SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

Aug 2024

Data Manipulation Seminar 4

ICT233 Data Programming

Veronica Hu huhe001@suss.edu.sg

RECAP

S3 Key Learning Objectives

1) Appreciate the features and usage possibilities of the Pandas library as a data analytics package.
2) Understand the basic usage of the Python Pandas library, including loading files, counting data, and determining item structure and types in the data.
3) Learn the basic manipulation of data using Pandas, such as row and column selection, and itemized or vector operations on Pandas DataFrame.
4) Conduct operations on Pandas DataFrame, including subsetting, slicing, and indexing.
5) Present and visualize data in Pandas DataFrame using charting and plotting libraries like Matplotlib and Seaborn.

SEMINAR OVERVIEW

Data Manipulation – LEARNING OBJECTIVES

1) Assemble datasets together for analysis using Pandas
2) Understand the needs of concatenating datasets and performing the operations on them
3) Understand the needs of merging datasets and performing the operations on them
4) Learn what missing data are and how they are created
5) Work with data issues such as missing and incomplete data during analysis
6) Learn how to use pivot, melt, and normalization operations on datasets

ETL Process

Seminar 4

Extract

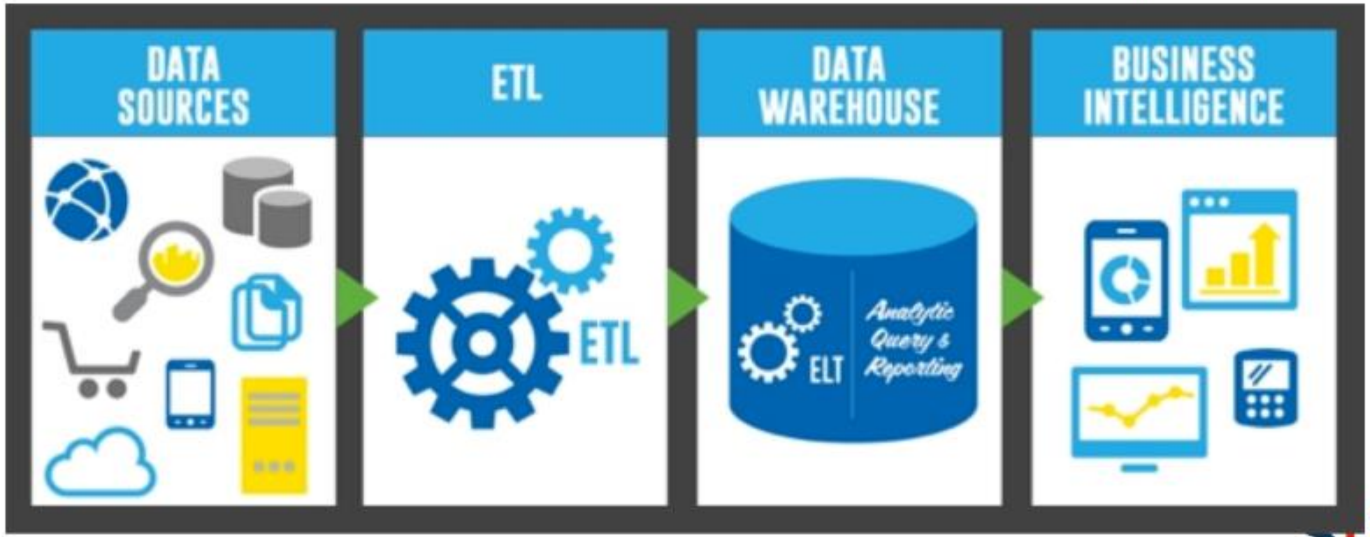
- One or more source systems containing customer, financial, or product data (CRM, Accounting system, Warehouse, MES)
- Files types - Flat files, XML, Oracle, IBM DB2, SQL Server,, IBM Websphere MQ, ODBC, JDBC, Hadoop Distributed File System (HDFS), Hive/HCatalog, JSON, Mainframe (IBM z/OS), Salesforce.com, SAP/R3

Transform

- Applying business rules, cleansing, and validating the data.
- Aggregation, Copy, Join, Sort, Merge, Partition, Filter, Reformat, Lookup
- Mathematical: +, -, x, /, Abs, IsValidNumber, Mod, Pow, Rand, Round, Sqrt, ToNumber, Truncate, Average, Min, Max
- Logical: And, Or, Not, IfThenElse, RegEx, Variables
- Text: Concatenate, CharacterLengthOf, LengthOf, Pad, Replace, ToLower, ToText, ToUpper, Translate, Trim, Hash
- Date: DateAdd, DateDiff, DateLastDay, DatePart, IsValidDate
- Format: ASCII, EBCDIC, Unicode

