

รายงาน

เรื่อง ความสัมพันธ์ระหว่างมือข้างที่ถนัดและสมองซีกที่ถนัด

โดย

นางสาวธนพร ป้อมสถิตย์ รหัสนักศึกษา 64070505206

นายธีรภัทร ชำพาลี รหัสนักศึกษา 64070505231

อาจารย์ผู้สอน

อาจารย์อัญชลิสา แท้ตระกูล

รายงานนี้เป็นส่วนหนึ่งของรายวิชา CPE102

ภาคเรียนที่ 1 ปีการศึกษา 2564

สารบัญ

สารบัญ	ก
สารบัญภาพ	ข
บทนำ	1
ที่มาและความสำคัญ	1
วัตถุประสงค์	1
ขอบเขตของการศึกษา	1
ข้อมูลและวิธีการเก็บข้อมูล	2
วิธีการคำนวณ	4
Contingency table	4
Cramer's V	5
Goodman- Kruskal's Lambda	7
สรุปผล	10
อธิบายโปรแกรมคำนวณ	12
เอกสารอ้างอิง	19

สารบัญภาพ

ภาพที่ 1 ตัวอย่างการแสดงผลจากการทำแบบทดสอบความถนัดของสมอง	2
ภาพที่ 2 ตารางแจกแจงความถี่แบ่งตามเพศ มือที่ถนัดและสมองซีกที่ถนัดจากการทดสอบ	3
ภาพที่ 3 ตารางแจกแจงความถี่แบ่งตามเพศ มือที่ถนัดและสมองซีกที่ถนัดจากการประเมินตัวเอง	3
ภาพที่ 4 ตารางแจกแจงความถี่แบ่งตามสมองที่ถนัดและเรื่องที่เด่น	3
ภาพที่ 5 ตารางแจกแจงความถี่แบ่งตามมือข้างที่ถนัดและสมอง แบบมีการนับรวม	4
ภาพที่ 6 ตารางแจกแจงความถี่แบ่งตามมือข้างที่ถนัดและสมอง แบบไม่มีการนับรวม	4
ภาพที่ 7 ตารางตีความระดับความสัมพันธ์จากค่า Cramer's V และ degree of freedom	6
ภาพที่ 8 ตารางแจกแจงความถี่ในการทำนายสมองด้วยมือแบบมีผลรวม	8
ภาพที่ 9 ตารางแจกแจงความถี่ในการทำนายสมองด้วยมือแบบไม่มีผลรวม	8
ภาพที่ 10 ตารางแจกแจงความถี่ในการทำนายมือด้วยสมองแบบมีผลรวม	9
ภาพที่ 11 ตารางแจกแจงความถี่ในการทำนายมือด้วยสมองแบบไม่มีผลรวม	9
ภาพที่ 12 แผนภูมิแท่งแสดงจำนวนคนที่ถนัดสมองซีกซ้ายและขวาแบ่งตามความถนัดของมือ	10
ภาพที่ 13 Swarm plotแสดงการกระจายตัวของเปอร์เซ็นต์ความถนัดสมอง	10
ภาพที่ 14 แผนภูมิแท่งแสดงจำนวนคนที่ถนัดสมองซีกซ้ายขวา แบ่งตามความถูกต้องของการประเมินตัวเอง	11
ภาพที่ 15 แผนภูมิมวงกลมแบ่งเรื่องที่ถนัดตามสมองซีกที่ถนัด	11

บทนำ

ที่มาและความสำคัญ

จากความคิดที่ว่าการถนัดซ้ายเป็นเรื่องที่หลายคนคิดว่าเป็นเรื่องที่ไม่ดี เพราะจะทำให้การใช้ชีวิตประจำวันลำบากขึ้น นักวิทยาศาสตร์และนักจิตวิทยาจึงได้พยายามหาเหตุผลว่าอะไรคือสาเหตุที่ทำให้คนบางคนถนัดมือซ้าย รวมถึงหาสาเหตุที่ทำให้คนบางคนใช้ตาซ้ายในการเห็นได้ดีกว่าตาขวา และบางคนมีหูซ้ายที่สามารถได้ยินชัดกว่าหูขวา ส่วนความเชื่อที่ว่าคนที่ถนัดซ้ายมักเป็นอัจฉริยะมากกว่าคนที่ถนัดขวานั้นเป็นความจริงหรือไม่ และยังมีคำถามอื่น ๆ ที่เกี่ยวกับความถนัดซ้าย-ขวามาก เพราะนักวิทยาศาสตร์ไม่เชื่อว่า ธรรมชาติจะจงใจสร้างคนให้มีความถนัดที่เหลื่อมล้ำกัน และคิดว่าการมีความเสมอภาคกันในธรรมชาติ น่าจะทำให้คนเรามีความถนัดซ้ายหรือขวาในจำนวนที่มากพอ ๆ กัน ด้วยเหตุนี้เองจึงเป็นที่มาในการสำรวจและศึกษาเพื่อหาความสัมพันธ์ระหว่างการทำงานของสมองและความถนัดของมือมนุษย์ ทั้งนี้เพื่อเป็นประโยชน์ในการเข้าใจธรรมชาติของมนุษย์มากขึ้น

วัตถุประสงค์

1. เพื่อศึกษาความสัมพันธ์ระหว่างความถนัดของมือและสมอง
2. เพื่อให้แต่ละคนเข้าใจตัวเองมากขึ้น
3. เพื่อฝึกทักษะในการทำงานกับข้อมูลในอนาคต

