

วิเคราะห์ข้อมูลการเกิดอุบัติเหตุในประเทศไทย

โดยวิธี Exploratory data analysis

จัดทำโดย

นายชลพัฒน์	ปิ่นมณี	รหัสนักศึกษา 653380126-0
นายวสิษฐ์พล	พันชนกกุล	รหัสนักศึกษา 653380023-0
นายสุรภูมิ	มณีราชกิจ	รหัสนักศึกษา 653380283-4

อาจารย์ที่ปรึกษา

อาจารย์ ธนพล ตั้งชูพงศ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา CP351101 วิทยาการคำนวณ
ภาคเรียนที่ 2 ปีการศึกษา 2565 ภาควิชาวิทยาการคอมพิวเตอร์
คณะวิทยาลัยการคอมพิวเตอร์มหาวิทยาลัยขอนแก่น

นายชลพัฒน์	ปิ่นมณี	หน้าที่ สืบค้นข้อมูล เขียนโปรแกรม
นายวิสิทธิ์พล	พັນชนกกุล	หน้าที่ สืบค้นข้อมูล จัดทำรูปเล่ม
นายสุรภูมิ	มณีราชกิจ	หน้าที่ สืบค้นข้อมูล ทำ Presentation

คำนำ

รายงานการจัดทำโครงงานวิทยาศาสตร์ เรื่อง วิเคราะห์ข้อมูลการเกิดอุบัติเหตุในประเทศไทย โดยวิธี Exploratory data analysis เป็นการนำข้อมูลการเกิดอุบัติเหตุในประเทศไทยมาคัดกรองและสรุปผลออกมาในลักษณะของกราฟ โดยผู้จัดทำโครงงานได้ทำการศึกษาเพื่อให้ทราบถึงปัจจัยของการเกิดอุบัติเหตุในประเทศไทย

โดยคณะผู้จัดทำก็หวังเป็นอย่างยิ่งว่ารายงานโครงงานเล่มนี้จะมีประโยชน์สำหรับผู้ที่ศึกษาต่อในเรื่องวิเคราะห์ข้อมูลการเกิดอุบัติเหตุในประเทศไทยโดยวิธี Exploratory data analysis ไม่น่ามากก็น้อย และถ้าหากมีสิ่งใดที่บกพร่อง คณะผู้จัดทำขอน้อมรับไปพิจารณา เพื่อปรับปรุงให้ดียิ่งขึ้น

คณะผู้จัดทำ

โครงการ	วิเคราะห์ข้อมูลการเกิดอุบัติเหตุในประเทศไทยโดยวิธี Exploratory data analysis		
ผู้จัดทำ	นายชลพัฒน์	ปิ่นมณี	รหัสนักศึกษา 653380126-0
	นายวชิษฐ์พล	พັນชนกกุล	รหัสนักศึกษา 653380023-0
	นายสุรภูมิ	มณีราชกิจ	รหัสนักศึกษา 653380283-4
อาจารย์ที่ปรึกษา	อาจารย์ ธนพล ตั้งชูพงศ์		
สถานที่ปรึกษา	มหาวิทยาลัยขอนแก่น		

บทคัดย่อ

โครงการนี้เป็นส่วนหนึ่งของรายวิชา CP351101 วิทยาการคำนวณ คณะผู้จัดทำได้ทำการศึกษาเรื่อง วิเคราะห์ข้อมูลการเกิดอุบัติเหตุในประเทศไทยโดยวิธี Exploratory data analysis โดยทางคณะผู้จัดทำได้ ทำการศึกษาข้อมูลการเกิดอุบัติเหตุบนท้องถนน แล้วนำมาเขียนโปรแกรมเพื่อคัดกรองข้อมูล แล้วสรุปเป็นกราฟออกมา โดยมีจุดประสงค์เพื่อให้ผู้อ่านได้ทราบถึงปัจจัยต่าง ๆ ของการเกิดอุบัติเหตุของในประเทศไทย ซึ่งโครงการนี้มีการใช้ความรู้เกี่ยวกับ ไลบรารีของไพธอน อย่าง numpy matplotlib pandas

คณะผู้จัดทำขอขอบพระคุณ อาจารย์ ธนพล ตั้งชูพงศ์ ผู้ให้ความรู้และชี้แนะแนวทางในการศึกษา ทางคณะผู้จัดทำหวังว่าโครงการฉบับนี้จะเป็นประโยชน์แก่ผู้อ่านทุกท่าน

สารบัญ

	หน้า
คำนำ	ก
บทคัดย่อ	ข
สารบัญ	ค
สารบัญรูปภาพ	ง
สารบัญตาราง	ญ
บทที่ 1 บทนำ	1
- 1.1 ที่มาและความสำคัญ	1
- 1.2 วัตถุประสงค์	1
- 1.3 ขอบเขตของโครงการ	2
- 1.4 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและเนื้อหาที่เกี่ยวข้อง	3
- 2.1 ทฤษฎี	3
- 2.2 งานที่เกี่ยวข้อง	3
บทที่ 3 วิธีดำเนินการทำโครงการ	4
- 3.1 ขั้นตอนวิธีที่จะใช้แก้ปัญหา	4
- 3.2 เครื่องมือที่ใช้ในโครงการ	5
- 3.3 แผนการดำเนินโครงการ	5
- 3.3.1 การดำเนินโครงการ	6
บทที่ 4 ผลการดำเนินงาน	31
บทที่ 5 สรุปผลการดำเนินงาน และข้อเสนอแนะ	46
- 5.1 วัตถุประสงค์ของโครงการ	46
- 5.2 เครื่องมือที่ใช้ในโครงการ	46
- 5.3 สรุปผลการดำเนินงาน	46
- 5.4 ข้อเสนอแนะ	48
บรรณานุกรม	ฎ

สารบัญรูปภาพ

	หน้า
4.1 Data transformation / Cleaning data	
รูปภาพที่ 1 อ่านไฟล์ข้อมูล	6
รูปภาพที่ 2 ตัวอย่างผลลัพธ์จากการอ่านไฟล์	6
รูปภาพที่ 3 การเปลี่ยนชื่อแถวในแนวนคอลัมภ์	7
รูปภาพที่ 4 การเปลี่ยนชื่อแถวในแนวนคอลัมภ์	7
รูปภาพที่ 5 การลบคอลัมภ์ที่ไม่ต้องการ	8
รูปภาพที่ 6 อัปเดตคอลัมภ์ที่เหลืออยู่	8
รูปภาพที่ 7 เช็คข้อมูลที่ว่างหรือขาดหาย	8
รูปภาพที่ 8 ตัวอย่างผลรวมของค่าว่างและเปอร์เซ็นต์ของแต่ละคอลัมภ์	9
รูปภาพที่ 9 ลบข้อแถวที่มีค่าขาดหายหรือค่าว่างออกไป	9
รูปภาพที่ 10 อัปเดตข้อมูลที่ว่างหลังจากลบแถวที่มีค่าว่างออก	10
รูปภาพที่ 11 การแยก Date ออกมาเป็น Day/Month/Year	10
รูปภาพที่ 12 ตัวอย่างผลลัพธ์ของ Day/Month/Year	11
รูปภาพที่ 13 การแยก Time ออกมาเป็น Hour/Minute/Second	11
รูปภาพที่ 14 ผลลัพธ์หลังจากแยก Hour/Minute/Second	12

4.2 Data analysis / Visualization

- เดือนใดมีความถี่การเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและมีค่าเฉลี่ยเท่าใด

รูปภาพที่ 15 วิเคราะห์ข้อมูลระหว่างความถี่ในการเกิดอุบัติเหตุต่อเดือน 13

รูปภาพที่ 16 ผลลัพธ์ความสัมพันธ์ระหว่างความถี่การเกิดอุบัติเหตุต่อเดือน 13

รูปภาพที่ 17 เขียนโปรแกรมเพื่อพล็อตกราฟ 14

- วันไหนของเดือนที่มีความถี่การเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและมีค่าเฉลี่ยเท่าใด

รูปภาพที่ 18 วิเคราะห์ข้อมูลระหว่างความถี่การเกิดอุบัติเหตุต่อวัน 14

รูปภาพที่ 19 ผลลัพธ์ของการวิเคราะห์ข้อมูล 14

รูปภาพที่ 20 เขียนโปรแกรมเพื่อพล็อตกราฟ 15

- เวลาที่ชั่วโมงใดมีความถี่ในการเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและค่าเฉลี่ยเป็นเท่าใด

รูปภาพที่ 21 วิเคราะห์ข้อมูลของความสัมพันธ์ระหว่างความถี่ในการเกิดอุบัติเหตุต่อชั่วโมง 15

รูปภาพที่ 22 ผลลัพธ์ของการวิเคราะห์ข้อมูล 15

รูปภาพที่ 23 เขียนโปรแกรมในการพล็อตกราฟ 16

- จังหวัดใดมีความถี่การเกิดอุบัติเหตุมากที่สุด 5 อันดับแรก

รูปภาพที่ 24 วิเคราะห์ข้อมูลความถี่การเกิดอุบัติเหตุของแต่ละจังหวัด 16

รูปภาพที่ 25 ผลลัพธ์ของการวิเคราะห์ข้อมูล 17

รูปภาพที่ 26 การหาค่าความถี่ของจังหวัดที่เกิดอุบัติเหตุมากที่สุด น้อยที่สุด

และค่าเฉลี่ยของการเกิดอุบัติเหตุ 17

รูปภาพที่ 27 ผลลัพธ์ของการวิเคราะห์ข้อมูล 17

รูปภาพที่ 28 เขียนโปรแกรมในการพล็อตกราฟ	18
- ถนนหรือทางแบบใดที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก	
รูปภาพที่ 29 วิเคราะห์ข้อมูลว่าระหว่างความถี่ในการเกิดอุบัติเหตุต่อจังหวัด	18
รูปภาพที่ 30 ผลลัพธ์ของการวิเคราะห์ข้อมูล	18
รูปภาพที่ 31 เขียนโปรแกรมเพื่อพล็อตกราฟ	19
- ลักษณะการเกิดอุบัติเหตุแบบใดที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก	
รูปภาพที่ 32 วิเคราะห์ข้อมูลระหว่างความถี่ในลักษณะการเกิดอุบัติเหตุของแต่ละแบบ	19
รูปภาพที่ 33 ผลลัพธ์ของการวิเคราะห์ข้อมูล	19
รูปภาพที่ 34 เขียนโปรแกรมเพื่อพล็อตกราฟ	20
- เดือนใดที่มีจำนวนผู้เสียชีวิตและผู้บาดเจ็บรวมสูงที่สุด	
รูปภาพที่ 35 วิเคราะห์ข้อมูลความถี่ผู้บาดเจ็บ และผู้เสียชีวิตในแต่ละเดือน	20
รูปภาพที่ 36 ผลลัพธ์ของการวิเคราะห์ข้อมูล	21
รูปภาพที่ 37 เขียนโปรแกรมเพื่อพล็อตกราฟ	21
- ประเภทของทางบนถนนแบบใดที่มีความถี่ของการเกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับแรกในแต่ละเดือน	
รูปภาพที่ 38 วิเคราะห์ข้อมูลความถี่ประเภทของทางบนถนนที่เกิดอุบัติเหตุต่อเดือน	22
รูปภาพที่ 39 ผลลัพธ์ของการวิเคราะห์ข้อมูล	22
รูปภาพที่ 40 เขียนโปรแกรมเพื่อพล็อตกราฟ	22
- ลักษณะการเกิดอุบัติเหตุแบบใดที่มีความถี่ในการเกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับในแต่ละเดือน	
รูปภาพที่ 41 วิเคราะห์ข้อมูลความถี่ของลักษณะการเกิดอุบัติเหตุต่อเดือน	23
รูปภาพที่ 42 ผลลัพธ์ของการวิเคราะห์ข้อมูล	23

