FIT5196 Data wrangling S1 2018

Assessment 1: Parsing Data

Please carefully review all the requirements below to ensure you have a good understanding of what is required for your assessment.

1. **Due Date**
2. **Instructions & Brief**
3. **Assessment Resources**
4. **Assessment Criteria**
    1. **Grading Rubric**
    2. **Penalties**
5. **How to Submit**

Note: Please read each section carefully before you start doing the assessment tasks.

---

1. Due Date

This specific due date and time can be **viewed below in the Grading Summary**.

---

2. Assessment Description

Data from different resources are always stored in different formats, such as CSV, XML, JSON, Excel, and PDF. In this assessment, you are required to write Python (either Python 2 or Python 3) script to

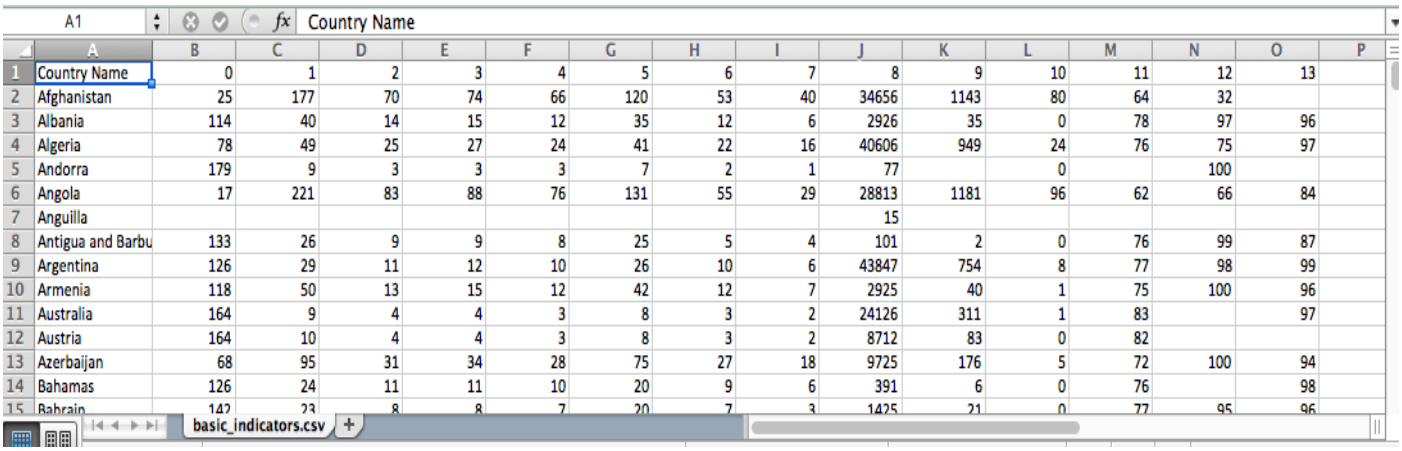
1. extract data from an Excel file,
2. extract data from a PDF file,
3. and convert data stored in an XML file to a JSON file.

The data we used in this assessment are originally derived from the UNICEF databases, which contain multiple statistical tables. These tables are used to support UNICEF's focus on the study of children's rights and development in the world.

This is an individual assignment and worth 30% of your total mark for FIT5196.

Details of tasks:

- **Task 1 Extract Data from an Excel File.** The first task is to extract the Basic Indicators table from an Excel file, **basic_indicators.xlsx**, and save the table in a CSV file as



