Carlton,60005,8,11,12/02/2016 10:41:23 AM +0000,"(-37.79625, 144.960858)"
9,RMIT - Swanston St / Franklin St - City,60006,5,6,12/02/2016 10:41:23 AM +0000,"(-37.807699, 144.963095)"
10,St Paul's Cathedral - Swanston St / Flinders St - City,60007,7,4,12/02/2016 10:41:23 AM +0000,"(-37.817189, 144.967409)"
11,MSAC - Aughtie Dve - Albert Park,60008,17,10,12/02/2016 10:41:23 AM +0000,"(-37.842395, 144.961868)"
12,Fitzroy Town Hall - Moor St - Fitzroy,60009,5,6,12/02/2016 10:41:23 AM +0000,"(-37.801813, 144.979209)"
15,Coventry St / St Kilda Rd - Southbank,60017,8,3,12/02/2016 10:41:23 AM +0000,"(-37.828887, 144.970822)"
16,NAB - Harbour Esp / Bourke St - Docklands,60011,11,22,12/02/2016 10:41:23 AM +0000,"(-37.818306, 144.945923)"
17,Yarra's Point - Lorimer St / Yarra River - Docklands,60024,2,13,12/02/2016 10:41:23 AM +0000,"(-37.82[illegible]805, 144.943051)"
18,Argyle Square - Lygon St - Carlton,60021,4,7,12/02/2016 10:41:23 AM +0000,"(-37.80212[illegible], 144.9[illegible]6[illegible]7)"
20,Parliament Station - Nicholson St / Albert St - City,60025,5,10,12/02/2016 10:41:23 AM +0000,"(-37.809072, 144.972762)"
21,Bridport St / Montague St - Albert Park,60014,17,6,12/02/2016 10:41:23 AM +0000,"(-37.840885, 144.955303)"
22,Pickles St / Ingles St - Port Melbourne,60027,16,7,12/02/2016 10:41:23 AM +0000,"(-37.835803, 144.94852)"
23,Yarra's Edge - River Esp / Yarra River - Docklands,60015,0,11,12/02/2016 10:41:23 AM +0000,"(-37.824468, 144.946033)"
24,North Melbourne Station - Adderley St - North Melbourne,60038,4,11,12/02/2016 10:41:24 AM +0000,"(-37.807021, 144.941854)"
25,Sandridge Bridge - Southbank,60026,25,1,12/02/2016 10:41:24 AM +0000,"(-37.820836, 144.962266)"
26,Beach St - Port Melbourne,60029,17,22,12/02/2016 10:41:24 AM +0000,"(-37.841954, 144.935296)"
27,New Quay Prom - Harbour Esp - Docklands,60028,16,11,12/02/2016 10:41:24 AM +0000,"(-37.814117, 144.94[illegible]081)"
28,Queensbridge St - Yarra River - Southbank,6003[illegible],7,8,12/02/2016 10:41:24 AM +0000,"(-37.8[illegible]1071, 144.96[illegible]1[illegible]3)"
29,Southern Cross Station - Spencer St - City,60034,5,22,12/02/2016 10:41:24 AM +0000,"(-37.818[illegible]24, 144.95[illegible]581)"
30,Gasworks Arts Park - Pickles St - Albert Park,60023,3,8,12/02/2016 10:41:24 AM +0000,"(-37.842174, 144.945051)"
```

- TSV: Tab Separated Values
- Software: Microsoft Excel, Open Office Calc, and Google Spreadsheets.

CSV: tools

- Pandas functions for reading tabular data
  - read_csv(): Read delimited data from a file, URL, or file-like object. Use comma as default delimiter.
  - read_table(): Read delimited data from a file, URL, or file-like object. Use tab as default delimiter.
  - read_fwf(): read data in fixed-width column format, i.e., no delimiters.
  - read_clipboard(): Version of read_table that reads data from the clipboard. Useful for converting tables from web pages.

Outline

1 Extracting Data From CSV files

2 Extracting Data From XML files

3 Extracting Data From JSON files

4 Extracting Data From PDF files

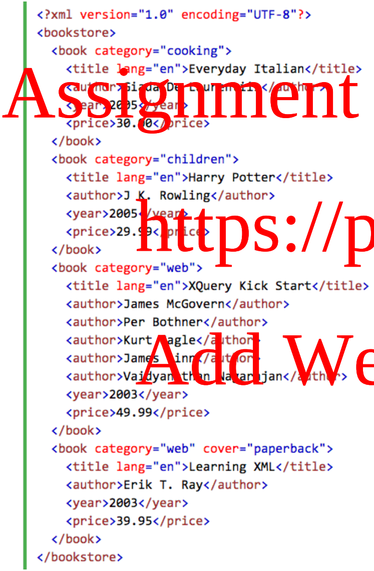
5 Summary

XML: Extensible Markup Language[1]

- XML is a software- and hardware-independent tool for storing and transporting data.
  - It simplifies data sharing and platform changes — no need to worry about issues of exchanging data between incompatible systems
  - It simplifies data transport — XML stores data in plain text format
  - It simplifies data availability — With XML, data can be available to all kinds of "reading machines"
- XML was designed to be both human- and machine-readable.

[1] Materials in the following 4 slides are based on http://www.w3schools.com/xml/default.asp

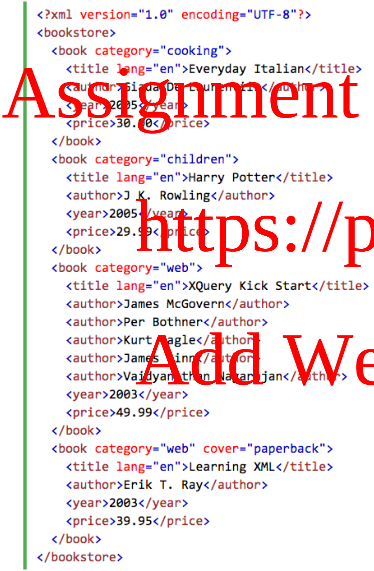
XML: DOM tree