รูปภาพที่ 43 เขียนโปรแกรมเพื่อพล็อตกราฟ	24
- เดือนใดที่มีจำนวนรถที่เกิดเหตุทั้งหมดมากที่สุด	
รูปภาพที่ 44 วิเคราะห์ข้อมูลความถี่ของจำนวนรถที่เกิดเหตุต่อเดือน	24
รูปภาพที่ 45 เขียนโปรแกรมเพื่อพล็อตกราฟ	24
- จากลักษณะการเกิดอุบัติเหตุแต่ละแบบส่วนใหญ่เกิดกับยานพาหนะประเภทใด	
รูปภาพที่ 46 วิเคราะห์ข้อมูลความถี่ประเภทยานพาหนะต่อลักษณะการเกิดอุบัติเหตุ	25
รูปภาพที่ 47 ผลลัพธ์ของการวิเคราะห์ข้อมูล	25
รูปภาพที่ 48 เขียนโปรแกรมเพื่อพล็อตกราฟ	25
- ช่วงวันใดควรมีมาตรการระวังการเดินทางโดยพิจารณาจากจำนวนรถที่เกิดเหตุต่อวันและเดือน	
รูปภาพที่ 49 กระจายข้อมูลความถี่แต่ละวันในแต่ละเดือนต่อจำนวนรถที่เกิดเหตุต่อวัน	
และเดือน	26
รูปภาพที่ 50 ลัพธ์ของการวิเคราะห์ข้อมูล	26
รูปภาพที่ 51 เขียนโปรแกรมเพื่อพล็อตกราฟ	27
- อุบัติเหตุที่เกิดในสภาพอากาศแบบใดบ้าง โดยเทียบกับผลรวมของยานพาหนะที่เกิดเหตุ	
รูปภาพที่ 52 วิเคราะห์ข้อมูลความถี่สภาพอากาศต่อผลรวมของยานพาหนะที่เกิดเหตุ	27
รูปภาพที่ 53 ผลลัพธ์ของการวิเคราะห์ข้อมูล	28
รูปภาพที่ 54 เขียนโปรแกรมเพื่อพล็อตกราฟ	28
- สัดส่วนความถี่ของผู้ที่บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บ, ไม่เสียชีวิตเลยเป็นอย่างไร	
รูปภาพที่ 55 วิเคราะห์ข้อมูลความถี่ของที่บาดเจ็บอย่างน้อย 1 คน,	

ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บ และ ไม่เสียชีวิตเลย	29
รูปภาพที่ 56 ผลลัพธ์ของการวิเคราะห์ข้อมูล	29
- เทียบสาเหตุการเกิดอุบัติเหตุแต่ละแบบ ว่ามีผลรวมของผู้เสียชีวิตกับผู้บาดเจ็บเป็นอย่างไร	
รูปภาพที่ 57 เขียนโปรแกรมเปรียบเทียบสาเหตุการเกิดอุบัติเหตุ	
เพื่อหาผลรวมผู้เสียชีวิตกับผู้บาดเจ็บแล้วนำมาพล็อตกราฟ	30
ผลการดำเนินงาน	
รูปภาพที่ 58 กราฟแสดงความถี่การเกิดอุบัติเหตุต่อเดือน	31
รูปภาพที่ 59 กราฟความสัมพันธ์ระหว่างความถี่ในการเกิดอุบัติเหตุต่อวัน	31
รูปภาพที่ 60 กราฟเส้นแสดงความสัมพันธ์ระหว่างความถี่ในการเกิดอุบัติเหตุต่อชั่วโมง	32
รูปภาพที่ 61 กราฟแสดงความถี่การเกิดอุบัติเหตุของแต่ละจังหวัดโดยเรียงลำดับจากมากไปน้อย	33
รูปภาพที่ 62 จังหวัดที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 5 อันดับแรก	33
รูปภาพที่ 63 กราฟถนนหรือทางที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก	34
รูปภาพที่ 64 กราฟที่ถนนหรือทางที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก	34
รูปภาพที่ 65 สรุปยอดผู้เสียชีวิตรวมจากอุบัติเหตุสูงมากที่สุด	35
รูปภาพที่ 66 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 1 – 3	36
รูปภาพที่ 67 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 4 – 6	36
รูปภาพที่ 68 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 7 – 9	37
รูปภาพที่ 69 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 10 – 12	37
รูปภาพที่ 70 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 1 – 3	38
รูปภาพที่ 71 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 4 – 6	38

รูปภาพที่ 72 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 7 – 9	39
รูปภาพที่ 73 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 10 – 12	39
รูปภาพที่ 74 กราฟสรุปเดือนที่มีจำนวนรถที่เกิดเหตุมากที่สุดทั้งแบบรวมคันแรก และแบบรวมคนเดินเท้า	40
รูปภาพที่ 75 กราฟสรุปประเภทของยานพาหนะที่เกิดอุบัติเหตุ	41
รูปภาพที่ 76 กราฟแบบ Heatmap ของวันที่มีโอกาสเกิดอุบัติเหตุสูง	42
รูปภาพที่ 77 กราฟสรุปจำนวนของรถที่เกิดอุบัติเหตุตามสภาพอากาศต่าง ๆ	43
รูปภาพที่ 78 ผลสรุปอุบัติเหตุส่วนใหญ่ที่เกิดขึ้นจากผู้ประสบอุบัติเหตุ	44
รูปภาพที่ 79 กราฟสรุปสาเหตุของการเกิดอุบัติเหตุ กับผู้ที่ประสบอุบัติเหตุ	45

สารบัญตาราง

หน้า

แผนการดำเนินโครงการ

5

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

อุบัติเหตุทางท้องถนนในประเทศไทยนั้นเกิดขึ้นได้ทุกที่ทุกเวลาเกิดขึ้นได้และเกิดได้กับทุกเพศทุกวัย ซึ่งทำให้มีผู้บาดเจ็บ และผู้เสียชีวิตเป็นจำนวนมาก อีกทั้งยังสร้างความเสียหายแก่ทรัพย์สิน และระบบเศรษฐกิจในไทย ซึ่งสาเหตุของการเกิดอุบัติเหตุมีหลายรูปแบบ และหลายปัจจัยด้วยกัน

จากปัญหาที่เกิดขึ้นคณะผู้จัดทำจึงสนใจที่จะทำนายอัตราการเกิดอุบัติเหตุทางถนนในประเทศไทย โดยอาศัยวิธี Exploratory data analysis และจะทำการตั้งคำถามที่ทางคณะผู้จัดทำสนใจ และเขียนโปรแกรม เพื่อดูผลลัพธ์ของการทดลองว่ามีความสัมพันธ์กันอย่างไร และนำไปสรุป

ดังนั้นทางคณะผู้จัดทำจึงอยากนำความรู้ในการเขียนโปรแกรมมาเรียบเรียงและทำการวิเคราะห์ข้อมูลนี้ โดยจะแปรผลข้อมูลให้อยู่ในรูปตารางหรือกราฟหรือในรูปแบบอื่นๆ ที่ให้อ่านสามารถอ่านแล้วเข้าใจได้ง่าย

1.2 วัตถุประสงค์

1. เพื่อหาว่าเดือนใดที่มีอัตราการเกิดอุบัติเหตุมากที่สุดและน้อยที่สุด และมีค่าเฉลี่ยเท่าใด
2. เพื่อหาว่าวันไหนของเดือนที่มีอัตราการเกิดอุบัติเหตุมากที่สุด และน้อยที่สุดมีค่าเฉลี่ยเท่าใด
3. เพื่อหาช่วงเวลาที่เกิดอุบัติเหตุมากที่สุด และน้อยที่สุด
4. เพื่อหา 5 อันดับแรกของจังหวัดที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
5. เพื่อหา 3 อันดับแรกของถนนหรือทางที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
6. เพื่อหา 3 อันดับแรกของลักษณะการเกิดอุบัติเหตุที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
7. เพื่อหาว่าเดือนใดที่มีจำนวนผู้เสียชีวิต และผู้บาดเจ็บรวมจากอุบัติเหตุที่มากที่สุด
8. เพื่อหาว่า 3 อันดับแรกของประเภทรถที่ทำให้เกิดอุบัติเหตุที่สุดในแต่ละเดือน
9. เพื่อหาว่า 3 อันดับของลักษณะการเกิดอุบัติเหตุที่มากที่สุดในแต่ละเดือน
10. เพื่อหาว่าเดือนใดที่มีจำนวนรถที่เกิดอุบัติเหตุมากที่สุด
11. เพื่อหาว่าจากลักษณะการเกิดอุบัติเหตุแต่ละแบบมักเกิดกับยานพาหนะประเภทใด
12. เพื่อหาว่าช่วงวันไหนของเดือนที่ควรระมัดระวังในการเดินทาง
13. เพื่อหาว่าอุบัติเหตุเกิดขึ้นนั้นเกิดในสภาพอากาศลักษณะใดบ้าง
14. เพื่อหาสัดส่วนของอัตราผู้บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บ และไม่เสียชีวิตเลย

15. เพื่อนำมาเปรียบเทียบสาเหตุการเกิดอุบัติเหตุแต่ละแบบว่าผลรวมของผู้เสียชีวิตเป็นอย่างไรบ้าง

1.3 ขอบเขตของโครงการ

สิ่งที่ศึกษา

- วิเคราะห์การเกิดอุบัติเหตุในประเทศไทยโดยวิธี Exploratory data analysis จากการตั้งคำถามต่างๆ

