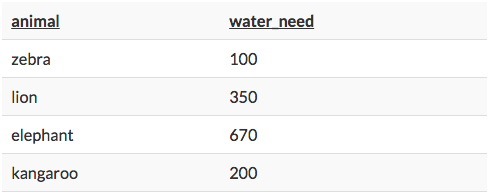
**Pandas**

**การรวมข้อมูล (Data Aggregation) และการจัดกลุ่ม (Grouping) ข้อมูล สถิติพื้นฐาน**

การรวมข้อมูล (Data Aggregation) มีวัตถุประสงค์เพื่อรวบรวมข้อมูลบางอย่างที่อยู่ใน dataset ให้เป็นข้อมูลเดียว เพื่อให้ง่ายต่อการวิเคราะห์ข้อมูล กลับมาที่ตัวอย่างข้อมูล zoo เช่น



รูปแบบหนึ่งสำหรับการทำ Data Aggregation คือการหาผลรวม เช่น หาผลรวมของ water\_need ซึ่งจะเท่ากับ 100+350+670+200 = 1320 หรืออีกแบบคือการหาผลรวมของประเภท animal ซึ่งจะเท่ากับ 4

1. **ให้นักเรียนโหลดข้อมูล zoo.csv และเก็บค่าไว้ในตัวแปรที่ชื่อว่า zoo และแสดงผลจำนวน 6 แถว**

หากต้องการนับจำนวนสามารถใช้ฟังก์ชัน .count() ซึ่งเป็นการหาความถี่ในการมีอยู่ของข้อมูลใน dataframe เช่น zoo.count() ผลที่ได้คือ ใน column animal, uniq\_id และ water\_need มีข้อมูลปรากฏอยู่ทั้งหมด 22 ค่า



1. **ให้นักเรียนแสดงค่า count เฉพาะ column animal เท่านั้น (ให้ดูจาก worksheet ที่แล้วสำหรับ syntax ในการเฉพาะเจาะจงไปที่ column ใด column หนึ่ง)**
2. **ให้นักเรียนแสดงค่า count สำหรับ column animal และ water\_need**

ในกรณีที่เราต้องการหาผลรวม (Summation) เราสามารถใช้ฟังก์ชัน .sum()

1. **ให้นักเรียนหาผลรวมของ water\_need**
2. **ให้นักเรียนหาผลรวมของ animal (สังเกตข้อแตกต่างของผลที่ได้จากข้อนี้กับข้อก่อนหน้านี้ บันทึกลงใน Jupyter notebook ด้วย)**

เราสามารถหาค่า max min โดยใช้ฟังก์ชัน .max() และ .min() ตามลำดับ

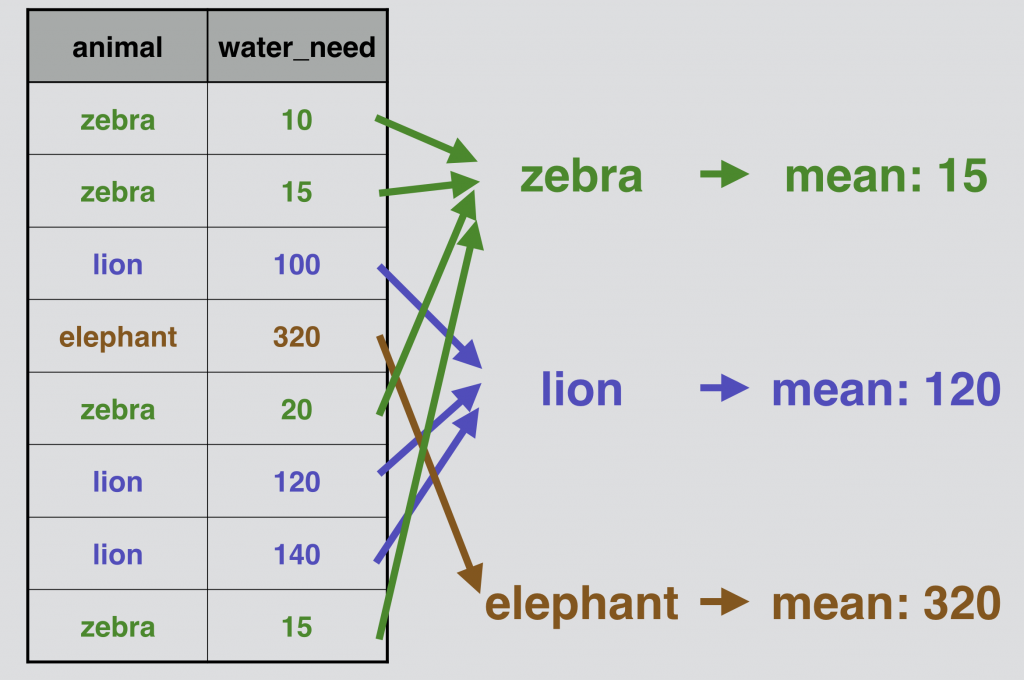
1. **จงหาค่า max และ min ของ water\_need**