Load

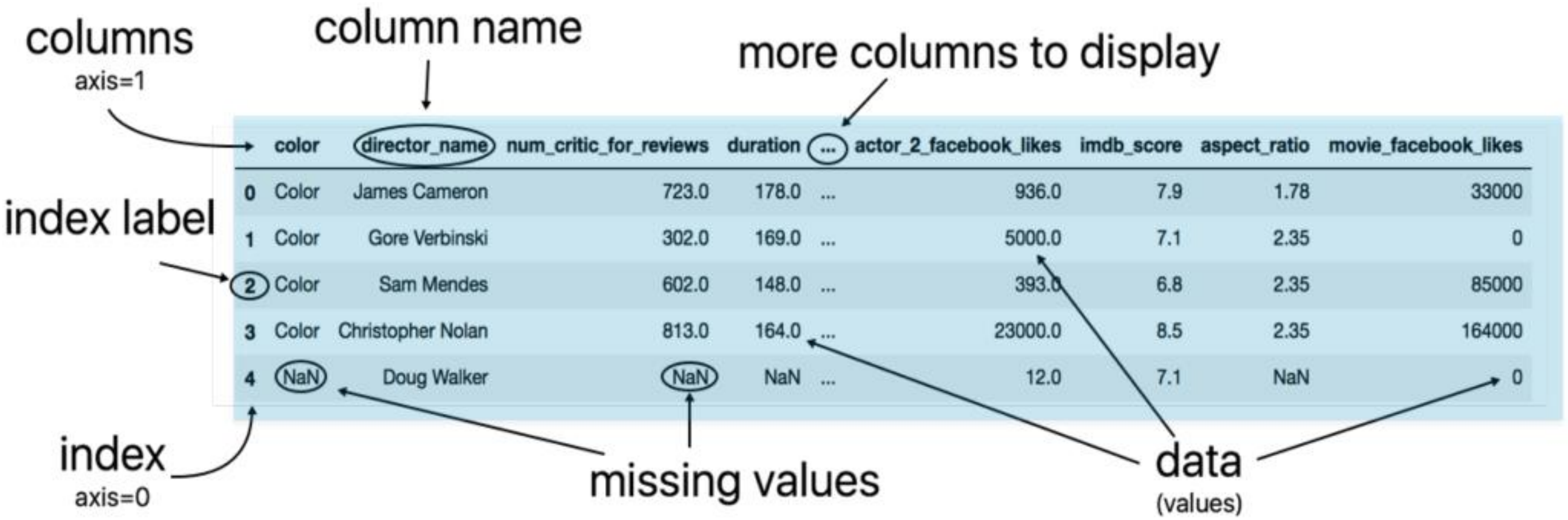
- Load the results into one or more target systems such as a data warehouse, datamart, or business intelligence reporting system.
- Output: Flat files, XML, Oracle, IBM DB2, SQL Server, Teradata, Sybase, Vertica, Netezza, Greenplum, ODBC, JDBC, Hadoop Distributed File System (HDFS), Hive/HCatalog, Mainframe (IBM z/OS), Salesforce.com, Tableau, QlikView



Chapter 1: Assembling Data

1.1 Combining DataFrames

- Tidy data can be seen to meet the following criteria:
  - Each row in an observation
  - Each column is a variable
  - Each type of observational unit forms a table



	color	director_name	num_critic_for_reviews	duration	...	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes
0	Color	James Cameron	723.0	178.0	...	936.0	7.9	1.78	33000
1	Color	Gore Verbinski	302.0	169.0	...	5000.0	7.1	2.35	0
2	Color	Sam Mendes	602.0	148.0	...	393.0	6.8	2.35	85000
3	Color	Christopher Nolan	813.0	164.0	...	23000.0	8.5	2.35	164000
4	NaN	Doug Walker	NaN	NaN	...	12.0	7.1	NaN	0

Chapter 1: Assembling Data

1.1 Combining DataFrames

Data cleaning/tidy processes

- Handling of missing values
    - Replacing nulls with values
        - Using another value
        - Using existing data
    - Drop the data from our data set
- Outliers