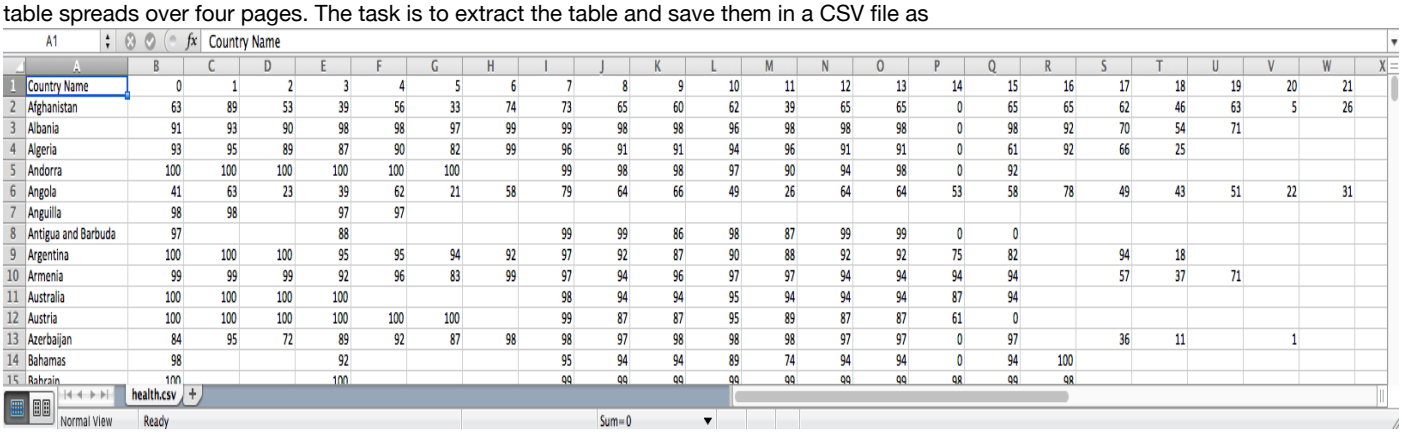
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Country Name	0	1	2	3	4	5	6	7	8	9	10	11	12	13
2	Afghanistan	25	177	70	74	66	120	53	40	34656	1143	80	64	32	
3	Albania	114	40	14	15	12	35	12	6	2926	35	0	78	97	96
4	Algeria	78	49	25	27	24	41	22	16	40606	949	24	76	75	97
5	Andorra	179	9	3	3	3	7	2	1	77		0		100	
6	Angola	17	221	83	88	76	131	55	29	28813	1181	96	62	66	84
7	Anguilla									15					
8	Antigua and Barbu	133	26	9	9	8	25	5	4	101	2	0	76	99	87
9	Argentina	126	29	11	12	10	26	10	6	43847	754	8	77	98	99
10	Armenia	118	50	13	15	12	42	12	7	2925	40	1	75	100	96
11	Australia	164	9	4	4	3	8	3	2	24126	311	1	83		97
12	Austria	164	10	4	4	3	8	3	2	8712	83	0	82		
13	Azerbaijan	68	95	31	34	28	75	27	18	9725	176	5	72	100	94
14	Bahamas	126	24	11	11	10	20	9	6	391	6	0	76		98

where the first column contains 202 country names, and the following 14 columns contain the different types of indicators. In order to finish this task,

  - you must correctly parse and extract the table;
  - existing Python packages can be used;

- it is **not required** to extract the column labels. Except for the first column, which should be named with "Country Name", the other columns should be indexed with integers as shown in the figure;
- your script must be written in a Jupyter notebook named as "**excel.ipynb**";
- and the extracted data must be saved in a CSV file named as "**basic_indicators.csv**";
- the input file must only be "**basic_indicators.xlsx**".

- **Task 2 Extract Data from a PDF File.** The PDF file, "**health.pdf**", contains the children's health data over 202 countries in the world. The table spreads over four pages. The task is to extract the table and save them in a CSV file as