```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

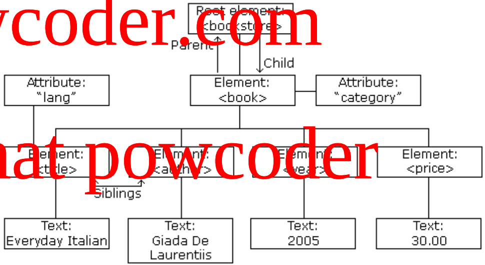
- According to the DOM (Document Object Model), everything in an XML document is a node
- The DOM says:
  - The entire document is a document node
  - Every XML element is an element node
  - The text in the XML elements are text nodes
  - Every attribute is an attribute node
  - Comments are comment nodes

XML: DOM tree



```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

- According to the DOM (Document Object Model), everything in an XML document is a node.



XML: tools

- ElementTree
  - https://docs.python.org/2/library/xml.etree.elementtree.html
  - Python's built-in XML parser.
- lxml:
  - http://lxml.de/
  - Strong performance in parsing very large files
- Beautifulsoup
  - https://www.crummy.com/software/BeautifulSoup/bs4/doc/
  - A Python library for pulling data out of HTML and XML files
  - Works with your favourite parser, e.g., html.parser and lxml-xml

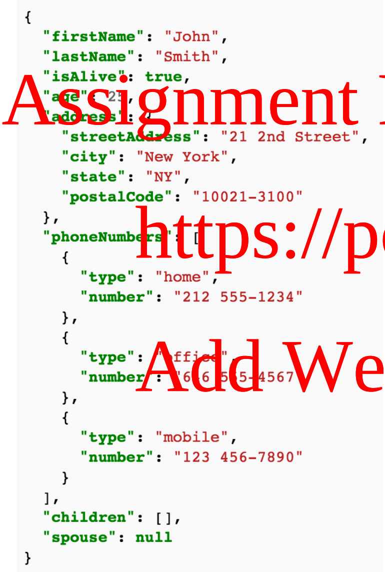
Demonstration with Jupyter notebook.

JSON: JavaScript Object Notation

- JSON: one of the most commonly used formats for transferring data between web services and other applications via HTTP.
- JSON is completely language independent but uses conventions that are familiar to programmers of the C-family of languages.
- JSON is built on two structures[2]:
  - A collection of name/value pairs. In various languages, this is realised as an object, record, struct, dictionary, hash table, keyed list, or associative array.
  - An ordered list of values. In most languages, this is realised as an array, vector, list, or sequence.

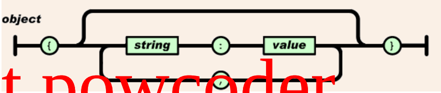
[2] Materials on the following 3 slides are based on http://www.json.org/

JSON: Structure

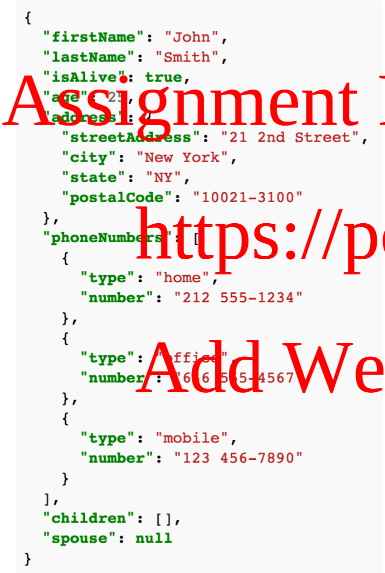