ขอบเขตของการศึกษา

ในการศึกษาเรื่องมือที่ถนัดและสมองซีกที่ถนัดนั้น มือข้างที่ถนัด หมายถึง มือข้างที่ใช้ในการเขียนหนังสือเป็นประจำ ส่วนความถนัดของสมอง หมายถึง เป็นเปอร์เซ็นต์ที่ได้จากการทำแบบทดสอบจากเว็บไซต์ <https://arealme.com/left-right-brain/th/> หากมีเปอร์เซ็นต์ข้างใดมากกว่าจะสรุปว่ามีความถนัดในด้านนั้นมากกว่า การศึกษาในครั้งนี้เป็นเพียงการหาความสัมพันธ์ในกลุ่มตัวอย่างขนาดเล็กเท่านั้น ไม่สามารถเป็นนำไปเป็นบทสรุปขอความสัมพันธ์ที่มีระหว่างมือที่ถนัดและสมองได้

ข้อมูลและวิธีการเก็บข้อมูล

การเก็บข้อมูลทำโดย สร้างแบบสอบถามผ่าน Google Form การตอบคำถามส่วนใหญ่จะเป็นลักษณะเลือกได้เพียงหนึ่งตัวเลือก ทั้งนี้เพื่อป้องกันความผิดพลาดเนื่องจากการพิมพ์ผิด มีเพียงข้อมูลบางชนิดที่ผู้ตอบคำถามจำเป็นต้องกรอกด้วยตัวเอง ได้แก่ รหัสนักศึกษา เพอร์เซ็นต์ความถนัดสมอง เรื่องที่ถนัด

แบ่งตามจุดประสงค์การนำไปใช้ ได้เป็น 3 ประเภท

- 1) ข้อมูลส่วนตัว : รหัสนักศึกษา เพศ มือข้างที่ถนัด
- 2) ข้อมูลจากการประเมินตัวเอง : สมองข้างที่ถนัดจากการประเมินตัวเอง
- 3) ข้อมูลที่ได้จากการทดสอบ : เพอร์เซ็นต์ความถนัดสมองซีกซ้ายและซีกขวา เรื่องที่ถนัด เช่น ศิลปะ การวิเคราะห์ การใช้เหตุผล

แบ่งตามชนิดของข้อมูล ได้เป็น 2 ประเภท

- 1) ข้อมูลนามบัญญัติ (Nominal scale) จำนวน 5 ข้อมูล ประกอบด้วย รหัสนักศึกษา เพศ มือข้างที่ถนัด สมองข้างที่ถนัดจากการประเมินตัวเอง และเรื่องที่ถนัด
- 2) ข้อมูลเชิงลำดับ (Ordinal scale) จำนวน 2 ข้อมูล ประกอบด้วย เพอร์เซ็นต์ความถนัดสมองซีกซ้ายและซีกขวา

ส่วนของการทำแบบทดสอบความถนัดของสมองจะใช้แบบทดสอบจากเว็บไซต์ <https://arealme.com/left-right-brain/th/> ภายในประกอบไปด้วยคำถามเชิงจิตวิทยาจำนวน 31 ข้อ เมื่อผู้ทดสอบตอบคำถามครบทุกข้อ จะปรากฏหน้าประมวลผลให้ ดังภาพด้านล่าง สำหรับข้อมูลที่ต้องนำมาตอบใน google form ได้แก่ เพอร์เซ็นต์ความถนัดสมองซีกซ้าย เพอร์เซ็นต์ความถนัดสมองซีกขวา และเรื่องที่ถนัด



ภาพที่ 1 ตัวอย่างการแสดงผลจากการทำแบบทดสอบความถนัดของสมอง

ตารางแจกแจงความถี่จากคำตอบของผู้ทำแบบสอบถามจำนวน 50 คน

		Expected_brain	Left brain	Right brain	All
Gender	Hand				
Female	Left hand		2	4	6
	Right hand		15	14	29
Male	Left hand		1	1	2
	Right hand		8	5	13
All			26	24	50

ภาพที่ 2 ตารางแจกแจงความถี่แบ่งตามเพศ มือที่ถนัดและสมองซีกที่ถนัดจากการทดสอบ

		Realbrain	Left brain	Right brain	All
Gender	Hand				
Female	Left hand		3	3	6
	Right hand		20	9	29
Male	Left hand		0	2	2
	Right hand		10	3	13
All			33	17	50

ภาพที่ 3 ตารางแจกแจงความถี่แบ่งตามเพศ มือที่ถนัดและสมองซีกที่ถนัดจากการประเมินตัวเอง

Personality	Analysis	Art	Fairness	Imagination	Instinct	Mathematics	Memory	Reasoning	Symbolic	Tidiness	All
Realbrain											
Left brain	4	0	6	0	0	3	0	18	1	1	33
Right brain	0	7	0	3	1	0	6	0	0	0	17
All	4	7	6	3	1	3	6	18	1	1	50

ภาพที่ 4 ตารางแจกแจงความถี่แบ่งตามสมองที่ถนัดและเรื่องที่เด่น

วิธีการคำนวณ

Contingency table

เป็นการสรุปข้อมูลในรูปตารางแจกแจงความถี่แบบสองทาง โดยใช้ข้อมูลจากคอลัมน์ที่สนใจจำนวน 2 คอลัมน์ สำหรับ ส่วนของโปรแกรมที่เกี่ยวกับการสร้าง contingency table จะใช้ pandas.crosstab() ซึ่งภายในประกอบไปด้วย 2 ฟังก์ชัน ได้แก่