Chapter 1: Assembling Data

1.1 Combining DataFrames

- The need to combine Data / DataFrame
  - Finding the data you need
  - e.g. find stock prices within the tech industry
- When splitting the data into separate tables
  - Advantage
    - Reduce redundant information
  - Drawback
    - The need to combine relevant data to answer questions

Company information

		Company Info			KEY FINANCIALS				
Rank	Company Name	Country	Number of Employees	Previous Rank	Revenues (Smillions)	Revenue Change	Profits (Smillions)	Profit Change	Assets (Smillions)
1	Walmart	USA	2.300.000	1	$500.343	3,0%	$9.862,0	-27,7%	$204.522
2	State Grid	China	913.546	2	$348.903	10,7%	$9.533,4	-0,4%	$585.278
3	Sinopec Group	China	667.793	3	$326.953	22,2%	$1.537,8	22,2%	$346.545
4	China National Petroleum	China	1.470.193	4	$326.008	24,2%	-$690,5	-137,0%	$629.411
5	Royal Dutch Shell	Netherlands	84.000	7	$311.870	29,9%	$12.977,0	183,7%	$407.097
6	Toyota Motor	Japan	369.124	5	$265.172	4,1%	$22.510,1	33,2%	$473.133
7	Volkswagen	Germany	642.292	6	$260.028	8,2%	$13.107,3	120,8%	$506.956
8	BP	Britain	74.000	12	$244.582	31,1%	$3.389,0	2847,0%	$276.515
9	Exxon Mobil	USA	71.200	10	$244.363	17,4%	$19.710,0	151,4%	$348.691
10	Berkshire Hathaway	USA	377.000	8	$242.137	8,3%	$44.940,0	86,7%	$702.095
11	Apple	USA	123.000	9	$229.234	6,3%	$48.351,0	5,8%	$375.319
12	Samsung Electronics	South Korea	320.671	15	$211.940	21,8%	$36.575,4	89,3%	$281.906
13	McKesson	USA	68.000	11	$208.357	4,9%	$67,0	-98,7%	$60.381
14	Glencore	Switzerland	82.681	16	$205.476	18,2%	$5.777,0	318,9%	$135.593
15	UnitedHealth Group	USA	260.000	13	$201.159	8,8%	$10.558,0	50,5%	$139.058
16	Daimler	Germany	289.321	17	$185.235	9,3%	$11.863,9	25,8%	$306.922
17	CVS Health	USA	203.000	14	$184.765	4,1%	$6.622,0	24,5%	$95.131
18	Amazon.com	USA	566.000	26	$177.866	30,8%	$3.033,0	27,9%	$13.131
19	EXOR Group	Italy	307.637	20	$161.677	4,4%	$1.569,1	140,9%	$196.656

Stock information

52W high	52W low	Stock	Ticker	Div	Yield %	P/E	Vol 00s	High	Low	Close	Net chg
$45.39	19.75	ResMed	RMD			52.5	3831	42.00	39.51	41.50	-1.90
11.63	3.55	Revlon A	REV				162	6.09	5.90	6.09	+0.12
77.25	55.13	RioTinto	RTP	2.30	3.2		168	72.75	71.84	72.74	+0.03
31.31	16.63	RitchieBr	RBA			20.9	15	24.49	24.29	24.49	-0.01
8.44	1.75	RiteAid	RAD				31028	4.50	4.20	4.31	+0.21
$38.63	18.81	RobtHalf	RHI			26.5	6517	27.15	26.50	26.50	+0.14
51.25	27.69	Rockwell	ROK	1.02	2.1	14.5	6412	47.99	47.00	47.54	+0.24
Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12

Chapter 1: Assembling Data

1.1 Combining DataFrames - Concatenation

- Horizontally (axis=1)
- Vertically (axis=0)
- Concatenation with different indices
    - Missing data is introduced

```
horizontal_stacked = pd.concat([df1,df2,df3], axis=1)
display(horizontal_stacked)
```

	0	1	2	3	0	1	2	3	0	1	2	3
0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	3.0	3.0	3.0	3.0
1	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	3.0	3.0	3.0	3.0
2	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	3.0	3.0	3.0	3.0
3	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	3.0	3.0	3.0	3.0

```
# axis default as 0
verticall_stacked = pd.concat([df1,df2,df3], axis=0)
display(verticall_stacked)
```

	0	1	2	3
0	1.0	1.0	1.0	1.0
1	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0
0	2.0	2.0	2.0	2.0
1	2.0	2.0	2.0	2.0
2	2.0	2.0	2.0	2.0
3	2.0	2.0	2.0	2.0
0	3.0	3.0	3.0	3.0
1	3.0	3.0	3.0	3.0