```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

- Three basic elements:
  - Object: an unordered set of name/value pairs

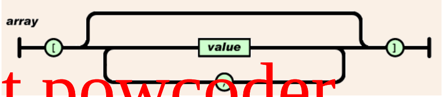


JSON: Structure

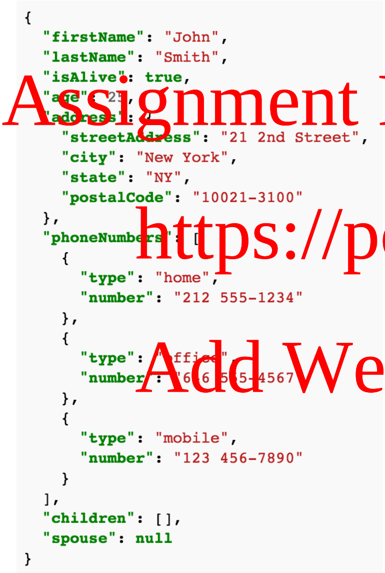


```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

- Three basic elements:
  - Array: an array is an ordered collection of values

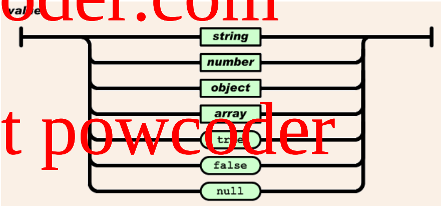


JSON: Structure



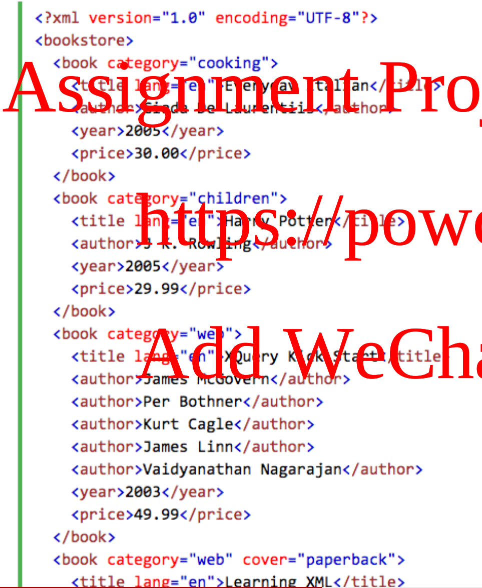
```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

- Three basic elements:
  - Value: a string in double quotes, or a number, or true or false or null, or an object or an array.

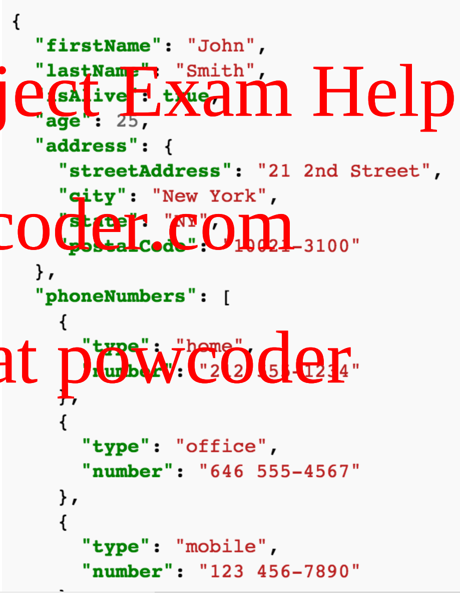


JSON v.s. XML

MONASH University



```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
```



```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
```

JSON: tools

- json: a built-in Python library used to parse JSON files.
- pandas json functions:
  - read_json(): Convert a JSON string to pandas object
  - json_normalize(): “Normalise” semi-structured JSON data into a flat table

PDF: Portable Document Format

- A file format used to present and exchange documents
  - "looks really do matter" from Adobe
    - PDF can contains text, image, link, button, form field, audio and video.
    - PDF file encapsulates a complete description of the layout information, e.g., fonts, graphics, and other meta information of the document.
- Not a data format