**ให้นักเรียนอธิบายความแตกต่างของค่า มัธยฐาน (Median) และค่าเฉลี่ยเลขคณิต (Mean หรือ เป็นทางการคือ Arithmetic Mean) อาจารย์จะสุ่มถามนะครับ**

ใน Pandas การหาค่า Mean และ Median ก็จะมีฟังก์ชัน .mean() และ .median() ตามลำดับ

1. **จงหาค่า mean และ median ของ water\_need**

สำหรับการจัดกลุ่มข้อมูล (Grouping) ถือว่ามีความสำคัญมากในการวิเคราะห์ข้อมูล เช่น เราสามารถหาค่า Mean ของ water\_need (**หากนักเรียนทำถูกต้องจะได้ค่า Mean = 347.727273**) สำหรับ animal ทั้งหมด แต่ในการวิเคราะห์ที่ลงลึกไป เราจำเป็นต้องรู้ค่า Mean ของ water\_need สำหรับสัตว์แต่ละประเภท เช่น



จากตัวอย่างข้างบนจะเห็นได้ว่าหากวิเคราะห์ลงลึกไป ค่า Mean ของ water\_need แยกตามประเภทสัตว์ จะให้ข้อมูลที่ละเอียดมากขึ้น ข้อมูลเหล่านี้สามารถนำไปวางแผนในการจัดสรรน้ำต่อไป

มาดูกันว่าการจัดกลุ่มข้อมูลใน Pandas ด้วยฟังก์ชัน .groupby( ) ทำได้อย่างไร ให้นักเรียนลองคำสั่งต่อไปนี้

zoo.groupby('animal').mean()

ผลที่ได้คือ ค่า Mean ของแต่ละ column ที่จัดกลุ่มโดย animal



ค่า uniq\_id เป็น id ประจำตัวของสัตว์ต่างๆ ซึ่งจะดูแปลกหาเรามาหาค่า Mean ให้ uniq\_id

1. **ให้นักเรียนแสดงผลค่า Mean เฉพาะ water\_need เท่านั้น**

ให้นักเรียนโหลดข้อมูลไว้ที่ article\_read จากไฟล์ pandas\_tutorial\_read.csv

1. **ให้นักเรียนหาว่า source ใดมีจำนวนคนอ่านมากที่สุด (hint: ต้องทำการ .groupby('source') แล้วนับค่า)**
2. **ต่อจากข้อ 9 ให้นักเรียนแสดงผลเฉพาะจำนวน user\_id**
3. **ให้นักเรียนแสดงผลเฉพาะข้อมูลที่มากจาก country\_2 ว่า source ใด ที่มีค่าทั้ง topic และ source (รวมกัน) มีจำนวนมากที่สุด (hint: อันดับแรกต้องทำการ filter เอาเฉพาะ country\_2 และ ทำการ .groupby โดยกำหนดทั้ง topic และ source และนับค่า)**