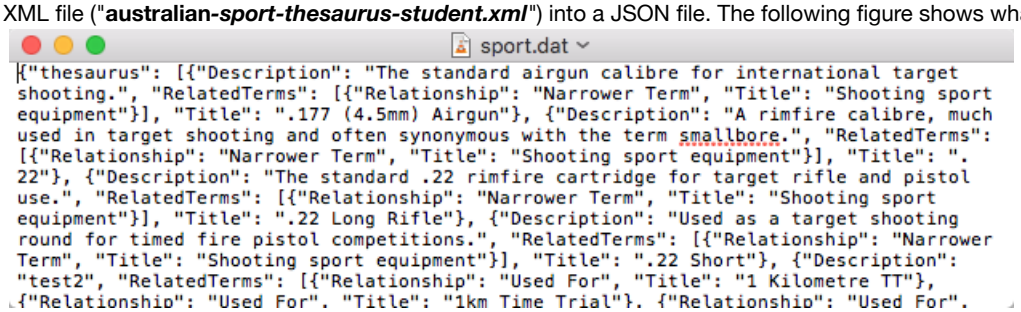


	Country Name	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	Afghanistan	63	89	53	39	56	33	74	73	65	60	62	39	65	65	0	65	65	62	46	63	5	26
3	Albania	91	93	90	98	98	97	99	99	98	98	96	98	98	98	0	98	92	70	54	71		
4	Algeria	93	95	89	87	90	82	99	96	91	91	94	96	91	91	0	61	92	66	25			
5	Andorra	100	100	100	100	100	100		99	98	98	97	90	94	98	0	92						
6	Angola	41	63	23	39	62	21	58	79	64	66	49	26	64	64	53	58	78	49	43	51	22	31
7	Anguilla	98	98		97	97																	
8	Antigua and Barbuda	97			88				99	99	86	98	87	99	99	0	0						
9	Argentina	100	100	100	95	95	94	92	97	92	87	90	88	92	92	75	82		94	18			
10	Armenia	99	99	99	92	96	83	99	97	94	96	97	97	94	94	94	94		57	37	71		
11	Australia	100	100	100	100				98	94	94	95	94	94	94	87	94						
12	Austria	100	100	100	100	100	100		99	87	87	95	89	87	87	61	0						
13	Azerbaijan	84	95	72	89	92	87	98	98	97	98	98	98	97	97	0	97		36	11		1	
14	Bahamas	98			92				95	94	94	89	74	94	94	0	94	100					
15	Bahrain	100			100				99	99	99	99	99	99	99	98	99	98					

where the first column contains the country names, and the following 22 columns contain various health information. In order to finish this task,

- you must correctly parse and extract the table;
- existing Python packages (like: pdfminer or pdftables) can be used, however, APIs (like pdftables_api), which requires API keys to push the PDF file to the server in order to get the file parsed, must not be used;
- it is **not required** to extract the column labels. Except for the first column, which should be named as "Country Name", the other columns should be indexed with integers as shown in the figure;
- if the number followed by a character "x" in the pdf file, "x" must be dropped in your script;
- your script must be written in a Jupyter notebook named as "**pdf.ipynb**";
- and the extracted table should be saved in a CSV file named as "**health.csv**";
- the input file must only be "**health.pdf**".

- **Task 3 Convert Data Stored in an XML File to a JSON File.** This task focuses on converting the Austrailian Sport Thesaurus stored in an XML file ("**australian-*sport-thesaurus-student.xml***") into a JSON file. The following figure shows what the JSON file should look like.



In order to finish this task,

- you must correctly extract the thesaurus in the XML file and store it in the JSON file;
- while extracting the thesaurus from the XML file, existing Python Packages that are written to parse XML files (e.g., Beautiful-soup, lxml and ElementTree) **must not be used**. You must write your own Python script to extract the thesaurus. **Hint**: Regular Expressions can be used.
- Python packages, like json, can be used to save the extracted thesaurus;
- your script must be written in a Jupyter notebook named as "**xml_json.ipynb**";
- the JSON data should be saved in a file named as "**sport.dat**";
- the input file must only be "**australian-*sport-thesaurus-student.xml***".

For all the tasks, the provided files cannot be modified before being loaded into your scripts. Manually parsing the files is not acceptable. In other words, the three tasks must be implemented in Python.

---

3. Assessment Resources

Before you start writing your code, you will need to download the following data file:

- basic_indicators.xlsx: contains a statistical table of five types of basic indicators.
- health.pdf contains a statistical table of health indicators.
- australian-sport-thesaurus-student.xml contains The *Australian Sports Thesaurus, which is* an accumulation of over 15,000 terms and definitions covering concepts and objects related to sport and sporting organizations with a strong emphasis on Australian sport.

- notebook template for this assessment. While using Jupyter Notebook, you should consider the use of different cells to make notes as you go. However, be precise and concise, please do not include things that are not relevant to the tasks in your notebook.

You must ensure that you can clearly explain what your code does with clear comments.You may need to review the FIT citation style tutorial to make you're familiar with appropriate citing and referencing for this assessment. Also, review the demystifying citing and referencing for help.

---

4. Assessment Criteria

The following outlines the criteria which you will be assessed against.

4.1 Grading Rubric

Grading details for this assessment are:

1. The submitted scripts in the notebook should work without any errors and should give the correct results. If the submitted notebook cannot be run by the assessor, which will be double-checked by the head tutor and the lecturer, zero marks will then be given to the corresponding task.
   - task 1: 6 out of 30
   - task 2: 11 out of 30
   - task 3: 11 out of 30
2. The code should be well structured and properly commented
   - 1 out of 30
3. The notebook should be structured in a logical way so that it clearly shows how students finish the tasks in the assessment.
   - 1 out of 30
4. Criteria 2 and 3 will be assessed if and only if the mark for criteria 1 is greater than or equal to 20.