ขอบเขตของประชากร

- ผู้ขับขี่รถบนท้องถนน ในประเทศไทย

ขอบเขตของข้อมูล

- เป็นข้อมูลการเกิดอุบัติเหตุบนทางถนนและในประเทศไทยเท่านั้น

โดยนำข้อมูลมาจาก <https://datagov.mot.go.th/> เป็นเว็บไซต์กระทรวงคมนาคมของไทย

1.4 ผลที่คาดว่าจะได้รับ

1. ได้ทราบเดือนที่มีอัตราการเกิดอุบัติเหตุมากที่สุดและน้อยที่สุด พร้อมค่าเฉลี่ย
2. ได้ทราบว่าวันไหนของเดือนที่มักเกิดอุบัติเหตุมากที่สุด และน้อยที่สุด พร้อมค่าเฉลี่ย
3. ได้ทราบช่วงเวลาที่เกิดอุบัติเหตุมากที่สุด และน้อยที่สุด
4. ได้ทราบ 5 อันดับแรกของจังหวัดที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
5. ได้ทราบ 3 อันดับแรกของถนนหรือทางที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
6. ได้ทราบ 3 อันดับแรกของลักษณะการเกิดอุบัติเหตุที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
7. ได้ทราบถึงเดือนที่มีจำนวนผู้เสียชีวิต และผู้บาดเจ็บรวมจากอุบัติเหตุที่มากที่สุด
8. ได้ทราบ 3 อันดับแรกของประเภทรถที่ทำให้เกิดอุบัติเหตุมากที่สุดในแต่ละเดือน
9. ได้ทราบ 3 อันดับของลักษณะการเกิดอุบัติเหตุที่มากที่สุดในแต่ละเดือน
10. ได้ทราบว่าเดือนใดที่มีจำนวนรถที่เกิดอุบัติเหตุมากที่สุด
11. ได้ทราบประเภทยานพาหนะที่มักเกิดอุบัติเหตุเมื่อนำมาเปรียบเทียบกับลักษณะการเกิดอุบัติเหตุแต่ละแบบ
12. ได้ทราบว่าช่วงวันไหนของเดือนที่ควรระมัดระวังในการเดินทางมากกว่าปกติ
13. ได้ทราบลักษณะของสภาพอากาศที่เกิดอุบัติเหตุ
14. ได้ทราบสัดส่วนของอัตราผู้บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บและไม่เสียชีวิตเลย
15. สามารถนำเอาข้อมูลมาเปรียบเทียบสาเหตุการเกิดอุบัติเหตุแต่ละแบบทำให้ทราบถึงผลรวมของผู้เสียชีวิตได้

บทที่ 2

ทฤษฎีและเนื้อหาที่เกี่ยวข้อง

2.1 ทฤษฎี

การวิเคราะห์ข้อมูลเชิงสำรวจ (Exploratory Data Analysis / EDA) เป็นแนวทางในการวิเคราะห์ชุดข้อมูลที่มุ่งเน้นไปที่การทำความเข้าใจโครงสร้างพื้นฐานของข้อมูลและความสัมพันธ์ระหว่างตัวแปร EDA เป็นขั้นตอนที่สำคัญในกระบวนการวิเคราะห์ข้อมูลเนื่องจากสามารถช่วยระบุรูปแบบความผิดปกติและแนวโน้มที่สามารถแจ้งการวิเคราะห์เพิ่มเติมได้

EDA เกี่ยวข้องกับเทคนิคหลายอย่าง มีทั้งการวิเคราะห์ทางสถิติ การสร้างแผนภูมิ และการลดขนาด การวิเคราะห์ทางสถิติ เช่น ค่าเฉลี่ย ค่ามัธยฐาน และส่วนเบี่ยงเบนมาตรฐาน ให้ภาพรวมระดับสูงของแนวโน้ม ศูนย์กลางและความแปรปรวนของชุดข้อมูล เทคนิคการสร้างแผนภูมิ เช่น แผนภาพกระจาย ฮิสโตแกรม และแผนภาพกล่อง สามารถช่วยระบุความสัมพันธ์และรูปแบบระหว่างตัวแปรได้ เทคนิคการลดขนาด เช่น Principal Component Analysis (PCA) สามารถช่วยลดจำนวนของตัวแปรในชุดข้อมูล ในขณะที่รักษาข้อมูลพื้นฐานไว้ให้มากที่สุด

2.2 งานที่เกี่ยวข้อง

มีการศึกษามากมายเกี่ยวกับการใช้ EDA ในสาขาต่างๆ รวมถึงการเงิน การดูแลสุขภาพ และวิศวกรรม ตัวอย่างเช่น ในด้านการเงิน EDA ถูกนำมาใช้เพื่อระบุแนวโน้มและรูปแบบในข้อมูลตลาดหุ้นเพื่อเป็นข้อมูลกลยุทธ์การลงทุน ในการดูแลสุขภาพ EDA ถูกนำมาใช้เพื่อระบุปัจจัยเสี่ยงสำหรับโรคบางชนิด และเพื่อทำความเข้าใจประสิทธิภาพของการรักษาต่างๆ ในทางวิศวกรรม EDA ถูกนำมาใช้เพื่อวิเคราะห์ข้อมูลเซ็นเซอร์จากเครื่องจักรเพื่อระบุปัญหาที่อาจเกิดขึ้นก่อนที่จะร้ายแรงกว่านี้

นอกเหนือจากการใช้งานเหล่านี้แล้วยังมีการศึกษามากมายเกี่ยวกับประสิทธิภาพของเทคนิค EDA ต่างๆ ตัวอย่างเช่น การศึกษาที่เปรียบเทียบประสิทธิผลของเทคนิคการแสดงผลที่แตกต่างกันในการระบุความสัมพันธ์ระหว่างตัวแปร และการศึกษาประสิทธิผลของเทคนิคการลดขนาดที่แตกต่างกันในการเก็บรักษาข้อมูลพื้นฐานในชุดข้อมูล

โดยรวมแล้ว EDA เป็นแนวทางที่หลากหลายและมีประสิทธิภาพในการวิเคราะห์ข้อมูล ซึ่งสามารถนำไปใช้กับฟิลด์และชุดข้อมูลต่างๆ มากมาย ด้วยเทคนิคที่หลากหลาย EDA สามารถช่วยให้นักวิเคราะห์ได้รับความเข้าใจที่ลึกซึ้งยิ่งขึ้นเกี่ยวกับข้อมูล และให้ข้อมูลการวิเคราะห์และการตัดสินใจเพิ่มเติมได้

บทที่ 3

วิธีดำเนินการทำโครงการ

3.1 ขั้นตอนวิธีที่จะใช้แก้ปัญหา

1.เริ่มจากการทำ Data Transformation / Data Cleansing

ข้อมูลดิบก็เหมือนน้ำมันดิบ ถึงจะมีค่ามาก แต่ถ้ายังไม่ได้ถูกนำมาแปลงและวิเคราะห์อย่างเหมาะสม ก็แทบจะไม่มีค่าเลยการที่เราจะนำข้อมูลไปวิเคราะห์หรือสร้างแบบจำลองทางสถิติต่อไปได้จะต้องผ่านการ เตรียมข้อมูล หรือ Data Cleansing ให้พร้อมก่อน โดยขั้นตอนการทำความสะอาดข้อมูลนี้ถือเป็นขั้นตอนที่ต้องใช้ระยะเวลานาน และสำคัญที่สุด

ตัวอย่างการเตรียมข้อมูลเช่น

- Handle missing values จัดการกับแถวที่มีข้อมูลหายไป ด้วยการลบหรือเติมค่าลงไปให้เหมาะสม
- การเปลี่ยน Common Data types ให้เหมาะสม

2. การวิเคราะห์ข้อมูล หรือ Data Analysis

เมื่อเราเตรียมข้อมูลพร้อมแล้ว เราต้องวิเคราะห์เพื่อหาข้อมูลเชิงลึกมาต่อยอด ไม่ว่าจะเป็น

- ความสัมพันธ์ต่างๆ ของตัวแปร
- Patterns ที่ปรากฏ
- การคำนวณค่าสถิติต่างๆ

โดยจะวิเคราะห์ข้อมูลผ่านการเขียนโปรแกรมภาษา Python และไลบรารีต่างๆ เช่น Numpy, Pandas เป็นต้น

3. Data Visualization

เมื่อเราวิเคราะห์ข้อมูลออกมาได้ สิ่งสำคัญคือการสื่อสารออกไปให้คนอื่นในทีมเห็นเป็นภาพเดียวกัน การทำ Data Visualization คือการสร้างกราฟ หรือ Chart ต่างๆ เพื่อให้เราเข้าใจข้อมูลเชิงลึกได้ง่ายขึ้น จากการแปลงข้อมูลให้เป็นภาพ มองปราดเดียวแล้วเข้าใจง่ายขึ้น

- ใช้ libraries ของ python numpy pandas ในการจัดการกับข้อมูลดิบให้อยู่ในรูปแบบที่ต้องการ และใช้ matplotlib ในการแสดงผลและวิเคราะห์แนวโน้มที่เกิดขึ้น

3.2 เครื่องมือที่ใช้ในโครงการ

- Google Colab ใช้ในการเขียนโปรแกรมภาษา Python
- ไลบรารีต่างๆของ Python ตัวอย่างเช่น Numpy matplotlib Pandas เป็นต้น

3.3 แผนการดำเนินโครงการ

ลำดับ	สัปดาห์ที่	กิจกรรม
1	6 กพ. - 12 กพ.	กำหนดหัวข้อโครงการที่สนใจ
2	13 กพ. - 19 กพ.	นำเสนอเค้าโครงโครงการ
3	20 กพ. - 26 กพ.	สืบค้นและรวบรวมข้อมูล
4	27 กพ. - 5 มี.ค.	เริ่มเขียนโค้ด
5	6 มี.ค. - 12 มี.ค.	ตรวจสอบและแก้ไขปัญหาที่เกิดขึ้น
6	13 มี.ค. - 19 มี.ค.	ส่งโครงการ
7	20 มี.ค. - 21 มี.ค.	นำเสนอโครงการ

3.3.1 การดำเนินโครงการ

1.Data transformation / Cleaning data

```
Accidents = pd.read_excel('accident2022.xlsx') #read data
Accidents
```

รูปภาพที่ 1 อ่านไฟล์ข้อมูล

	ปีที่เกิดเหตุ	วันที่เกิดเหตุ	เวลา	วันที่รายงาน	เวลาที่รายงาน	ACC_CODE	หน่วยงาน	รหัสสายทาง	สายทาง	ก.ม.	...	รถบรรทุก 6 ล้อ	รถบรรทุกมากกว่า 6 ล้อ ไม่เกิน 10 ล้อ	รถบรรทุกมากกว่า 10 ล้อ (รถพ่วง)	รถอื่น	อื่นๆ	คนเดินเท้า	จำนวนผู้เสียชีวิต	จำนวนผู้บาดเจ็บสาหัส	จำนวนผู้บาดเจ็บเล็กน้อย	รวมจำนวนผู้บาดเจ็บ
0	2022	2022-01-01	00:01:00	2022-01-02	11:45:00	6566872	กรมทางหลวงชนบท	ชน.5016	เทศบาลตำบลวัดสิงห์ - บ้านน้ำพุ (ช่วงหน้าตลาด)	17.430	...	0	0	0	0	0	0	0	1	0	1
1	2022	2022-01-01	00:01:00	2022-01-02	11:44:00	6566880	กรมทางหลวงชนบท	มค.4012	แยกทางหลวงหมายเลข 2152 (กม.ที่ 31+700) - บ้านค...	0.500	...	0	0	0	0	0	0	0	0	1	1
2	2022	2022-01-01	00:03:00	2022-02-09	08:41:00	5706553	กรมทางหลวง	4	พุดตาสินข้าง - รังครก	469.800	...	0	0	0	0	0	0	1	0	0	0
3	2022	2022-01-01	00:05:00	2022-01-02	06:21:00	5485750	กรมทางหลวง	4030	ถลาง - หาดราไวย์	17.125	...	0	0	0	0	0	0	0	0	1	1
4	2022	2022-01-01	00:05:00	2022-01-24	09:59:00	5624452	กรมทางหลวง	216	ถนนวงแหวนรอบเมืองอุดรธานีด้านทิศตะวันออก	25.350	...	0	0	0	0	0	0	0	0	2	2

