**Pandas เบื้องต้น**

**การโหลดไฟล์ การเซฟไฟล์ ตัวแปรประเภท DataFrames การเลือกข้อมูล**

Pandas เป็นหนึ่งใน Python Libraries ที่ได้รับความนิยมเป็นอย่างมากสำหรับวิทยาการข้อมูล (Data Science ต่อจากนี้ไปจะขอใช้คำ Data Science แทนภาษาไทย) ก่อนจะเริ่มใช้งาน Pandas ก็ต้องทำการติดตั้ง Library นี้ (เช่นเดียวกับที่เราติดตั้ง Pygame)

1. **ให้นักเรียนติดตั้ง Pandas บนเครื่องคอมพิวเตอร์ที่ใช้งาน (แบบที่ติดตั้ง Pygame)**

หลังจากติดตั้ง Pandas และก่อนจะเริ่มต้นกันที่การโหลดข้อมูล (เปิดไฟล์) เบื้องต้น ขอให้นักเรียนทำความเข้าใจกับประเภทตัวแปรที่เป็น Data Structure ใน Pandas ได้แก่ ตัวแปรประเภท Series และ DataFrames

**Series:** ใน Pandas ตัวแปรประเภท Series เป็นโครงสร้างข้อมูลแบบหนึ่งมิติ (one dimensional data structure หรือ array หนึ่งมิติที่เก็บข้อมูลพร้อมกับ index เช่น



**DataFrame**: ใน pandas ตัวแปรประเภท DataFrame เป็นโครงสร้างข้อมูลแบบสองมิติ (หรือมากกว่าสองมิติ) ให้เข้าใจอย่างง่ายคือตารางข้อมูลแบบ rows และ columns โดยที่ columns ระบุชื่อ และ rows have เก็บค่า เช่น



ส่วนใหญ่ DataFrames จะเป็นที่นิยมเนื่องจากโดยทั่วไปเวลาทำการวิเคราะห์ข้อมูล ข้อมูลจะถูกจัดเก็บอยู่ในรูปแบบสองมิติเป็นส่วนใหญ่

ต่อไปจะเป็นวิธีการโหลดข้อมูลใน Pandas ก่อนอื่น ลองสร้างข้อมูลสองมิติแบบง่ายๆ โดยใช้ Jupyter Notebook ไฟล์ที่สร้างจะเป็น .CSV (Comma Separated Value หรือแต่ละค่าแยกกันด้วยเครื่องหมาย comma) จากข้อมูลดิบ

animal,uniq\_id,water\_need

elephant,1001,500

elephant,1002,600

elephant,1003,550

tiger,1004,300

tiger,1005,320

tiger,1006,330

tiger,1007,290

tiger,1008,310

zebra,1009,200

zebra,1010,220

zebra,1011,240

zebra,1012,230

zebra,1013,220

zebra,1014,100

zebra,1015,80

lion,1016,420

lion,1017,600

lion,1018,500

lion,1019,390

kangaroo,1020,410

kangaroo,1021,430

kangaroo,1022,410

ให้นักเรียนกดปุ่ม New ที่ Home Tab แล้วเลือก Text File



และทำการ Copy ข้อมูลดิบทั้งหมดจากด้านบนใส่ลงไป



เปลี่ยนชื่อเป็น zoo.csv



นักเรียนก็จะมีไฟล์ zoo.csv ปรากฎอยู่ในโฟลเดอร์เดียวกับ ipynb ที่มีอยู่

กลับมาที่ notebook เริ่มต้น ให้นักเรียนเปลี่ยนชื่อ .ipynb ตามรูปแบบต่อไปนี้ (โดยมีข้อมูล เลขที่ รหัสนักเรียน ชื่อ) เช่น

1\_45655\_Napas\_SamsenDA\_worksheet1.ipynb

อันดับแรกให้ทำการ import pandas และ numpy



ฟังก์ชันสำหรับการโหลดไฟล์ csv คือ read\_csv( )

1. **ให้นักเรียนทำการโหลดข้อมูลจาก zoo.csv โดยใช้คำสั่งต่อไปนี้**

**pd.read\_csv('zpp.csv', delimiter = ',') และกด run นักเรียนจะเห็นข้อมูลใน zoo.csv โหลดมาที่ Jupyter Notebook**

จากข้อมูลที่ปรากฎเลขแถวซ้ายสุดคือ index ใน Pandas DataFrames และชื่อในแต่ละ Column (Column names) ก็เอามาจากแถวแรกในไฟล์ zoo.csv โดยอัตโนมัติ

จาก exercise ข้างบนที่ผ่านมานักเรียนได้เรียนรู้วิธีการสร้างข้อมูลแบบสองมิติและเซฟในรูปแบบไฟล์ .csv

1. **ให้นักเรียนไปที่ portal และเซฟไฟล์ชื่อ pandas\_tutorial\_read.csv มาและนำไปไว้ที่โฟลเดอร์เดียวกันกับไฟล์ .ipynb ที่กำลังทำงานอยู่**
2. **ให้นักเรียนโหลดข้อมูลจาก pandas\_tutorial\_read.csv โดยใช้คำสั่งที่ใช้ใน exercise 2 และนักเรียนควรเห็นข้อมูลต่อไปนี้**