PDF: An example

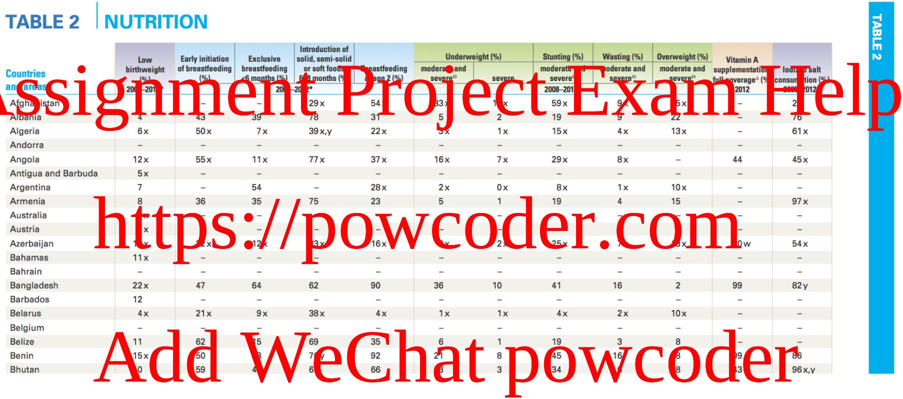


TABLE 2 | NUTRITION

Countries and areas	Low birthweight (%)	Early initiation of breastfeeding (%)	Exclusive breastfeeding <6 months (%)	Introduction of solid, semi-solid or soft foods 6–8 months (%)	Breastfeeding at age 2 (%)	Underweight (%) moderate and severe$^{\Theta}$	Underweight (%) severe	Stunting (%) moderate and severe$^{\Theta}$	Wasting (%) moderate and severe$^{\Theta}$	Overweight (%) moderate and severe$^{\Theta}$	Vitamin A supplementation full coverage$^{\Delta}$ (%)	Iodized salt consumption (%)
	2008–2012*	2008–2012*				2008–2012*					2012	2008–2012*
Afghanistan	–	–	–	29x	54x	33x	[illegible]	59x	9x	[illegible]	–	[illegible]
Albania	4	43	39	78	31	5	2	19	9	22	–	76
Algeria	6x	50x	7x	39x,y	22x	3x	1x	15x	4x	13x	–	61x
Andorra	–	–	–	–	–	–	–	–	–	–	–	–
Angola	12x	55x	11x	77x	37x	16x	7x	29x	8x	–	44	45x
Antigua and Barbuda	5x	–	–	–	–	–	–	–	–	–	–	–
Argentina	7	–	54	–	28x	2x	0x	8x	1x	10x	–	–
Armenia	8	36	35	75	23	5	1	19	4	15	–	97x
Australia	[illegible]	–	–	–	–	–	–	–	–	–	–	–
Austria	[illegible]	–	–	–	–	–	–	–	–	–	–	–
Azerbaijan	[illegible]	[illegible]	12x	[illegible]	16x	[illegible]	2x	25x	[illegible]	[illegible]	[illegible]	54x
Bahamas	11x	–	–	–	–	–	–	–	–	–	–	–
Bahrain	–	–	–	–	–	–	–	–	–	–	–	–
Bangladesh	22x	47	64	62	90	36	10	41	16	2	99	82y
Barbados	12	–	–	–	–	–	–	–	–	–	–	–
Belarus	4x	21x	9x	38x	4x	1x	1x	4x	2x	10x	–	–
Belgium	–	–	–	–	–	–	–	–	–	–	–	–
Belize	11	62	15	69	35	6	1	19	3	8	–	–
Benin	15x	50	[illegible]	[illegible]	92	21	8	45	16	8	99	86
Bhutan	[illegible]	59	[illegible]	[illegible]	66	[illegible]	3	34	[illegible]	8	43	96x,y

TABLE 2

PDF: Parsing Tools

- pdfminer: A tool for extracting text, images, object coordinates, metadata from PDF documents.
- pdftable: A tool for extracting tables from PDF files, it uses pdfminer to get information on the locations of text elements.
- slate: A small Python module that wraps pdfminer's API.
- Tabula: A simple tool for extracting data tables out of PDF files

Summary: what to do this week

1. Download, run and read the notebooks provided in Moodle, and also read the recommended reading materials associated with each notebook.
2. Try to finish the exercises in each chapter, and post your findings and experience in the discussion forum.
3. Attend tutorial 3 in the following week.
   - Parsing Excel files
4. **Assessment 1** released