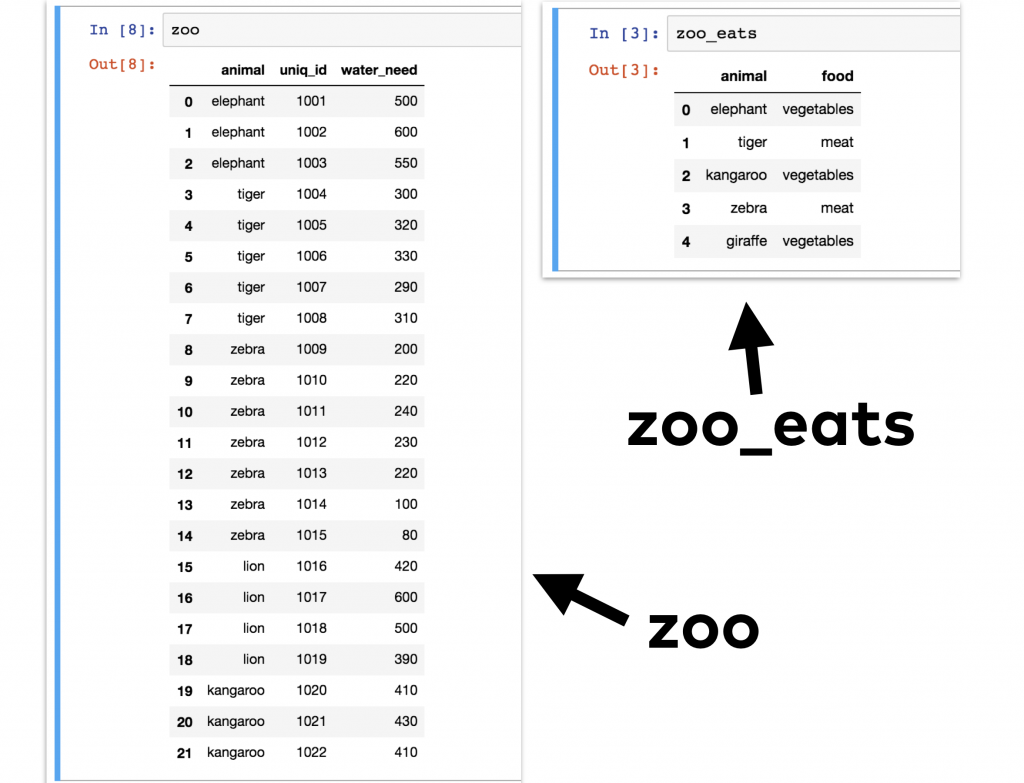
**Pandas**

**การทำ Data Merging**

ในการทำ Big Data Project ในบางกรณีเราจะไม่ทำการเก็บข้อมูลทั้งหมดไว้ใน table เดียว แต่จำทำการเก็บข้อมูลแยกเป็น table ย่อยๆ เพื่อให้การบริหารจัดการข้อมูลสามารถทำได้สะดวกและง่ายขึ้น (กว่าการจัดการข้อมูลขนาดใหญ่) นอกจากนี้ยังมีข้อดีอื่นๆ เช่น ประหยัดพื้นที่จัดเก็บ การเรียกดูข้อมูลจากตารางเล็กๆ สามารถทำได้เร็วกว่า

ดังนั้นระหว่างการวิเคราะห์ข้อมูล ต้องมีการดึง (หรือโหลด) ข้อมูลมาจากหลายๆ table ซึ่งวิธีการดังกล่าวต้องใช้เทคนิคที่เรียกว่า Data Merging