รูปภาพที่ 2 ตัวอย่างผลลัพธ์จากการอ่านไฟล์เบื้องต้น

```
# Rename each columns
Mapping = {
    'ปีที่เกิดเหตุ' : 'Year',
    'วันที่เกิดเหตุ' : 'Date',
    'เวลา' : 'Time',
    'วันที่รายงาน' : 'DateReport',
    'เวลาที่รายงาน' : 'TimeReport',
    'ACC_CODE' : 'Acc',
    'หน่วยงาน' : 'Agency',
    'รหัสสายทาง' : 'RouteCode',
    'สายทาง' : 'Route',
    'ก.ม.' : 'Km',
    'จังหวัด' : 'Province',
    'รถคันที่ 1' : 'FirstCar',
    'บริเวณที่เกิดเหตุ/ลักษณะทาง' : 'RoadType',
    'มูลเหตุสันนิษฐาน' : 'Cause',
    'ลักษณะการเกิดอุบัติเหตุ' : 'AccidentType',
    'สภาพอากาศ' : 'Weather',
    'LATITUDE' : 'Lat',
    'LONGITUDE' : 'Long',
    'จำนวนรถที่เกิดเหตุ (รวมคันที่ 1)' : 'NumOfCar_Include_firstCar',
    'จำนวนที่เกิดเหตุทั้งหมด (รวมคนเดินเท้า)' : 'NumOfAccident_Include_pedestrian',
    'รถจักรยานยนต์' : 'Motorcycle',
    'รถสามล้อเครื่อง' : 'tricycle',
    'รถยนต์นั่งส่วนบุคคล/รถยนต์นั่งสาธารณะ' : 'Public_Cars',
    'รถตู้' : 'Van',
    'รถปิคอัพโดยสาร' : 'Passenger_pickup_truck',
    'รถโดยสารมากกว่า 4 ล้อ' : 'Bus_LessThan4wheels',
    'รถปิคอัพบรรทุก 4 ล้อ' : 'PickupTruck_4wheels',
    'รถบรรทุก 6 ล้อ' : 'Truck_6wheels',
    'รถบรรทุกมากกว่า 6 ล้อ ไม่เกิน 10 ล้อ' : 'Truck_LessThan10wheels',
```

รูปภาพที่ 3 เปลี่ยนชื่อแถวในแนวคอลัมน์

```
'รถบรรทุกมากกว่า 10 ล้อ (รถพ่วง)' : 'Truck_MoreThan10wheels',
'รถอีแต่น' : 'Etan_car',
'อื่นๆ' : 'Else',
'คนเดินเท้า' : 'Pedestrians',
'จำนวนผู้เสียชีวิต' : 'Dead',
'จำนวนผู้บาดเจ็บสาหัส' : 'Seriously_injured',
'จำนวนผู้บาดเจ็บเล็กน้อย' : 'Minor_Injured',
'รวมจำนวนผู้บาดเจ็บ' : 'Total_Injured'
}
Accidents.rename(index=str, columns=Mapping, inplace=True)
```

รูปภาพที่ 4 เปลี่ยนชื่อแถวในแนวคอลัมน์

```
# Delete column ที่ไม่ต้องการ
Accidents.drop(['Year'],axis=1,inplace = True)
Accidents.drop(['Acc'],axis=1,inplace = True)
Accidents.drop(['RouteCode'],axis=1,inplace = True)
Accidents.drop(['Route'],axis=1,inplace = True)
Accidents.drop(['Km'],axis=1,inplace = True)
Accidents.drop(['Lat'],axis=1,inplace = True)
Accidents.drop(['Long'],axis=1,inplace = True)
Accidents.drop(['DateReport'],axis=1,inplace = True)
Accidents.drop(['TimeReport'],axis=1,inplace = True)
```

รูปภาพที่ 5 ลบคอลัมภ์ที่ไม่ต้องการ

```
Index(['Date', 'Time', 'Agency', 'Province', 'FirstCar', 'RoadType', 'Cause',
      'AccidentType', 'Weather', 'NumOfCar_Include_firstCar',
      'NumOfAccident_Include_pedestrian', 'Motorcycle', 'tricycle',
      'Public_Cars', 'Van', 'Passenger_pickup_truck', 'Bus_LessThan4wheels',
      'PickupTruck_4wheels', 'Truck_6wheels', 'Truck_LessThan10wheels',
      'Truck_MoreThan10wheels', 'Etan_car', 'Else', 'Pedestrians', 'Dead',
      'Seriously_injured', 'Minor_Injured', 'Total_Injured'],
      dtype='object')
```

รูปภาพที่ 6 อัปเดตคอลัมภ์ที่เหลืออยู่

```
# เช็คค่าความถี่ของข้อมูลที่ขาดหาย
def missing_data(data): # check Nan value >>> NaN คือ ข้อมูลว่างหรือขาดหาย
    total = data.isnull().sum().sort_values(ascending = False)
    percent = (data.isnull().sum()/data.isnull().count()*100).sort_values(ascending = False)
    return pd.concat([total, percent], axis=1, keys=['Total NaN Values', 'Percentage of NaN Values'])

missing_data(Accidents) # เช่น RoadType มีค่า NaN value ทั้งหมด 2312 คิดเป็น % ได้ 2312 / 21031 * 100 = 10.99
```

รูปภาพที่ 7 เช็คข้อมูลที่ว่างหรือขาดหาย

	Total NaN Values	Percentage of NaN Values
RoadType	2312	10.992773
AccidentType	1158	5.505896
Cause	694	3.299734
FirstCar	479	2.277482
Weather	59	0.280525
Province	6	0.028528

รูปภาพที่ 8 ตัวอย่างผลรวมของค่าว่างและเปอร์เซ็นต์ของแต่ละคอลัมน์

```
# ลบแถว (row) ที่มีค่า NaN ออกไปในกรณีที่ที่มีค่าใดค่าหนึ่งใน row นั้นมีค่าเป็น NaN
Accidents.dropna(inplace=True) # inplace=True คือ ทำให้เมธอดนี้เป็นการเปลี่ยนแปลงแทนที่ DataFrame ตัวเก่าด้วยตัวที่ลบ NaN ออกไปแล้วทันที
missing_data(Accidents)
```

รูปภาพที่ 9 ลบข้อแถวที่มีค่าขาดหายหรือค่าว่างออกไป

	Total NaN Values	Percentage of NaN Values
Date	0	0.0
Time	0	0.0
Minor_Injured	0	0.0
Seriously_injured	0	0.0
Dead	0	0.0
Pedestrians	0	0.0
Else	0	0.0
Etan_car	0	0.0

รูปภาพที่ 10 ทำการอัปเดตข้อมูลที่ว่างหลังจากลบแถวที่มีค่าว่างออก

```
# แยก Date ออกมาเป็น Day/Month/Year
Accidents['Date'] = pd.to_datetime(Accidents.Date)
Accidents['year'] = Accidents['Date'].dt.year
Accidents['month'] = Accidents['Date'].dt.month
Accidents['day'] = Accidents['Date'].dt.day
#print(Accidents.day,Accidents.month,Accidents.year)
#Accidents.month
Accidents[['day','month','year']]
```

รูปภาพที่ 11 แยก Date ออกมาเป็น Day/Month/Year

	day	month	year
2	1	1	2022
3	1	1	2022
4	1	1	2022
6	1	1	2022
7	1	1	2022
...
21023	31	12	2022
21027	31	12	2022
21029	31	12	2022
21030	31	12	2022
21031	31	12	2022

16846 rows × 3 columns

รูปภาพที่ 12 ตัวอย่างผลลัพธ์ของ Day/Month/Year

```
# แปลง Datatype ของ column Time (object) >>> Time (datetime64[ns])
t2 = Accidents['Time'].astype(str)
Accidents['Time'] = pd.to_datetime(t2,format='%H:%M:%S')

# แยก Time ออกมาเป็น Hour:Minute:Second
Accidents['Time'] = pd.to_datetime(Accidents.Time)
Accidents['hour'] = Accidents['Time'].dt.hour
Accidents['minute'] = Accidents['Time'].dt.minute
Accidents['second'] = Accidents['Time'].dt.second
#Accidents.hour
Accidents[['hour','minute','second']]
```

รูปภาพที่ 13 แยก Time ออกมาเป็น Hour/Minute/Second

	hour	minute	second
2	0	3	0
3	0	5	0
4	0	5	0
6	0	8	0
7	0	10	0
...
21023	22	55	0
21027	23	11	0
21029	23	30	0
21030	23	45	0
21031	23	55	0

16846 rows × 3 columns

รูปภาพที่ 14 ผลลัพธ์หลังจากแยก Hour/Minute/Second

4.2 Data analysis / Visualization

1. เดือนใดมีความถี่การเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและมีค่าเฉลี่ยเท่าใด

```
# Convert Month number -> Month name
Month_idx = Accidents.groupby(['month'])['month'].count().index.tolist() #เก็บ index เดือน [0,1,2,...,12]
Month_value = Accidents.groupby(['month'])['month'].count().values.tolist() #เก็บความถี่แต่ละเดือน
Month_name = [calendar.month_name[x] for x in Month_idx]

#นำข้อมูลจากข้างบนมาใส่ DataFrame แล้วแสดงผล
df_MonthPerCount = pd.DataFrame({'Month': Month_name, 'Count': Month_value})
df_MonthPerCount = df_MonthPerCount.set_index(['Month'])
df_MonthPerCount['Count'] = df_MonthPerCount['Count'].apply(lambda x: x[0])
#df_MonthPerCount = df_MonthPerCount.reset_index()
# Display result

print("ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = ", df_MonthPerCount['Count'].max())
print("ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = ", df_MonthPerCount['Count'].min())
print("ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = {0:.2f}" .format(df_MonthPerCount['Count'].sum() / len(df_MonthPerCount['Count'])))
df_MonthPerCount
```

รูปภาพที่ 15 วิเคราะห์ข้อมูลระหว่างความถี่ในการเกิดอุบัติเหตุต่อเดือน

ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 1993
 ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 1183
 ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = 1403.83