จากข้อมูลที่โหลดมาสังเกตได้ว่าไม่มี Column names ข้อมูลแถวแรกจะกลายเป็น Column names แทน ทางแก้คือไปเพิ่ม Column names ในไฟล์ต้นฉบับ แต่ในบางครั้งเราไม่ต้องการแก้ไขไฟล์ต้นฉบับ ดังนั้นทางแก้คือใช้ function ใน Pandas จัดการ

เราสามารถโหลดข้อมูลและเพิ่ม Column names เข้าไปพร้อมกัน โดยใช้คำสั่ง

pd.read\_csv('pandas\_tutorial\_read.csv', delimiter=';', names = ['my\_datetime', 'event', 'country', 'user\_id', 'source', 'topic'])

ที่ผ่านมาการโหลดข้อมูลและแสดงผลข้อมูลโดยใช้คำสั่ง pd.read\_csv( ) หากต้องการแสดงผลข้อมูลอีกครั้งก็ต้องพิมพ์คำสั่งเดิมซ้ำๆ วิธีการที่สะดวกกว่านั้นคือการโหลดข้อมูลและเก็บไว้ในตัวแปร เช่น

article\_read = pd.read\_csv('pandas\_tutorial\_read.csv', delimiter=';', names = ['my\_datetime', 'event', 'country', 'user\_id', 'source', 'topic'])

ข้อมูลจะถูกโหลดมาเก็บไว้ที่ article\_read หากต้องการแสดงค่าก็เรียกแค่ article\_read

1. **ให้นักเรียนเขียนคำสั่งข้างบน และ ทำการเรียก article\_read ใน cell ถัดไป**

ในบางครั้งการแสดงข้อมูลปริมาณมากเกินบนหน้าจอก็เป็นการสะดวก เราสามารถกำหนดปริมาณข้อมูลที่ต้อการให้แสดงได้ โดยใช้คำสั่ง article\_read.head() ซึ่งจะแสดงผลห้าแถวแรก (ไม่ต้องใส่ค่าอะไรในวงเล็บ default = 5) หากต้องการแสดงโดยระบุจำนวนแถวก็ได้

1. **ให้นักเรียนทำการแสดงผล 5 10 15 และ 20 แถว**

เราสามารถแสดงข้อมูลแถวท้ายๆ ได้ โดยใช้คำสั่ง article\_read.tail() หรือจะแสดงผลแบบสุ่มซักห้าแถวก็ได้ โดยใช้คำสั่ง article\_read.sample(5)

1. **ให้นักเรียนลองคำสั่งด้านบนทั้งสองเพื่อแสดงผลข้อมูล**

การแสดงผลข้อมูลเฉพาะ Column ก็สามารถทำได้ เช่น ต้องการแสดงผลเฉพาะ Column country และ user\_id โดยใช้คำสั่งต่อไปนี้ article\_read[['country', 'user\_id']]

1. **ให้นักเรียนลองแสดงผล country และ user\_id สลับกัน และสองแสดงผล Column อื่นๆ**

ในการวิเคราะห์ข้อมูล (โดยเฉพาะการทำ data prediction) เราอาจจะจำเป็นต้องใช้ข้อมูลแบบ Series เราสามารถดึงข้อมูลจาก DataFrames มาเป็น Series ได้โดยใช้คำสั่ง (เลือกอย่างใดอย่างหนึ่ง แบบแรก็จะพิมพ์ง่ายหน่อย)

article\_read.user\_id
article\_read['user\_id']

การ filter เพื่อแสดงข้อมูลแบบเฉพาะเจาะจง เช่น เราต้องการต้องการดูข้อมูลใน Column source ที่เป็นข้อมูลเฉพาะ 'SEO' เราต้องใช้คำสั่ง article\_read[article\_read.source == 'SEO']

ลองมาทำความเข้าใจว่า Pandas มองคำสั่งนี้อย่างไร สำหรับ article\_read.source == 'SEO' pandas จะดูทุกค่าใน DataFrames ใน column source ว่าเป็น SEO หรือไม่ ถ้าใช่ค่าที่ได้คือ True ถ้าไม่ใช่คือ False ในส่วนของ article\_read ก็จะแสดงผลเฉพาะแถวที่มีค่า source เป็น True



1. **ให้นักเรียนลองแสดงผล source ที่มาจาก Reddit และแสดงผล country เฉพาะ Europe**

ใน Pandas ตรรกะจะเป็นลักษณะ Linear อีกความหมายคือเราสามารถซ้อนฟังก์ชันได้ เช่น

article\_read.head()[['country', 'user\_id']]

ในกรณีนี้เราเลือกที่จะแสดงผลห้าแถว แล้วเจาะจงไปที่ country และ user\_id ในทางกลับกัน

article\_read[['country', 'user\_id']].head()

เราเลือก country และ user\_id ก่อนและให้แสดงผลห้าแถว ผลลัพธ์ที่ได้จากทั้งสองคำสั่งคือได้ผลเดียวกัน และลำดับการทำคำสั่งสลับกัน

1. **ให้นักเรียนลองคำสั่งด้านบนทั้งสองเพื่อดูผล**
2. **ให้นักเรียนแสดงข้อมูล user\_id, country และ topic เฉพาะที่มาจาก country\_2 เป็นจำนวน 5 แถว (Hint: นักเรียนต้อง filter ข้อมูลเอาเฉพาะ country\_2 และเลือกเฉพาะ user\_id, country, topic และเรียก .head() )**
3. **ให้นักเรียนแสดงข้อมูล user\_id source topic โดยมี source จาก Reddit และ topic เป็น Asia จำนวน 10 แถว**