4.2 Penalties

- Late submission: for all assessment items handed in after the official due date, and without an approved extension, a 5% penalty applies to the student's mark for each day after the due date (including weekends, and public holidays) for up to 10 days. **Assessment items handed in after 10 days will not be considered.**

---

5. How to Submit

Once you have completed your work, take the following steps to submit your work.

1. **Save** all of your files as a **ZIP file, named as "surname_studentID_ass1.zip",** which includes:
   1. Three Python notebooks (**Note:** please clean all the outputs in the notebooks.):
      - excel.ipynb
      - pdf.ipynb
      - xml_json.ipynb
   2. All the output files including
      - basic_indicators.csv
      - health.csv
      - sport.dat
   3. The original data files
      - basic_indicators.xlsx
      - health.pdf
      - australian-sport-thesaurus-student.xml
2. Click the **Add Submission** button below to submit and upload your assignment. **Please do remember to accept the submission statement! Only the submitted assignments will be marked. Those shown as a draft won't be marked!**

If you need further guidance on how to submit an assessment item, please review the **Submitting an assessment** overview. If you need further assistance, please go to **Help & Support**.

Submission status

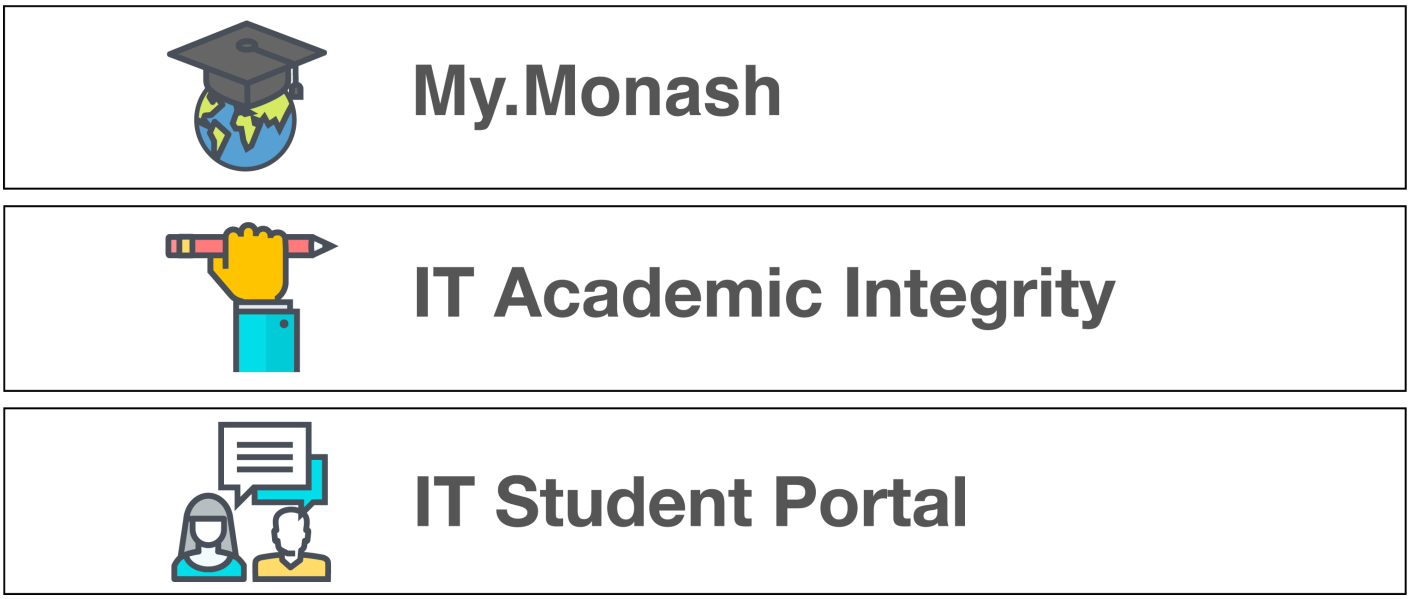
Attempt number	This is attempt 1.
Submission status	No attempt
Grading status	Not graded
Due date	Sunday, 8 April 2018, 11:55 PM
Time remaining	7 days

Last modified	-
Submission comments	Comments (0)

Add submission

Make changes to your submission

QUICK LINKS



Click here to access the Academic integrity course

NAVIGATION

Dashboard
- Site home

Site pages