Count 	
Month	
January	1897
February	1221
March	1284
April	1993
May	1348
June	1183
July	1363
August	1279
September	1224
October	1300
November	1218
December	1536

รูปภาพที่ 16 ผลลัพธ์ความสัมพันธ์ระหว่างความถี่การเกิดอุบัติเหตุต่อเดือน

```
def create_bar_data(col, df): # parameter 2 ตัว คือ column ที่ต้องการและ DataFrame
    aggregated = df[col].value_counts().sort_index()
    x_values = aggregated.index.tolist() # เก็บ index ไว้เป็น list
    y_values = aggregated.values.tolist() # เก็บ values ไว้เป็น list
    return x_values, y_values

x2, y2 = create_bar_data('month', Accidents)
xn = [calendar.month_name[int(x)] for x in x2] # เปลี่ยน month number -> month name
vn = y2
print(xn,vn)
colors = ['#3182bd', '#6baed6', '#9ecae1', '#c6dbef', '#e6550d', '#fd8d3c', '#fdae6b', '#fdd0a2', '#31a354', '#74c476', '#a1d99b', '#c7e9c0']
trace1 = go.Bar(x=xn, y=vn, opacity=0.75, name="month", text = vn, marker=dict(color=colors))
layout = dict(height=400, title='Number of Accidents per Month', xaxis = dict(title = 'Months'), yaxis = dict(title = 'Number of Accidents'));
fig = go.Figure(data=[trace1], layout=layout)
iplot(fig, filename='stacked-bar')
```

รูปภาพที่ 17 เขียนโปรแกรมเพื่อพล็อตกราฟ

2. วันไหนของเดือนที่มีความถี่การเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและมีค่าเฉลี่ยเท่าใด

```
day_count = Accidents.groupby(['day'])[['day']].count()
print("ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = ", Accidents[["day"]].value_counts().max())
print("ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = ", Accidents[["day"]].value_counts().min())
print("ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = {0:.2f}".format(Accidents[["day"]].value_counts().sum()
/ len(Accidents[["day"]].value_counts()))
day_count.T
```

รูปภาพที่ 18 วิเคราะห์ข้อมูลระหว่างความถี่การเกิดอุบัติเหตุต่อวัน

ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 733
 ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 404
 ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = 543.42

day	1	2	3	4	5	6	7	8	9	10	...	22	23	24	25	26	27	28	29	30	31
day	733	658	615	544	475	404	524	512	493	534	...	458	463	472	500	485	463	529	607	618	430

1 rows × 31 columns

รูปภาพที่ 19 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
x_day, y_day = create_bar_data('day', Accidents)
#print(x_day,y_day)
colors = ['#3182bd', '#6baed6', '#9ecae1', '#c6dbef', '#e6550d', '#fd8d3c', '#fdae6b', '#fdd0a2',
          '#31a354', '#74c476', '#a1d99b', '#c7e9c0', '#756bb1', '#9e9ac8', '#bcbddc', '#dadaeb',
          '#636363', '#969696', '#bdbdbd', '#d9d9d9', '#e7e7e7', '#8c6d31', '#bd9e39', '#e7ba52',
          '#f4c471', '#843c39', '#ad494a', '#d6616b', '#e7969c', '#7b4173', '#a55194', '#ce6dbd']
trace1 = go.Bar(x=x_day, y=y_day, opacity=0.75, text=y_day, marker=dict(color=colors))
layout = dict(width = 1300, height=400, title='Number of Accidents per Day',
              xaxis = dict(title = 'Days'), yaxis = dict(title = 'Number of Accidents'));
fig = go.Figure(data=[trace1], layout=layout)
iplot(fig) #filename='stacked-bar'
```

รูปภาพที่ 20 เขียนโปรแกรมเพื่อพล็อตกราฟ

3.เวลาที่ชั่วโมงใดมีความถี่ในการเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและค่าเฉลี่ยเป็นเท่าใด

```
hourly_counts = Accidents.groupby(['hour'])['hour'].count() #สามารถใช้ groupby ไว้จัดหมวดหมู่ได้
print("ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = ", Accidents[["hour"]].value_counts().max())
print("ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = ", Accidents[["hour"]].value_counts().min())
print("ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = {0:.2f}".format(Accidents[["hour"]].value_counts().sum() /
                                                         len(Accidents[["hour"]].value_counts()))))
hourly_counts.T
```

รูปภาพที่ 21 วิเคราะห์ข้อมูลของความสัมพันธ์ระหว่างความถี่ในการเกิดอุบัติเหตุต่อชั่วโมง

```
ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 974
ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 482
ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = 701.92
```

hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21	22	23
hour	550	624	527	482	503	557	550	689	716	774	...	898	974	931	808	709	789	685	611	572	611

1 rows × 24 columns

รูปภาพที่ 22 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```

dates = Accidents['hour']
aggregated = dates.value_counts().sort_index()
x_values = aggregated.index.tolist()
y_values = aggregated.values.tolist()
x1,y1 = x_values, y_values
trace1 = go.Scatter(x=x1, y=y1, mode='lines+markers', opacity=0.75, name="Hour", line=dict(width=2))
layout = dict(height=400, title='Frequency of Accidents per Day', xaxis=dict(title='Hour'), yaxis=dict(title='Number of Accidents'))
fig = go.Figure(data=[trace1], layout=layout)
iplot(fig, filename='line-graph')

```

รูปภาพที่ 23 เขียนโปรแกรมในการพล็อตกราฟ

4.จังหวัดใดมีความถี่การเกิดอุบัติเหตุมากที่สุด 5 อันดับแรก

```

Province_idx = Accidents.groupby(['Province'])['Province'].count().index.tolist()
Province_values = Accidents.groupby(['Province'])['Province'].count().values.tolist()

# combine the two lists
Province_list = list(zip(Province_idx, Province_values))
#print(Province_list)
# sort the list in descending order by Province_values
Province_list_sorted = sorted(Province_list, key=lambda x: x[1][0], reverse=True)

# separate the sorted lists back into Province_idx and Province_values
Province_idx_sorted = [x[0] for x in Province_list_sorted]
Province_values_sorted = [x[1] for x in Province_list_sorted]

# display result
#print(Province_idx_sorted)
#print(Province_values_sorted)

df_pv = pd.DataFrame({"Province_IDX" : Province_idx_sorted,
                      "Province_value" : Province_values_sorted})

df_pv = df_pv.set_index(['Province_IDX'])
df_pv['Province_value'] = df_pv['Province_value'].apply(lambda x: x[0])
df_pv.head(5)

```

รูปภาพที่ 24 วิเคราะห์ข้อมูลความถี่การเกิดอุบัติเหตุของแต่ละจังหวัด แล้วนำมาเรียงลำดับจากมากไปน้อย

Province_value	
Province_IDX	
กรุงเทพมหานคร	1332
ชลบุรี	1056
นครราชสีมา	762
เชียงใหม่	639
สุพรรณบุรี	494

รูปภาพที่ 25 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
print("ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = " , Accidents[["Province"]].value_counts().max())
print("ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = " , Accidents[["Province"]].value_counts().min())
print("ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = {0:.2f}" .format(Accidents[["Province"]].value_counts().sum() / len(Accidents[["Province"]].value_counts())))
```

รูปภาพที่ 26 ทำการหาค่าความถี่ของจังหวัดที่เกิดอุบัติเหตุมากที่สุด น้อยที่สุด และค่าเฉลี่ยของการเกิดอุบัติเหตุ

```
ค่ามากที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 1332
ค่าน้อยที่สุดของจำนวนอุบัติเหตุที่เกิดขึ้น = 35
ค่าเฉลี่ยจำนวนอุบัติเหตุที่เกิดขึ้น = 218.78
```

รูปภาพที่ 27 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
x1, y1 = create_bar_data('Province', Accidents)
trace1 = go.Bar(x=y1, y=x1, opacity=0.75, name="Province", marker=dict(color='#9ecae1'),orientation='h',text = y1)
layout = dict(height=2000, title='Accidents by Provinces');
fig = go.Figure(data=[trace1], layout=layout)
fig.update_layout(barmode='stack', yaxis={'categoryorder':'total ascending'}) # เรียงจากมากไปน้อยจากบนลงล่าง
iplot(fig, filename='stacked-bar')
```

รูปภาพที่ 28 เขียนโปรแกรมในการพล็อตกราฟ

5.ถนนหรือทางแบบใดที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก

```
RoadType_idx = Accidents.groupby(['RoadType'])['RoadType'].count().index.tolist()
RoadType_values = Accidents.groupby(['RoadType'])['RoadType'].count().values.tolist()
RoadType_list = list(zip(RoadType_idx, RoadType_values))
# sort the list in descending order by Province_values
RoadType_list_sorted = sorted(RoadType_list, key=lambda x: x[1][0], reverse=True)
# separate the sorted lists back into Province_idx and Province_values
RoadType_idx_sorted = [x[0] for x in RoadType_list_sorted]
RoadType_values_sorted = [x[1] for x in RoadType_list_sorted]
df_Rt = pd.DataFrame({"RoadType_IDX" : RoadType_idx_sorted,
                      "RoadType_value" : RoadType_values_sorted})
df_Rt = df_Rt.set_index(['RoadType_IDX'])
df_Rt['RoadType_value'] = df_Rt['RoadType_value'].apply(lambda x: x[0])
df_Rt.head(3)
```

รูปภาพที่ 29 วิเคราะห์ข้อมูลว่าระหว่างความถี่ในการเกิดอุบัติเหตุต่อจังหวัด

	RoadType_value
RoadType_IDX	
ทางตรง+ไม่มีความลาดชัน	13503
ทางโค้งกว้าง+ไม่มีความลาดชัน	1974
ทางโค้งกว้าง+ที่ลาดชัน	805

รูปภาพที่ 30 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
x_road, y_road = create_bar_data('RoadType', Accidents)
trace2 = go.Bar(x = y_road, y = x_road, opacity = 0.75, marker=dict(color='ff00ff'),
                name = "RoadType",orientation='h',text = y_road)
layout = dict(height=1000, title='Accidents by Type of Road');
fig = go.Figure(data=[trace2], layout=layout)
fig.update_layout(barmode='stack', yaxis={'categoryorder':'total ascending'})
iplot(fig, filename='stacked-bar')
```

รูปภาพที่ 31 เขียนโปรแกรมเพื่อพล็อตกราฟ

6.ลักษณะการเกิดอุบัติเหตุแบบใดที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก

```
AccidentType_idx = Accidents.groupby(['AccidentType'])['AccidentType'].count().index.tolist()
AccidentType_values = Accidents.groupby(['AccidentType'])['AccidentType'].count().values.tolist()
AccidentType_list = list(zip(AccidentType_idx, AccidentType_values))
AccidentType_list_sorted = sorted(AccidentType_list, key=lambda x: x[1][0], reverse=True)