- 1) ฟังก์ชันสร้าง contingency table ที่มีผลรวมในแต่ละแถวและคอลัมน์ : สามารถนำไปใช้ประโยชน์ต่อการคำนวณ chi-square และ Goodman – Kruskal’s lambda
- 2) ฟังก์ชันสร้าง contingency table ที่ไม่มีผลรวมในแต่ละแถวและคอลัมน์ : สามารถนำไปใช้ประโยชน์ต่อการคำนวณ Goodman – Kruskal’s lambda

ตัวอย่าง ผลลัพธ์จากการสร้าง Contingency Table โดยตัวแปรที่สนใจได้แก่ มือข้างที่ถนัด และสมองที่ถนัดซึ่งได้จากการทำแบบทดสอบ

Realbrain	Left brain	Right brain	All
Hand			
Left hand	3	5	8
Right hand	30	12	42
All	33	17	50

ภาพที่ 5 ตารางแจกแจงความถี่แบ่งตามมือข้างที่ถนัดและสมอง แบบมีการนับรวม

Realbrain	Left brain	Right brain
Hand		
Left hand	3	5
Right hand	30	12

ภาพที่ 6 ตารางแจกแจงความถี่แบ่งตามมือข้างที่ถนัดและสมอง แบบไม่มีการนับรวม

Cramer's V

การหาความสัมพันธ์ของข้อมูลระหว่างข้อมูลนามบัญญัติซึ่งเป็นอิสระต่อกันว่ามีความสัมพันธ์กันในระดับใด โดยมีสูตรการคำนวณดังนี้

$$V = \sqrt{\frac{X^2/n}{\min(c-1, r-1)}}$$

V หมายถึง ค่า Cramer's V

X² หมายถึง ค่า Chi-square

c หมายถึง จำนวนคอลัมน์ของ contingency table แบบไม่มี total

r หมายถึง จำนวนแถวของ contingency table แบบไม่มี total

min หมายถึง การหาจำนวนที่น้อยที่สุดในช่วง

การคำนวณหา Chi-square ทำได้โดย

$$X^2 = \sum_{i=1}^n \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

O_{ij} หมายถึง ค่าจริงที่ได้จากการทดสอบ

E_{ij} หมายถึง ค่าคาดหวัง

เมื่อคำนวณค่า Cramer's V ได้แล้ว ขั้นตอนต่อไปจะเป็นการตีความระดับความสัมพันธ์ของสองตัวแปรโดยจะเทียบความสัมพันธ์ในตาราง โดย degree of freedom คำนวณได้จาก $\min(c-1, r-1)$

Degrees of freedom	Small	Medium	Large
1	0.10	0.30	0.50
2	0.07	0.21	0.35
3	0.06	0.17	0.29
4	0.05	0.15	0.25
5	0.04	0.13	0.22

ภาพที่ 7 ตารางตีความระดับความสัมพันธ์จากค่า Cramer's V และ degree of freedom

จากการการคำนวณหาความสัมพันธ์ระหว่างมือข้างที่ถนัดและสมองซีกที่ถนัด ด้วย Cramer's v จะได้ผลลัพธ์อยู่ที่ 0.26 โดยมีค่า degree of freedom เท่ากับ 1 เมื่อเทียบในตารางตีความแล้วจะได้ว่า มือข้างที่ถนัดและสมองซีกที่ถนัดจะมีความสัมพันธ์กันในระดับต่ำ

Goodman- Kruskal's Lambda

เป็นการคำนวณความผิดพลาดที่ลดลงไปอันเนื่องมาจากการทราบข้อมูลจากตัวแปรเพิ่มขึ้น เปรียบเทียบกับความผิดพลาดที่เกิดจากการทำนายจากผลรวมอย่างเดียวโดยไม่ทราบข้อมูลเกี่ยวกับตัวแปรเลย ซึ่งเป็นรูปหนึ่งของการวัดความสัมพันธ์ระหว่างตัวแปร Proportional Reduction in Error (PRE) การที่ความผิดพลาดในการทำนายลดลง แสดงว่าตัวแปรตัวหนึ่งอธิบายตัวแปรอีกตัวแปรอีกตัวหนึ่งได้ ตัวแปรทั้งสองจึงมีความสัมพันธ์กัน โดยมีสูตรการคำนวณดังนี้

$$\lambda_{B|A} = \frac{P_E - P_{E|A}}{P_E}$$

$\lambda_{B|A}$ หมายถึง ความน่าจะเป็นของความผิดพลาดที่ลดลงในการทำนายตัวแปร B โดยใช้ข้อมูลจากตัวแปร A
 P_E หมายถึง ความน่าจะเป็นของความผิดพลาดในการทำนายตัวแปร B โดยไม่รู้ข้อมูลของตัวแปร A
 $P_{E|A}$ หมายถึง ความน่าจะเป็นของความผิดพลาดในการทำนายตัวแปร B โดยรู้ข้อมูลของตัวแปร A

สำหรับการคำนวณ P_E และ $P_{E|A}$ จำเป็นต้องใช้ข้อมูลจาก contingency table หากต้องการจะทำนายตัวแปรใดให้ นำตัวแปรนั้นไว้เป็นคอลัมน์ และตัวแปรที่เป็นตัวพยากรณ์ให้นำตัวแปรนั้นไว้เป็นแถว โดยมีสูตรการคำนวณดังนี้

$$P_E = 1 - \frac{\max_{col}}{N}$$

P_E หมายถึง ความน่าจะเป็นของความผิดพลาดในการทำนายตัวแปร B โดยไม่รู้ข้อมูลของตัวแปร A
 \max_{col} หมายถึง ค่าผลรวมคอลัมน์ที่มีค่ามากที่สุด โดยไม่นับรวมคอลัมน์สุดท้าย
 N หมายถึง ผลรวมของทุกช่องในตาราง