Chapter 1: Assembling Data

1.1 Combining DataFrames - Concatenation

Reflections

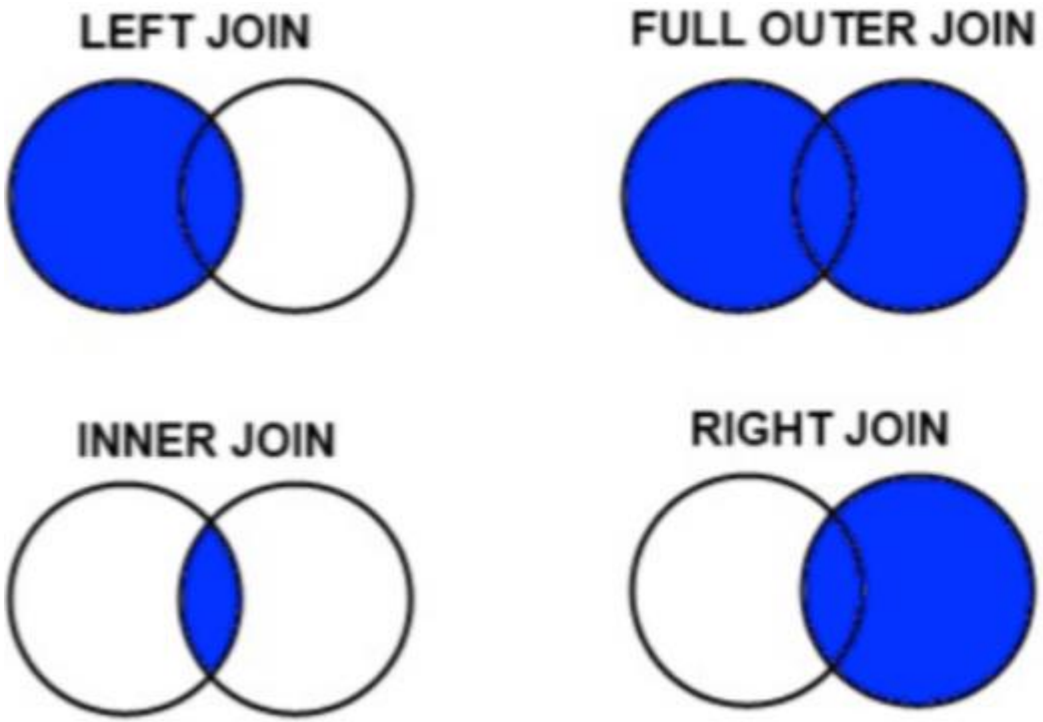
- When do we need to do concatenation operations on data?
- How do we calculate daily average sales for a retail shop, for the past 3 months?
  - When each file contains daily transaction details

Chapter 1: Assembling Data

1.2 Merging DataFrames

Merging Multiple Data Sets

Pandas	SQL	Description
left	left outer	Keep all the keys from the left
right	right outer	Keep all the keys from the right
outer	full outer	Keep all the keys from left & right
inner	inner	Keep only the keys that exist in both left and right




```
what_sites_were_visited = site.merge(visit, left_on='name', right_on='site', how='inner')

display(what_sites_were_visited)
```

	name	location	id	site	date
0	DR-1	location 1	1	DR-1	2018-02-20
1	DR-1	location 1	2	DR-1	2018-02-22
2	DR-2	location 2	3	DR-2	2018-02-25

Chapter 1: Assembling Data

1.2 Merging DataFrames

Reflection

- When do we need to do “inner” or “outer” join or merge operations on datasets ?
    - OUTER JOIN (LEFT or RIGHT)
        - Use when you want all rows from one table.
        - Retrieves only matching rows from the other table.
    - FULL OUTER JOIN:
        - Use when you want to get all rows from both tables.