AccidentType_idx_sorted = [x[0] for x in AccidentType_list_sorted]
AccidentType_values_sorted = [x[1] for x in AccidentType_list_sorted]

# display result
df_At = pd.DataFrame({"AccidentType_IDX": AccidentType_idx_sorted,
                     "AccidentType_value": AccidentType_values_sorted})

df_At = df_At.set_index(['AccidentType_IDX'])
df_At['AccidentType_value'] = df_At['AccidentType_value'].apply(lambda x: x[0])
df_At.head(3)
```

รูปภาพที่ 32 วิเคราะห์ข้อมูลระหว่างความถี่ในลักษณะการเกิดอุบัติเหตุของแต่ละแบบ

AccidentType_value	
AccidentType_IDX	
พลีกคว่า/ตถนนในทางตรง	7804
ชนท้าย	5205
พลีกคว่า/ตถนนในทางโค้ง	2170

รูปภาพที่ 33 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
x3, y3 = create_bar_data('AccidentType', Accidents)
trace3 = go.Bar(x = y3, y = x3, opacity = 0.75, marker=dict(color='#fdae6b'),
               name = "AccidentType",orientation='h',text = y3)
layout = dict(height=1000, title='Accidents by Type of Accidents');
fig = go.Figure(data=[trace3], layout=layout)
fig.update_layout(barmode='stack', yaxis={'categoryorder':'total ascending'})
iplot(fig, filename='stacked-bar')
```

รูปภาพที่ 34 เขียนโปรแกรมเพื่อพล็อตกราฟ

7.เดือนใดที่มีจำนวนผู้เสียชีวิตและผู้บาดเจ็บรวมสูงที่สุด

```
Injury_Death_index = Accidents.groupby(['month'])['Total_Injured','Dead'].sum().index.tolist()
Injury_value = Accidents.groupby(['month'])['Total_Injured'].sum().values.tolist()
Dead_value = Accidents.groupby(['month'])['Dead'].sum().values.tolist()

Months_name = [calendar.month_name[x] for x in Injury_Death_index]
#Months_name
df_Death_Injury = pd.DataFrame({"Injury_Death_IDX":Months_name,
                              "Injury_value":Injury_value,
                              "Dead_value":Dead_value})

df_Death_Injury = df_Death_Injury.set_index(['Injury_Death_IDX'])
df_Death_Injury
```

รูปภาพที่ 35 วิเคราะห์ข้อมูลความถี่ผู้บาดเจ็บ และผู้เสียชีวิตในแต่ละเดือน

	Injury_value	Dead_value
Injury_Death_IDX		
January	1631	229
February	813	146
March	811	145
April	1705	277
May	937	166
June	751	120
July	1073	182
August	948	134
September	804	134
October	902	129
November	923	150
December	1419	196

รูปภาพที่ 36 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```

df = pd.DataFrame(death_by_month).T
df.columns = ['Injuries', 'Dead']

sns.set_style("darkgrid")
sns.set_palette("husl")

ax = df.plot(kind="line", figsize=(25,6), marker='o') #ทำการ plot กราฟเส้นเพื่อดูแนวโน้ม

# ทำการ show values ของผู้บาดเจ็บบนกราฟ
for x,y in zip(range(len(df.index)), df['Injuries']):
    label = "{:.0f}".format(y)
    ax.annotate(label, (x,y), textcoords="offset points", xytext=(0,10), ha='center')

# ทำการ show values ของผู้เสียชีวิตบนกราฟ
for x,y in zip(range(len(df.index)), df['Dead']):
    label = "{:.0f}".format(y)
    ax.annotate(label, (x,y), textcoords="offset points", xytext=(0,10), ha='center')

plt.title("Injuries and Deaths by Month")
plt.xticks(range(len(df.index)), df.index)
plt.xlabel("Month")
plt.ylabel("Number of Occurrences")
plt.show()

```

รูปภาพที่ 37 เขียนโปรแกรมเพื่อพล็อตกราฟ

8.ประเภทของทางบนถนนแบบใดที่มีความถี่ของการเกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับแรกในแต่ละเดือน

```
TypeRoads_index = Accidents.groupby(['month'])['RoadType'].value_counts().index.tolist()
TypeRoads_value = Accidents.groupby(['month'])['RoadType'].value_counts().values.tolist()
df_TypeRoad_Months = pd.DataFrame({"TypeRoad_IDX":TypeRoads_index,
                                   "TypeRoads_Value":TypeRoads_value})
df_TypeRoad_Months
```

รูปภาพที่ 38 วิเคราะห์ข้อมูลความถี่ประเภทของทางบนถนนที่เกิดอุบัติเหตุต่อเดือน

	TypeRoad_IDX	TypeRoads_Value
0	(1, ทางตรง+ไม่มีความลาดชัน)	1511
1	(1, ทางโค้งกว้าง+ไม่มีความลาดชัน)	212
2	(1, ทางโค้งกว้าง+ที่ลาดชัน)	87
3	(1, ทางเชื่อมเข้าพื้นที่สาธารณะ/เชิงพาณิชย์)	42
4	(1, ทางเชื่อมเข้าพื้นที่ส่วนบุคคล)	25
...
106	(12, ทางตรง+ที่ลาดชัน)	11
107	(12, ทางโค้งหักศอก+ที่ลาดชัน)	7
108	(12, ทางเชื่อมเข้าบริเวณหน้าโรงเรียน)	4
109	(12, ทางสามแยก (T))	1
110	(12, ทางโค้งหักศอก+ไม่มีความลาดชัน)	1

111 rows × 2 columns

รูปภาพที่ 39 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
orders = Accidents['RoadType'].value_counts().index
Roadtype_month = sns.catplot(y="RoadType", data=Accidents,
                             kind="count", col= "month",
                             order = orders,height = 8,col_wrap=3);

# ทำการ show values ของ RoadType บนกราฟ
for ax in Roadtype_month.axes.flat:
    for patch in ax.patches:
        x = patch.get_width()
        y = patch.get_y() + patch.get_height() / 2
        ax.annotate(f"{x:.0f}", (x, y), xytext=(5, 0),
                    textcoords='offset points', va='center',fontsize = 16)

plt.show()
```

รูปภาพที่ 40 เขียนโปรแกรมเพื่อพล็อตกราฟ

9. ลักษณะการเกิดอุบัติเหตุแบบใดที่มีความถี่ในการเกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับในแต่ละเดือน

```
AccidentType_index = Accidents.groupby(['month'])['AccidentType'].value_counts().index.tolist()
AccidentType_value = Accidents.groupby(['month'])['AccidentType'].value_counts().values.tolist()
df_AccidentType_Months = pd.DataFrame({"TypeRoad_IDX":AccidentType_index
                                     ,"TypeRoads_Value":AccidentType_value})
df_AccidentType_Months
```

รูปภาพที่ 41 วิเคราะห์ข้อมูลความถี่ของลักษณะการเกิดอุบัติเหตุต่อเดือน

	TypeRoad_IDX	TypeRoads_Value
0	(1, พลิกคว่ำ/ตกถนนในทางตรง)	807
1	(1, ชนท้าย)	621
2	(1, พลิกคว่ำ/ตกถนนในทางโค้ง)	238
3	(1, ชนในทิศทางตรงกันข้าม (ไม่ใช่การแข่ง))	93
4	(1, ชนสิ่งกีดขวาง (บนผิวจราจร))	53
...
98	(12, ชนในทิศทางตรงกันข้าม (ไม่ใช่การแข่ง))	87
99	(12, ชนสิ่งกีดขวาง (บนผิวจราจร))	53
100	(12, ชนคนเดินเท้า)	28
101	(12, อื่นๆ)	23
102	(12, ชนด้านข้าง)	4

103 rows × 2 columns

รูปภาพที่ 42 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```

Acctype_month = sns.catplot(y="AccidentType", data=Accidents, kind="count",
                             col= "month",order=Accidents['AccidentType'].value_counts().index,
                             height = 6,col_wrap=3);

# ทำการ show values ของ AccidentType บนกราฟ
for ax in Acctype_month.axes.flat:
    for patch in ax.patches:
        x = patch.get_width()
        y = patch.get_y() + patch.get_height() / 2
        ax.annotate(f'{x:.0f}', (x, y), xytext=(5, 0), textcoords='offset points',
                    va='center',fontsize = 16)

plt.show()

```

รูปภาพที่ 43 เขียนโปรแกรมเพื่อพล็อตกราฟ

10.เดือนใดที่มีจำนวนรถที่เกิดเหตุทั้งหมดมากที่สุด

```

Car_IncludeFirstCar_index = Accidents.groupby(['month'])['NumOfCar_Include_firstCar','NumOfAccident_Include_pedestrian'].sum().index.tolist()
Car_IncludeFirstCar_value = Accidents.groupby(['month'])['NumOfCar_Include_firstCar'].sum().values.tolist()
Car_IncludePedestrian_value = Accidents.groupby(['month'])['NumOfAccident_Include_pedestrian'].sum().values.tolist()

Months_name = [calendar.month_name[x] for x in Car_IncludeFirstCar_index]
#Months_name
df_Car_Months = pd.DataFrame({"Car_month_IDX":Months_name,
                             "Car_IncludeFirstCar_Value":Car_IncludeFirstCar_value,
                             "Car_IncludePedestrian_Value":Car_IncludePedestrian_value})

df_Car_Months = df_Car_Months.set_index(['Car_month_IDX'])
df_Car_Months

```

รูปภาพที่ 44 วิเคราะห์ข้อมูลความถี่ของจำนวนรถที่เกิดเหตุต่อเดือน

```

df = pd.DataFrame(NumOfCar_by_monthssorted).T
df.columns = ['NumOfCar_Include_firstCar', 'NumOfAccident_Include_pedestrian']

sns.set_style("darkgrid")
sns.set_palette("husl")

ax = df.plot(kind="line", figsize=(25,6), marker='o')

plt.title('NumOfCar_Include_firstCar and NumOfAccident_Include_pedestrian per months')
plt.xticks(range(len(df.index)), df.index)
plt.xlabel('Month')
plt.ylabel('Number of Occurrences')
plt.show()

```

รูปภาพที่ 45 เขียนโปรแกรมเพื่อพล็อตกราฟ

11. จากลักษณะการเกิดอุบัติเหตุแต่ละแบบส่วนใหญ่เกิดกับยานพาหนะประเภทใด