*หมายเหตุ: การคำนวณ P_E ใช้ contingency table ชนิดมีการนับรวมแต่ละคอลัมน์/แถว

$$P_{E|A} = 1 - \frac{\Sigma \max_{row}}{N}$$

$P_{E|A}$ หมายถึง ความน่าจะเป็นของความผิดพลาดในการทำนายตัวแปร B โดยรู้ข้อมูลของตัวแปร A
 $\Sigma \max_{row}$ หมายถึง ผลรวมของค่ามากที่สุดในแต่ละแถว
 N หมายถึง ผลรวมของทุกช่องในตาราง

*หมายเหตุ: การคำนวณ $P_{E|A}$ ใช้ contingency table ชนิดไม่มีการนับรวมแต่ละคอลัมน์/แถว

ตัวอย่างการคำนวณ ความน่าจะเป็นของความผิดพลาดที่ลดลงในการทำนายสมองซีกที่ถนัด โดยใช้ข้อมูลจากมือข้างที่ถนัด

Realbrain	Left brain	Right brain	All
Hand			
Left hand	3	5	8
Right hand	30	12	42
All	33	17	50

$$P_{\text{brain}} = 1 - \frac{33}{50}$$
$$P_{\text{brain}} = 0.34$$

ภาพที่ 8 ตารางแจกแจงความถี่ในการทำนายสมองด้วยมือแบบมีผลรวม

Realbrain	Left brain	Right brain
Hand		
Left hand	3	5
Right hand	30	12

$$P_{\text{brain} | \text{hand}} = 1 - \frac{5 + 30}{50}$$
$$P_{\text{brain} | \text{hand}} = 0.3$$

ภาพที่ 9 ตารางแจกแจงความถี่ในการทำนายสมองด้วยมือแบบไม่มีผลรวม

$$\lambda_{\text{brain} | \text{hand}} = \frac{P_{\text{brain}} - P_{\text{brain} | \text{hand}}}{P_{\text{brain}}}$$
$$\lambda_{\text{brain} | \text{hand}} = \frac{0.34 - 0.3}{0.34}$$
$$\lambda_{\text{brain} | \text{hand}} = 0.117$$

จากการคำนวณได้ $\lambda_{\text{brain} | \text{hand}} = 0.117$ หมายความว่า หากต้องการทำสมองซีกที่ถนัด โดยใช้ข้อมูลของมือข้างที่ถนัดเป็นตัวพยากรณ์ จะสามารถลดความผิดพลาดในการทำนายลงไปได้ 11.7 %

ตัวอย่างการคำนวณ ความน่าจะเป็นของความผิดพลาดที่ลดลงในการทำนายมือข้างที่ถนัด โดยใช้ข้อมูลจากสมองซีกที่ถนัด

Hand	Left hand	Right hand	All
Realbrain			
Left brain	3	30	33
Right brain	5	12	17
All	8	42	50

$$P_{hand} = 1 - \frac{42}{50}$$

$$P_{hand} = 0.16$$

ภาพที่ 10 ตารางแจกแจงความถี่ในการทำนายมือด้วยสมองแบบมีผลรวม

Realbrain		
Left brain	3	30
Right brain	5	12

$$P_{hand | brain} = 1 - \frac{30 + 12}{50}$$

$$P_{hand | brain} = 0.16$$

ภาพที่ 11 ตารางแจกแจงความถี่ในการทำนายมือด้วยสมองแบบไม่มีผลรวม

$$\lambda_{hand|brain} = \frac{P_{hand} - P_{hand|brain}}{P_{hand}}$$

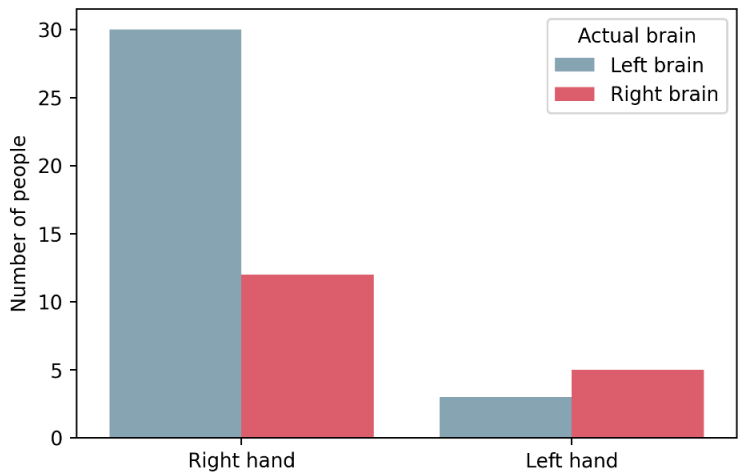
$$\lambda_{brain|hand} = \frac{0.16 - 0.16}{0.16}$$

$$\lambda_{brain|hand} = 0.00$$

จากการคำนวณได้ $\lambda_{brain|hand} = 0.00$ หมายความว่า หากต้องการทำสมองซีกที่ถนัดโดยใช้ข้อมูลของมือข้างที่ถนัดเป็นตัวพยากรณ์ จะสามารถลดความผิดพลาดในการทำนายลงไปได้ 0.00%