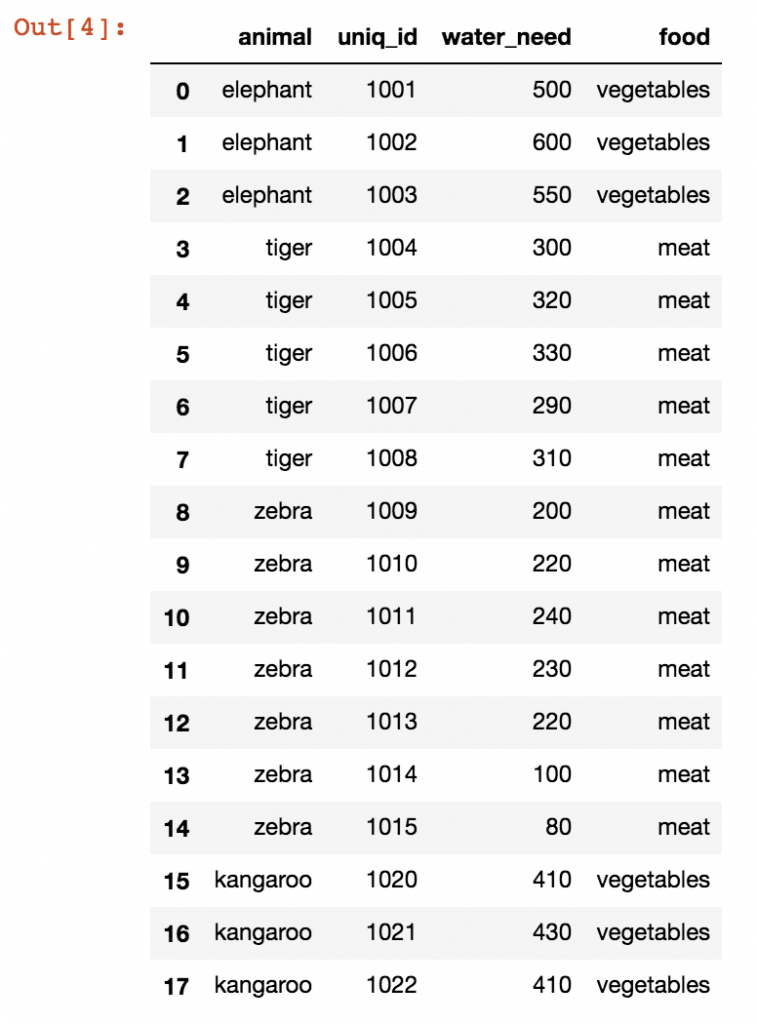
ให้นักเรียนโหลดข้อมูลจาก zoo.csv และสร้างข้อมูลชุดใหม่มีชื่อว่า zoo\_eats.csv และทำการโหลดไว้ที่ zoo\_eats แสดงดังภาพด้านล่าง (ดู worksheet 1 สำหรับการสร้างไฟล์ .csv ผ่าน Jupyter notebook)



จากข้อมูลทั้งสองตารางเราจะทำการ Merge ข้อมูลเข้าด้วยกัน เพื่อให้ได้ผลลัพธ์ต่อไปนี้ (หากนักเรียนสังเกตุให้ดี การเก็บข้อมูลทั้งหมดไว้ในตารางเดียวจะกินพื้นที่มากกว่า เพราะทุกๆ แถวจะต้องมีข้อมูล food จัดเก็บซึ่งต้องใช้พื้นที่ records ตาม ID มีอยู่ 21 records ถ้ามี 100,000 records ก็ต้องเก็บให้ครบ แต่การเก็บแยกจะใช้พื้นที่อีกไม่มากในตาราง zoo\_eats ใช้พื้นที่จัดเก็บข้อมูลไม่มาก )

วิธีการ Merge ข้อมูลใน Pandas สามารถทำได้อย่างง่ายด้วยฟังก์ชัน .merge( )