```
vehicles_by_accident_type = Accidents.groupby('AccidentType')['Motorcycle', 'tricycle',
'Public_Cars', 'Van', 'Passenger_pickup_truck', 'Bus_LessThan4wheels',
'PickupTruck_4wheels', 'Truck_6wheels', 'Truck_LessThan10wheels',
'Truck_MoreThan10wheels', 'Etan_car', 'Else'].sum()
vehicles_by_accident_type
```

รูปภาพที่ 46 วิเคราะห์ข้อมูลความถี่ประเภทยานพาหนะต่อลักษณะการเกิดอุบัติเหตุ

	Motorcycle	tricycle	Public_Cars	Van	Passenger_pickup_truck	Bus_LessThan4wheels	PickupTruck_4wheels	Truck_6wheels	Truck_LessThan10wheels	Truck_MoreThan10wheels	Etan_car	Else
AccidentType												
ชนขณะแข่ง	0	0	1	0	0	0	0	0	0	0	0	0
ชนคนเดินเท้า	38	1	48	5	0	0	57	3	4	5	0	15
ชนด้านข้าง	8	0	11	2	0	0	19	2	2	7	0	4
ชนท้าย	2088	78	3912	206	32	0	3808	451	341	894	0	255
ชนสิ่งกีดขวาง (บนผิวจราจร)	246	6	186	16	1	0	223	50	47	104	0	25
ชนเป็นมุมบริเวณทางแยก	3	0	3	1	0	0	4	0	0	1	0	0
ชนในทิศทางตรงกันข้าม (ไม่ใช่การแซง)	305	10	373	23	4	0	631	65	33	87	0	23
พลิกคว่ำ/ตกถนนในทางตรง	514	11	1968	103	16	0	3716	304	236	661	0	432
พลิกคว่ำ/ตกถนนในทางโค้ง	105	3	480	26	2	0	1065	130	73	237	0	70
อื่นๆ	109	3	80	5	3	0	134	17	13	43	0	11
เสียชีวิต/กอลยชน	2	0	1	0	0	0	2	0	0	0	0	0

รูปภาพที่ 47 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
vehicles_by_accident_type = Accidents.groupby('AccidentType')['Motorcycle', 'tricycle',
'Public_Cars', 'Van', 'Passenger_pickup_truck', 'Bus_LessThan4wheels',
'PickupTruck_4wheels', 'Truck_6wheels', 'Truck_LessThan10wheels',
'Truck_MoreThan10wheels', 'Etan_car', 'Else'].sum()

fig = go.Figure()
for column in vehicles_by_accident_type.columns:
    fig.add_trace(go.Bar(
        x=vehicles_by_accident_type.index,
        y=vehicles_by_accident_type[column],
        name=column,
        text=vehicles_by_accident_type[column],
        textposition='auto'
    ))

fig.update_layout(
    width=1300,
    height=650,
    title='Number of Vehicles by Accident Type',
    xaxis_title='Accident Type',
    yaxis_title='Number of Vehicles',
    legend_title='Vehicle Type',
    font=dict(size=16),
    barmode='stack'
)
fig.show()
```

รูปภาพที่ 48 เขียนโปรแกรมเพื่อพล็อตกราฟ

12. ช่วงวันใดควรระมัดระวังการเดินทางโดยพิจารณาจากจำนวนรถที่เกิดเหตุต่อวันและเดือน

```
acc_byDayAndMonth_index = Accidents.groupby(['day', 'month'])['NumOfCar_Include_firstCar'].sum().index.tolist()
acc_byDayAndMonth_value = Accidents.groupby(['day', 'month'])['NumOfCar_Include_firstCar'].sum().values.tolist()

df_acc_Months = pd.DataFrame({"day": [i[0] for i in acc_byDayAndMonth_index],
                              "month": [i[1] for i in acc_byDayAndMonth_index],
                              "acc_byDayAndMonth_Value": acc_byDayAndMonth_value})

df_acc_Months = df_acc_Months.set_index(['day', 'month'])
df_acc_Months
```

รูปภาพที่ 49 วิเคราะห์ข้อมูลความถี่แต่ละวันในแต่ละเดือนต่อจำนวนรถที่เกิดเหตุต่อวันและเดือน

CarAccident_byDayAndMonth_Value		
day	month	
1	1	375
	2	73
	3	79
	4	59
	5	68
...
31	5	77
	7	105
	8	74
	10	57
	12	230

365 rows × 1 columns

รูปภาพที่ 50 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```

# Group the data by day and month and count the number of CarAccidents
accidents_by_hour_and_day = Accidents.groupby(['day', 'month'])['NumOfCar_Include_firstCar'].sum()

# Reshape the data to create a pivot table
accidents_by_hour_and_day = accidents_by_hour_and_day.reset_index()
accidents_by_hour_and_day = accidents_by_hour_and_day.pivot('day', 'month', 'NumOfCar_Include_firstCar')

# Create a heatmap of the number of accidents by day and month
plt.figure(figsize=(15, 10))
sns.heatmap(accidents_by_hour_and_day, cmap='YlGnBu', annot=True, fmt = '.0f')
plt.xlabel('Months')
plt.ylabel('Day')
plt.title('Number of CarAccidents by Day and Month')
plt.show()

```

รูปภาพที่ 51 เขียนโปรแกรมเพื่อพล็อตกราฟ

13.อุบัติเหตุที่เกิดขึ้นในสภาพอากาศแบบใดบ้าง โดยเทียบกับผลรวมของยานพาหนะที่เกิดเหตุ

```

CarAccidents_weather_Idx = Accidents.groupby('Weather')['NumOfCar_Include_firstCar'].sum().index.tolist()
CarAccidents_weather_Value = Accidents.groupby('Weather')['NumOfCar_Include_firstCar'].sum().values.tolist()
df_CarAccidents_weather = pd.DataFrame({"CarAccidents_weather_IDX":CarAccidents_weather_Idx,
                                         "CarAccidents_weather_Values":CarAccidents_weather_Value})
df_CarAccidents_weather

```

รูปภาพที่ 52 วิเคราะห์ข้อมูลความถี่สภาพอากาศต่อผลรวมของยานพาหนะที่เกิดเหตุ

	Weathers_IDX	CarAccidents_Values
0	ฝนตก	3188
1	ภัยธรรมชาติ เช่น พายุ น้ำท่วม	13
2	มีหมอก/ควัน/ฝุ่น	151
3	มีดคร่ำ	5
4	แจ่มใส	21985

รูปภาพที่ 53 ผลลัพธ์ของการวิเคราะห์ข้อมูล

```
CarAcc_Weather = Accidents.groupby('Weather')['NumOfCar_Include_firstCar'].sum().sort_values()
ax = CarAcc_Weather.plot(kind='barh', color='blue', figsize = (15,10))

for i, v in enumerate(CarAcc_Weather):
    #print(i,v)
    ax.text(v + 50, i - 0.15, str(v), color='black', fontsize=16)

ax.set_title("Total of Car Accidents by Weather Condition")
ax.set_xlabel('Total of Car Accidents')
ax.set_ylabel('Weather Condition')

plt.show()
```

รูปภาพที่ 54 เขียนโปรแกรมเพื่อพล็อตกราฟ

14. สัดส่วนความถี่ของผู้ที่บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บ, ไม่เสียชีวิตเลยเป็นอย่างไร

```
No_Deaths = Accidents[Accidents['Dead']<1]
Deaths = Accidents[Accidents['Dead']>=1]
No_Injuries = Accidents[Accidents['Total_Injured']<1]
Injuries = Accidents[Accidents['Total_Injured']>=1]

#คิดความถี่ของผู้เสียชีวิต
print("ไม่มีจำนวนผู้เสียชีวิตเลย = ",No_Deaths.Death.count(),
      "คิดเป็น % เท่ากับ ",No_Deaths.Death.count()/Accidents['Dead'].count() * 100)
print("มีจำนวนผู้เสียชีวิตอย่างน้อย 1 คน = ",Deaths['Dead'].count(),
      "คิดเป็น % เท่ากับ ",Deaths['Dead'].count()/Accidents['Dead'].count() * 100)

print("รวมทั้งไม่มีผู้เสียชีวิตและเสียชีวิต = ",Accidents['Dead'].count())
print("=====")

#คิดความถี่ของผู้บาดเจ็บ
print("ไม่มีจำนวนผู้บาดเจ็บเลย = ",No_Injuries['Total_Injured'].count(),
      "คิดเป็น % เท่ากับ ",No_Injuries['Total_Injured'].count()/Accidents['Total_Injured'].count() * 100)
print("มีจำนวนผู้บาดเจ็บอย่างน้อย 1 คน = ",Injuries['Total_Injured'].count(),
      "คิดเป็น % เท่ากับ ",Injuries['Total_Injured'].count()/Accidents['Total_Injured'].count() * 100)
print("รวมทั้งผู้บาดเจ็บและไม่บาดเจ็บ = ",Accidents['Total_Injured'].count())
```

รูปภาพที่ 55 วิเคราะห์ข้อมูลความถี่ของผู้ที่บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บ และ ไม่เสียชีวิตเลย

```
ไม่มีจำนวนผู้เสียชีวิตเลย = 15141 คิดเป็น % เท่ากับ 89.87890300368039
มีจำนวนผู้เสียชีวิตอย่างน้อย 1 คน = 1705 คิดเป็น % เท่ากับ 10.1210969963196
รวมทั้งไม่มีผู้เสียชีวิตและเสียชีวิต = 16846
=====
ไม่มีจำนวนผู้บาดเจ็บเลย = 9220 คิดเป็น % เท่ากับ 54.73109343464324
มีจำนวนผู้บาดเจ็บอย่างน้อย 1 คน = 7626 คิดเป็น % เท่ากับ 45.26890656535676
รวมทั้งผู้บาดเจ็บและไม่บาดเจ็บ = 16846
```

รูปภาพที่ 56 ผลลัพธ์ของการวิเคราะห์ข้อมูล

15. เทียบสาเหตุการเกิดอุบัติเหตุแต่ละแบบ ว่ามีผลรวมของผู้เสียชีวิตกับผู้บาดเจ็บเป็นอย่างไร

```
idx_Death = Accidents.groupby(['Cause'])['Dead'].sum().index.tolist()
value_Death = Accidents.groupby(['Cause'])['Dead'].sum().values.tolist()

idx_Injury = Injuries.groupby(['Cause'])['Total_Injured'].sum().index.tolist()
value_Injury = Injuries.groupby(['Cause'])['Total_Injured'].sum().values.tolist()