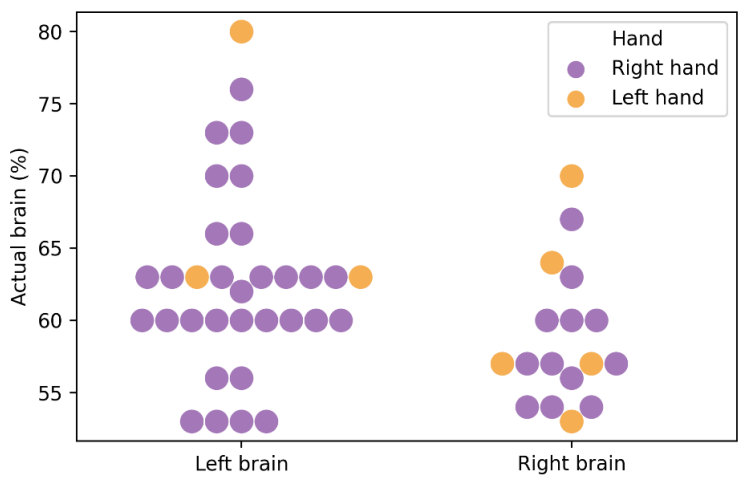
สรุปผลจากการคำนวณ Goodman-Kruskal's lambda ทั้ง 2 ตัวอย่าง พบว่า การทำนายสมองซีกที่ถนัดจากข้อมูลมือข้างที่ถนัดนั้น มีความน่าจะเป็นที่จะทายถูกได้มากกว่า การทำนายมือข้างที่ถนัดจากข้อมูลสมองซีกที่ถนัด

สรุปผล



ภาพที่ 12 แผนภูมิแท่งแสดงจำนวนคนที่ถนัดสมองซีกซ้ายและขวาแบ่งตามความถนัดของมือ

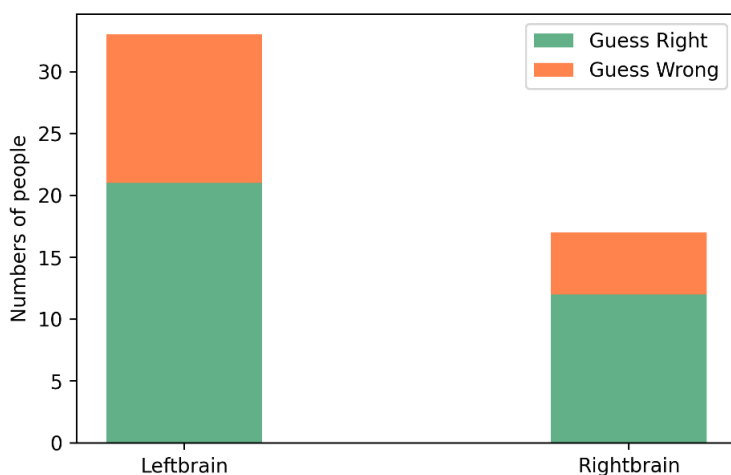
จากภาพที่ 12 แสดงให้เห็นว่าคนที่ถนัดมือขวาส่วนมากจะเป็นคนที่ถนัดสมองซีกซ้ายคิดเป็นจำนวน 30 คน (71.43% ของคนที่ถนัดมือขวา) และในทางกลับกันคนที่ถนัดมือซ้ายส่วนมากจะมีความถนัดสมองในซีกขวา คิดเป็นจำนวน 5 คน (62.5%ของคนที่ถนัดมือซ้าย)



ภาพที่ 13 Swarm plotแสดงการกระจายตัวของเปอร์เซ็นต์ความถนัดสมอง

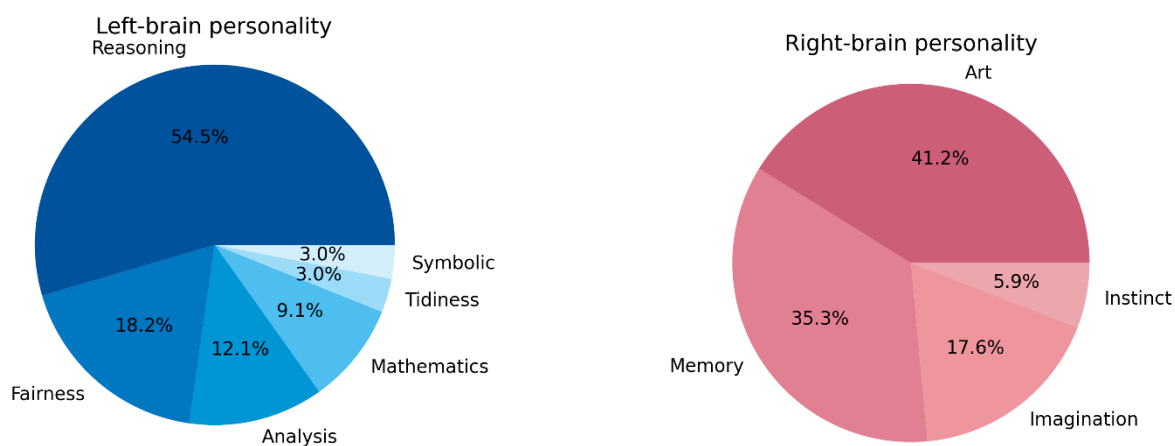
จากภาพที่ 13 จุดแต่ละจุดแสดงถึงข้อมูลของแต่ละคน โดยแบ่งความถนัดของมือด้วยสี จะเห็นได้ว่าคนที่ถนัดสมองซีกซ้าย ส่วนมากจะมีเปอร์เซ็นต์ความถนัดอยู่ในช่วง 60-65 %

สำหรับคนที่ถนัดสมองซีกขวา ส่วนมากจะมีเปอร์เซ็นต์ความถนัดอยู่ในช่วง 50-60 % และหากสนใจที่คนถนัดมือซ้ายและถนัดสมองซีกขวาจะมีช่วงเปอร์เซ็นต์ที่กระจายตัวกันมากกว่า เมื่อเทียบกับคนถนัดมือซ้ายและสมองซีกซ้าย ซึ่งจะมีเปอร์เซ็นต์ความถนัดที่ค่อนข้างสูง