Chapter 2: Handling Missing Data and Tidying up Information

2.1 Missing Information

Introduction

- Pandas displays missing values as NaN.
- NaN is not be equivalent to 0 or an empty string, ‘’
- Test for missing values

```
# import missing value defined in numpy library
from numpy import NaN, NAN, nan

import pandas as pd

print(pd.isnull(NaN))
```

```
True
```

```
print(pd.notnull(NaN), pd.notnull(888), pd.notnull('test'))
```

```
False True True
```

Chapter 2: Handling Missing Data and Tidying up Information

2.1 Missing Information

- Cleaning Missing Data
  - Testing for null values
  - Replacing nulls with values
    - Fill in with another value
    - Fill in using existing data
      - Fill Forward/Backward
      - Interpolate
  - Drop the data from our data set
- Calculations with Missing Data

Chapter 2: Handling Missing Data and Tidying up Information

2.2 Tidying and Organizing Information

- Multiple Columns with Same Variable
  - ‘wide’ data
  - melt()

	religion	<$10k	$10-20k	$20-30k	$30-40k	$40-50k	$50-75k	$75-100k	$100-150k	>150k	Don't know/refused
0	Agnostic	27	34	60	81	76	137	122	109	84	96
1	Atheist	12	27	37	52	35	70	73	59	74	76
2	Buddhist	27	21	30	34	33	58	62	39	53	54
3	Catholic	418	617	732	670	638	1116	949	792	633	1489
4	Don’t know/refused	15	14	15	11	10	35	21	17	18	116

Chapter 2: Handling Missing Data and Tidying up Information

2.2 Tidying and Organising Information

- Columns with Multiple Variables – e.g. Ebola dataset
  - a single column in a dataset may **represent** multiple variables
  - multiple steps to tidy the data

```
ebola.head()
```

	Date	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_Senegal
0	1/5/2015	289	2776.0	NaN	10030.0	NaN	NaN
1	1/4/2015	288	2775.0	NaN	9780.0	NaN	NaN
2	1/3/2015	287	2769.0	8166.0	9722.0	NaN	NaN
3	1/2/2015	286	NaN	8157.0	NaN	NaN	NaN
4	12/31/2014	284	2730.0	8115.0	9633.0	NaN	NaN

```
list(ebola)
```

```
['Date',
 'Day',
 'Cases_Guinea',
 'Cases_Liberia',
 'Cases_SierraLeone',
 'Cases_Nigeria',
 'Cases_Senegal',
 'Cases_UnitedStates',
 'Cases_Spain',
 'Cases_Mali',
 'Deaths_Guinea',
 'Deaths_Liberia',
 'Deaths_SierraLeone',
 'Deaths_Nigeria',
 'Deaths_Senegal',
 'Deaths_UnitedStates',
 'Deaths_Spain',
 'Deaths_Mali']
```

Chapter 2: Handling Missing Data and Tidying up Information

2.2 Tidying and Organizing Information

- Variables in Both Rows and Columns
  - multiple steps to tidy the data
  - melt() / pivot_table()

```
import pandas as pd
weather = pd.read_csv('weather.csv')
weather.head()
```

	id	year	month	element	d1	d2	d3	d4	d5	d6	...
0	MX17004	2010	1	tmax	NaN	NaN	NaN	NaN	NaN	NaN	...
1	MX17004	2010	1	tmin	NaN	NaN	NaN	NaN	NaN	NaN	...
2	MX17004	2010	2	tmax	NaN	27.3	24.1	NaN	NaN	NaN	...
3	MX17004	2010	2	tmin	NaN	14.4	14.4	NaN	NaN	NaN	...
4	MX17004	2010	3	tmax	NaN	NaN	NaN	NaN	32.1	NaN	...

Chapter 2: Handling Missing Data and Tidying up Information

2.2 Tidying and Organizing Information

Normalization

- Definition in Database Design:
  - Process of organizing and structuring data to eliminate redundancy.
  - Improves data integrity by defining dependencies and relationships between data.
- Key Steps:
  - Create tables to group related data.
  - Define relationships between attributes (columns) of these tables.

Chapter 2: Handling Missing Data and Tidying up Information

2.2 Tidying and Organising Information

Normalization

- Normalization in DataFrames
  - Starting Point:
    - Check if multiple observational units are represented in a single table.
  - Identify Redundancies:
    - Examine rows for cells or values that are repeated across rows.
  - Strategy:
    - Reorganize repeated information into separate tables.
    - Define relationships between these tables to maintain data integrity.

SUMMARY

Data Manipulation – LEARNING OBJECTIVES

1) Assemble Data sets together for analysis using Pandas
2) Understand the needs of concatenating data sets and performing the operations on them
3) Understand the needs of merging data sets and performing the operations on them
4) Learn what missing data are and how they are created
5) Work with data issues such as missing and incomplete data during analysis
6) Learn how to use pivot, melt and normalization operations on data sets

THANK YOU