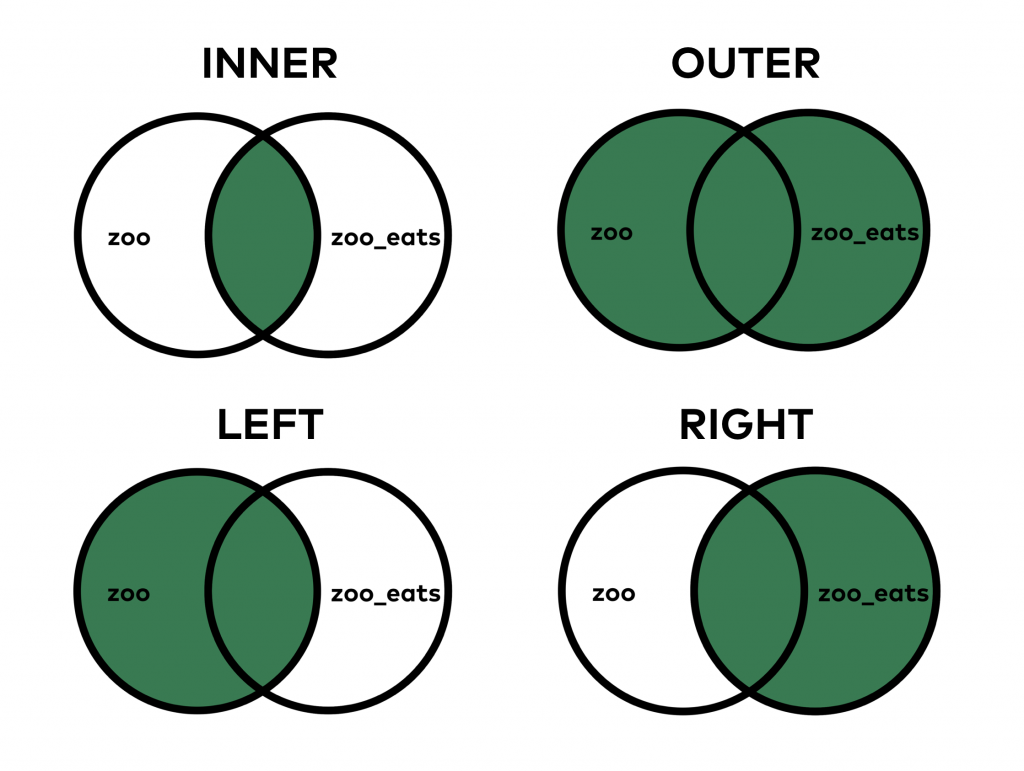
ให้นักเรียนเขียนคำสั่ง zoo.merge(zoo\_eats) คำสั่งนี้จะเป็นการ merge ข้อมูลจาก zoo\_eats (ทำการ map อาหารที่สัตว์แต่ละประเภทกิน) เข้าไปยัง zoo



แต่เดี๋ยวก่อนทำไมข้อมูลเหลืออยู่แค่ 17 จาก 21 records ข้อมูลของ Lion ทั้ง 4 records หากนักเรียนนั่งคิดตอนนี้จะปวดหัว เดี๋ยวเราจะกลับมาเรื่องนี้กัน

1. **ให้นักเรียนลองคำสั่ง zoo\_eats.merge(zoo) เพื่อดูความแตกต่างกันกับการใช้ zoo.merge(zoo\_eats)**

ใน Pandas รองรับการ Merge ข้อมูล 4 รูปแบบ แสดงดังรูปด้านล่าง



* INNER - แบบค่า Default ในการใช้ฟังก์ชัน merging ซึ่งเป็นการ merge ค่าในตารางที่มีรวมกัน (เป็นสาเหตุว่าทำไม lion จึงไม่ปรากฎในตารางเมื่อใช้คำสั่ง zoo.merge(zoo\_eats))
* OUTER – เป็นการ merge ทั้งหมดโดยไม่สนว่าแต่ละตารางจะมีค่าเหมือน หรือไม่เหมือนกัน
* LEFT - เป็นการ merge แบบอิงตารางซ้ายเป็นหลัก
* RIGHT - เป็นการ merge แบบอิงตารางขวาเป็นหลัก

การจะกำหนดว่าใช้ merge แบบใด ต้องเพิ่ม syntax ‘how = ’ ไว้ในคำสั่ง

1. **ให้นักเรียนลอง** 
   1. **zoo.merge(zoo\_eats, how = 'outer')**
   2. **zoo.merge(zoo\_eats, how = 'left')**
   3. **zoo.merge(zoo\_eats, how = 'right')**

**ในความคิดเห็นนักเรียนคำสั่งไหนเหมาะสมที่สุดสำหรับการ merge zoo และ zoo.eats เข้าด้วยกัน**