ภาพที่ 14 แผนภูมิแท่งแสดงจำนวนคนที่ถนัดสมองซีกซ้ายขวา แบ่งตามความถูกต้องของการประเมินตัวเอง

จากภาพที่ 14 แสดงให้เห็นว่าคนส่วนมากคิดเป็น 66% ประเมินตัวเองถูกต้องว่าถนัดสมองซีกใด ซึ่งคนที่ถนัดสมองซีกซ้ายมีอัตราส่วนในการประเมินถูกน้อยกว่าคนที่ถนัดสมองซีกขวา



ภาพที่ 15 แผนภูมิมวงกลมแบ่งเรื่องที่ถนัดตามสมองซีกที่ถนัด

ภาพที่ 15 เป็นการจัดกลุ่มตามความถนัดของสมองในเรื่องต่างๆ โดยคนที่ถนัดสมองซีกซ้าย จะถนัดในเรื่องการใช้เหตุผล ความเป็นกลางและการวิเคราะห์ ตามลำดับ สำหรับคนที่ถนัดสมองซีกขวา จะถนัดในเรื่องศิลปะและความจำ

อธิบายโปรแกรมคำนวณ

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import math
6 from google.colab import files
7
8 %matplotlib inline
```

ติดตั้งไลบรารี ที่จำเป็นในการเขียนโปรแกรม เช่น pandas numpy seaborn

```
1 # read file csv
2 df = pd.read_csv("/content/Data_raw_final.csv")
3 df.head(3)
```

อ่านไฟล์ในรูปแบบ csv และกำหนดให้เป็นตัวแปรที่ชื่อ df

```
1 # 5206 Kaew
2 # change column's name
3 change_col = {'รหัสนักศึกษา มจร. (หากไม่มีให้เว้นไว้)': 'Student_ID',
4               'เพศ': 'Gender',
5               'ถนัดมือข้างไหน': 'Hand',
6               'คิดว่าตัวเองถนัดใช้สมองซีกไหน': 'Expected_brain',
7               'สมองLeft brain (กรอกเฉพาะตัวเลข)': 'Left_brain',
8               'สมองRight brain (กรอกเฉพาะตัวเลข)': 'Right_brain',
9               'ถนัดเรื่องไหน (นำคำตอบมาจากกรอกหมายเลข 1)': 'Personality'
10              }
11 df.rename(columns=change_col, inplace=True)
12 df.head(2)
```

เปลี่ยนชื่อแต่ละคอลัมน์ให้ เพื่อให้เรียกใช้งานได้สะดวกมากขึ้น โดยลบออกจาก df เลย

```
1 # 5206 Kaew
2 # delete columns
3 del_col = ['Timestamp', 'Student_ID']
4 df.drop(del_col, axis=1, inplace=True)
5 df.head(2)
```

ทำการลบคอลัมน์ Timestamp และ Student_ID ออก เนื่องจากเป็นข้อมูลที่ไม่จำเป็นต่อการวิเคราะห์ผล

```
1 # 5206 Kaew
2 # count total records
3 dataShape = df.shape
4 n = dataShape[0]
5 m = dataShape[1]
6 print("Raw data contains",n,"records.")
7 print("Raw data contains",m,"columns.")
```

ดูขนาดของข้อมูลและแสดงผลออกมาเป็นจำนวนแถวและคอลัมน์

```

1 # 5231 Floyd
2 # Realbrain function
3 def realbrain(row):
4     if row['Left_brain'] > 50:
5         return 'Left brain'
6     elif row['Left_brain'] == 50:
7         return 'Both'
8     else :
9         return 'Right brain'

```

```

1 # 5231 Floyd
2 # apply realbrain function for create Realbrain column
3 df['Realbrain'] = df.apply(realbrain, axis=1)
4 df.head(3)

```

สร้างฟังก์ชัน realbrain เพื่อระบุว่าแต่ละคนมีความถนัดของสมองซีกใด โดยดูจากค่าเปอร์เซ็นต์ความถนัดของสมองซีกซ้าย หากมีค่ามากกว่า 50 แสดงว่าเป็นคนถนัดสมองซีกซ้าย ถ้าน้อยกว่าจะเป็นคนถนัดสมองซีกขวา กรณีเท่ากันจะเป็นคนที่ถนัดทั้งสองข้าง หลังจากนั้นนำฟังก์ชัน realbrain ไปสร้างคอลัมน์ใหม่ที่มีชื่อว่า Realbrain

```

1 # 5231 Floyd
2 # Percentage from real brain function
3 def realbrain_100(row):
4     if row['Realbrain'] == 'Left brain' :
5         return row['Left_brain']
6     if row['Realbrain'] == 'Right brain':
7         return row['Right_brain']

```

```

1 # 5231 Floyd
2 # apply realbrain_100 function for create Realbrain column
3 df['Percent_realbrain'] = df.apply(realbrain_100, axis=1)
4 df.head(3)

```

สร้างฟังก์ชัน realbrain_100 เพื่อดึงข้อมูลเปอร์เซ็นต์ตามความถนัดของสมองที่ได้จากการทดสอบ หากเป็นผู้ที่ถนัดสมองซีกขวาให้ดึงเปอร์เซ็นต์ความถนัดซีกขวามา เช่นเดียวกับซีกซ้าย หลังจากนั้นนำฟังก์ชัน realbrain_100 ไปสร้างคอลัมน์ใหม่ที่มีชื่อว่า Percent_realbrain


```

1 # 5206 Kaew
2 class ctg_table:
3     #raw_data = ""
4     columnA = ""
5     columnB = ""
6
7     def __init__(self, columnA, columnB):
8
9         self.columnA = columnA
10        self.columnB = columnB
11
12    def with_total(self):
13        ctg_with_total = pd.crosstab(self.columnA, self.columnB, margins=True)
14        return ctg_with_total
15
16    def without_total(self):
17        ctg_without_total = pd.crosstab(self.columnA, self.columnB)
18        return ctg_without_total

```

สร้างคลาสสำหรับการสร้าง contingency table โดยภายในประกอบไปด้วย 2 ฟังก์ชัน ได้แก่ ฟังก์ชันสำหรับ contingency table แบบไม่มีผลรวมแต่ละแถว/คอลัมน์ และฟังก์ชันสำหรับ contingency table แบบมีผลรวมแต่ละแถว/คอลัมน์

```

1 # 5206 Kaew
2 # display contingency table
3 # 1.Contingency table with total column
4 print("Contingency table of Hand and brain with total.")
5 print()
6 test1 = ctg_table(df["Hand"] , df["Realbrain"])
7 display(test1.with_total())
8 test1_ctg_total = test1.with_total()
9 print("*****")
10 # 2.Contingency table without total column
11 print("Contingency table of Hand and brain without total.")
12 print()
13 display(test1.without_total())
14 test1_ctg_nototal = test1.without_total()

```

ทำการเรียกใช้คลาสในการสร้าง contingency table โดยใช้ค่าจากมือข้างที่ถนัดและสมองข้างที่ถนัดเป็นแถวและคอลัมน์

Realbrain	Left brain	Right brain	All	Realbrain	Left brain	Right brain
Hand				Hand		
Left hand	3	5	8	Left hand	3	5
Right hand	30	12	42	Right hand	30	12
All	33	17	50			

ตัวอย่าง contingency table ระหว่างมือข้างที่ถนัดและสมองข้างที่ถนัด

```

1 # 5206 Kaew
2 # calculate chi-square(x^2)
3 def chi2_manual(ctg_withtotal):
4     # Expected values = (maxrow*maxcol)/total
5     #-----Expected Values-----#
6     exp_ctg = ctg_withtotal.copy()
7     exp_ctg.iloc[:, :] = np.multiply.outer(exp_ctg.sum(axis=1).values, exp_ctg.sum(axis=0).values) / exp_ctg.sum().sum()
8     print('---Expected (E)---')
9     display(exp_ctg)
10
11
12     # square distance = ((observe-expected)^2)/expected
13     # calculate chi-square values
14     chi2 = ((ctg_withtotal - exp_ctg)**2) / exp_ctg
15     chi2.loc[:, 'All'] = chi2.sum(axis=1)
16     chi2.loc['All'] = chi2.sum()
17     print('---Chi-Square---')
18     display(chi2)
19
20
21     #get chi-square score
22     chi_square_score = chi2.iloc[:-1, :-1].sum().sum()
23
24     return chi_square_score

```

ฟังก์ชันสำหรับการคำนวณ chi-square โดยรับตาราง contingency เข้ามาหลังจากนั้นทำการหาค่า expected values โดยการนำคิดจาก (ผลรวมคอลัมน์ ณ ตำแหน่งนั้น * ผลรวมแถว ณ ตำแหน่งนั้น) / จำนวนทั้งหมด เมื่อได้ตาราง expected value ให้หา ผลต่างกำลังสองของค่าจริงและค่าคาดหวัง / จำนวนทั้งหมด แล้วค่า chi-square หาได้โดยนำแต่ละค่ามาบวกกันโดยไม่นับแถวและคอลัมน์สุดท้าย

```

1 # 5206 Kaew
2 # calling chi-square function
3 chi2_score = chi2_manual(test1_ctg_total)
4 print("Chi square score between hand and brain is equal to ", chi2_score)

```

เรียกใช้ฟังก์ชันคำนวณค่า chi-square และแสดงผลออกมา

Chi square score between hand and brain is equal to 3.4472880061115356

ผลลัพธ์ที่ได้จากการคำนวณ chi-square ระหว่างมือที่ถนัดและสมองซีกที่ถนัด

```

1 # 5206 Kaew
2 # calculate cramer's v value
3 def cramer_v(x2, num_row, num_col, data_size):
4     cramer = math.sqrt((x2/data_size)/(min((num_col)-1, (num_row)-1)))
5     deg_freedom = min((num_col)-1, (num_row)-1)
6     return cramer, deg_freedom

```

ฟังก์ชันคำนวณค่า Cramer's V โดยใช้ค่า chi-square และตาราง contingency แบบไม่มีผลรวมแต่ละแถวและคอลัมน์มาคิดตามสูตร โดยส่งค่ากลับมาเป็นผลลัพธ์จากการคำนวณและค่า degree of freedom

```

1 # 5206 Kaew
2 cramer, deg_freedom = cramer_v(chi2_score, ncol_ctg, nrow_ctg, n )
3 print("Cramer's V score between hand and brain is equal to ",cramer)
4 print("Degree of freedom between hand and brain is equal to ",deg_freedom)

```

เรียกใช้ฟังก์ชันในการคำนวณค่า Cramer's V และแสดงผลด้วยข้อความที่เหมาะสม

```

1 # 5206 Kaew
2 # Cramer's V interpretation
3 def cramer_itp(score, deg):
4     # degree of freedom = 1
5     if deg == 1:
6         if score < 0.1:
7             return "no relationship"
8         elif score >= 0.1:
9             return "small"
10        elif score >= 0.3:
11            return "medium"
12        elif score >= 0.5:
13            return "large"
14
15    # degree of freedom = 2
16    elif deg == 12:
17        if score < 0.07:
18            return "no relationship"
19        elif score >= 0.07:
20            return "small"
21        elif score >= 0.21:
22            return "medium"
23        elif score >= 0.35:
24            return "large"

```

ฟังก์ชันสำหรับตีความระดับความสัมพันธ์โดยดูจากค่า Cramer's V ที่คำนวณได้และค่า degree of freedom โดย

เงื่อนไขสร้างมาจากตารางตีความของ Cramer's V

```

1 # 5206 Kaew
2 def goodman (ctg_rawtotal, ctg_rawnototal, n):
3     #----- Find maximum in each column from contingency with total -----#
4     #ctg_total = test1_ctg_total
5     ctg_total_arr = ctg_rawtotal.to_numpy()
6     maxcolArr = np.amax(ctg_total_arr, axis=0)
7     #print(maxcolArr,"MaxcolArr")
8     max_colArr = max(maxcolArr[0:-1])
9     print("Max value in column (exclude last column) : ",max_colArr)
10
11     #----- Find maximum in each row from contingency without total -----#
12     #ctg_nototal = test1_ctg_nototal
13     ctg_nototal_arr = ctg_rawnototal.to_numpy()
14     maxrowArr = np.amax(ctg_nototal_arr, axis=1)
15     #print("maxrowArr",maxrowArr)
16
17     #----- Sum maximum in each row -----#
18     summaxrow = 0
19     for element in maxrowArr:
20         summaxrow += element
21
22     summax_row =summaxrow
23     print("Summation of max value in each row : ",summax_row)
24     #----- Prob E -----#
25     probE = 1-(max_colArr/n)
26     print("Reduction error probability of event : ",probE)
27
28     #-----Prob E|x -----#
29     probEX = 1-(summax_row/n)
30     print("Reduction error probability of event depends on another variable : ",probEX)
31
32     #-----Lambda-----#
33     lambda_EX = (probE-probEX)/probE
34
35     return lambda_EX
36
37
38

```

ฟังก์ชันสำหรับคำนวณค่า Goodman-Kruskal's lambda สามารถดูสูตรในการคำนวณได้จากหน้าที่ 7

```

1 # 5206 Kaew
2 # display result from Goodman-Kruskal's lambda for predicting brain from hand data
3 know_hand = goodman(test1_ctg_total, test1_ctg_nototal, n)
4 pc_knowHand = know_hand*100
5 print("*****")
6 print("Reduction error probability for predicting brain from hand : %.3f " %know_hand)
7 print()
8 print(f"It's mean that if we predict brain based on hand data can reduce {pc_knowHand:.2f} % of error. ")

```

เรียกใช้ฟังก์ชันในการคำนวณค่า Goodman-Kruskal's lambda โดยส่งตาราง contingency ทั้งแบบที่มี total และไม่มี total ให้แก่ฟังก์ชัน แล้วแสดงผลออกมาทั้งในรูปแบบความน่าจะเป็นและเปอร์เซ็นต์พร้อมกับข้อความที่เหมาะสม


```

1 # 5206 Kaew
2 # tranposed contingency table for predicting hand based on brain data
3 Tp_total = test1_ctg_total.transpose()
4 display(Tp_total)
5 print("*****")
6
7 Tp_nototal = test1_ctg_nototal.transpose()
8 display(Tp_nototal)

```

เนื่องจากต้องการคำนวณค่า Goodman-Kruskal's lambda ในคู่ตัวแปรเดียวกัน เพียงแต่ต้องการทำนายมือที่ถนัดจากสมองข้างที่ถนัด จึงต้องทำการ transpose ตาราง contingency ก่อนส่งเข้าไปคำนวณในฟังก์ชันต่อไป

```

1 # 5206 Kaew
2 # display result from Goodman-Kruskal's lambda for predicting hand from brain data
3 know_brain = goodman(Tp_total, Tp_nototal, n)
4 pc_knowBrain = know_brain*100
5 print("*****")
6 print("Reduction error probability for predicting hand from brain : %.3f " %know_brain)
7 print()
8 print(f"It's mean that if we predict hand based on brain data can reduce {pc_knowBrain:.2f} % of error. " )

```

เรียกใช้ฟังก์ชันคำนวณค่า Goodman-Kruskal's lambda โดยส่งค่า contingency ที่ transpose แล้วไปและแสดงผลด้วยข้อความที่เหมาะสม

เอกสารอ้างอิง

Cramer's V interpretation [ออนไลน์].เข้าถึงได้จาก : <https://www.statology.org/interpret-cramers-v/>
(วันที่ค้นข้อมูล :4 ธันวาคม 2564).

Matplotlib.bar stack [ออนไลน์].เข้าถึงได้จาก:

https://matplotlib.org/stable/gallery/lines_bars_and_markers/bar_stacked.html. (วันที่ค้นข้อมูล : 6 ธันวาคม 2564).

seaborn.swarm plot. [ออนไลน์].เข้าถึงได้จาก :

<https://seaborn.pydata.org/generated/seaborn.swarmplot.html>
(วันที่ค้นข้อมูล : 9 ธันวาคม 2564).

Overview of correlation measures. [ออนไลน์].เข้าถึงได้จาก :

<https://medium.com/@outside2SDs/an-overview-of-correlation-measures-between-categorical-and-continuous-variables-4c7f85610365>

(วันที่ค้นข้อมูล : 2 ธันวาคม 2564).

การวัดความสัมพันธ์.[ออนไลน์].เข้าถึงได้จาก : http://project.astyleplus.net/lesson3_5.html

(วันที่ค้นข้อมูล : 5 ธันวาคม 2564).