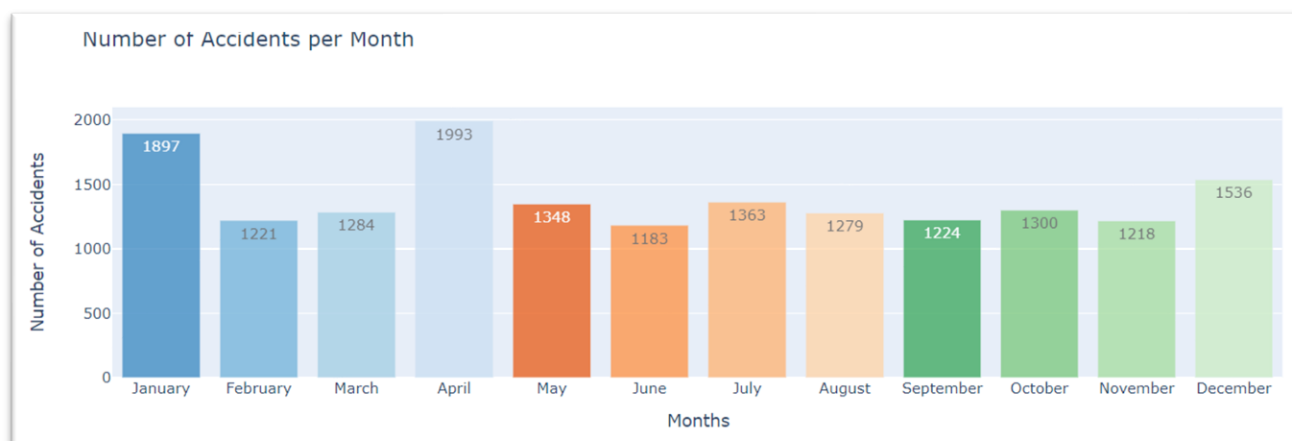
fig = go.Figure()
fig.add_trace(go.Bar(x=idx_Death, y=value_Death,
                    name='Deaths', marker_color='rgb(255, 53, 53)'))
fig.add_trace(go.Bar(x=idx_Injury, y=value_Injury,
                    name='Injuries', marker_color='rgb(26, 118, 255)'))
fig.update_layout(title='Total of Deaths and Injuries Accidents by Cause',
                  xaxis=dict(title='Cause of Accident'),
                  yaxis=dict(title='Total of Dead and Injury'),
                  barmode='stack', width = 2800, height = 700)
fig.show()
```

รูปภาพที่ 57 เขียนโปรแกรมเปรียบเทียบสาเหตุการเกิดอุบัติเหตุเพื่อหาผลรวมผู้เสียชีวิตกับผู้บาดเจ็บ
แล้วนำมาพล็อตกราฟ

บทที่ 4

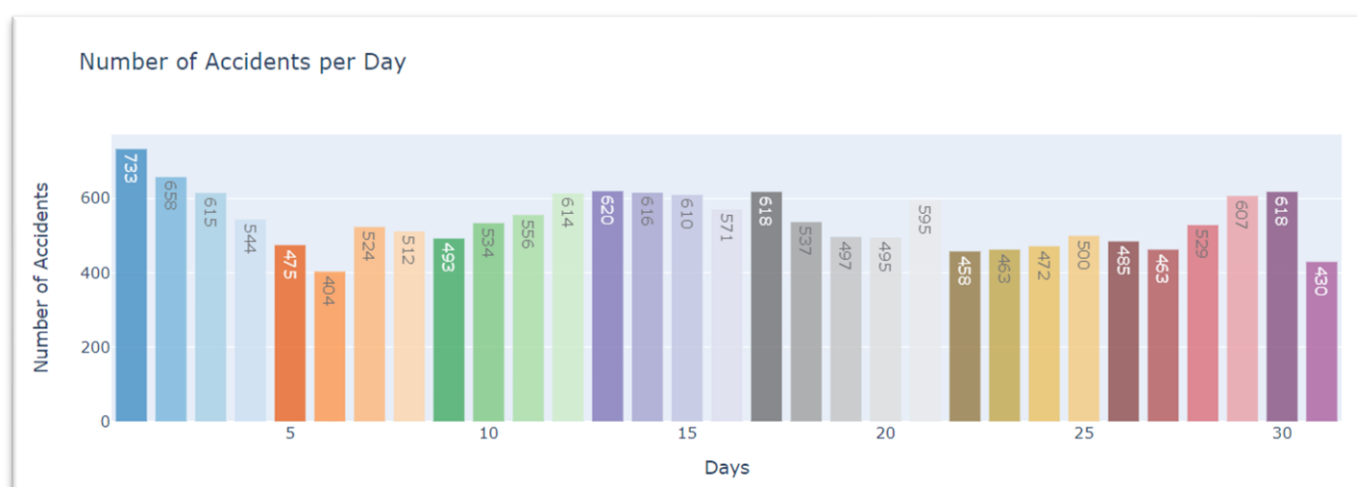
ผลการดำเนินงาน

1. เดือนใดมีความถี่การเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและมีค่าเฉลี่ยเท่าใด



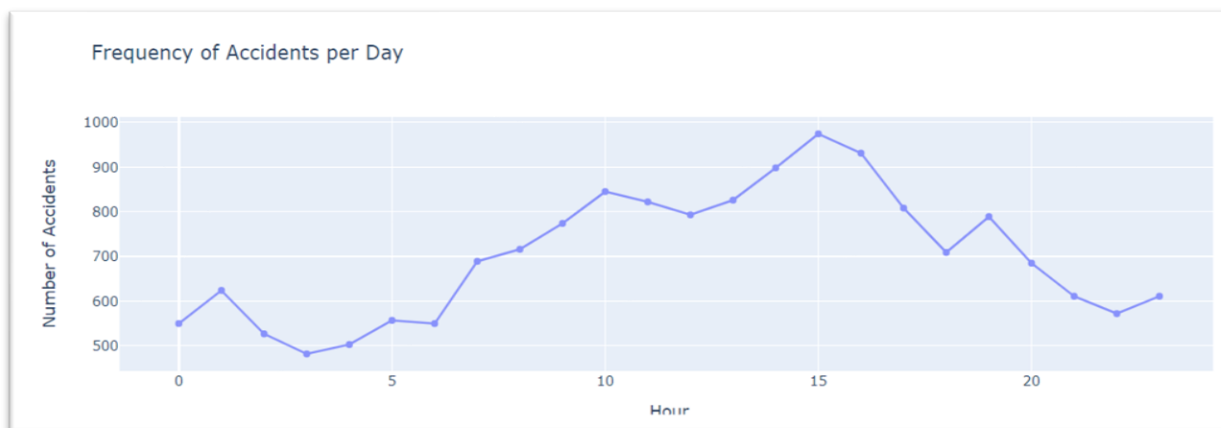
รูปภาพที่ 58 กราฟแสดงความถี่การเกิดอุบัติเหตุต่อเดือน

2. วันไหนของเดือนที่มีความถี่การเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและมีค่าเฉลี่ยเท่าใด



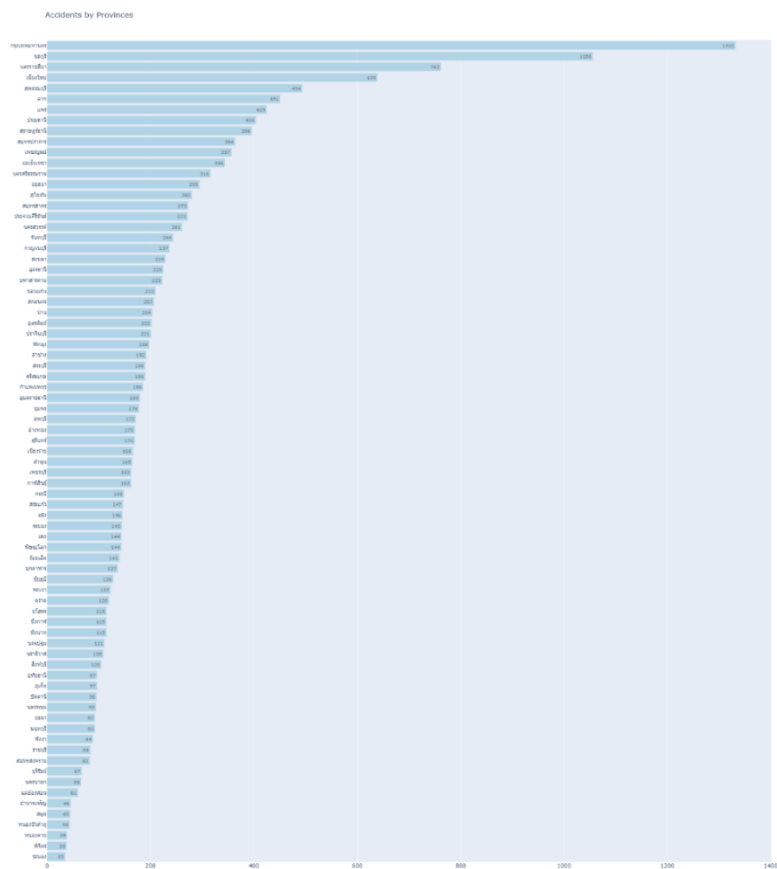
รูปภาพที่ 59 กราฟความสัมพันธ์ระหว่างความถี่ในการเกิดอุบัติเหตุต่อวัน

3.เวลาที่ชั่วโมงใดมีความถี่ในการเกิดอุบัติเหตุมากที่สุดและน้อยที่สุดและค่าเฉลี่ยเป็นเท่าใด

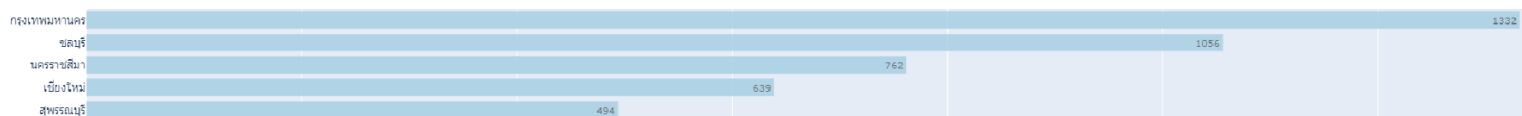


รูปภาพที่ 60 กราฟเส้นแสดงความสัมพันธ์ระหว่างความถี่ในการเกิดอุบัติเหตุต่อชั่วโมง

4.จังหวัดใดมีความถี่การเกิดอุบัติเหตุมากที่สุด 5 อันดับแรก



รูปภาพที่ 61 กราฟแสดงความถี่การเกิดอุบัติเหตุของแต่ละจังหวัดโดยเรียงลำดับจากมากไปน้อย

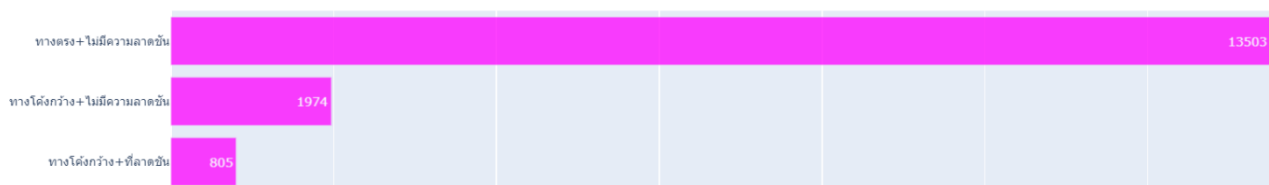


รูปภาพที่ 62 จังหวัดที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 5 อันดับแรก

คือ กรุงเทพมหานคร ชลบุรี นครราชสีมา เชียงใหม่ และ สุพรรณบุรี
โดยมีความถี่เป็น 1332, 1056 762 639 และ 494 ตามลำดับ

5.ถนนหรือทางแบบใดที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก

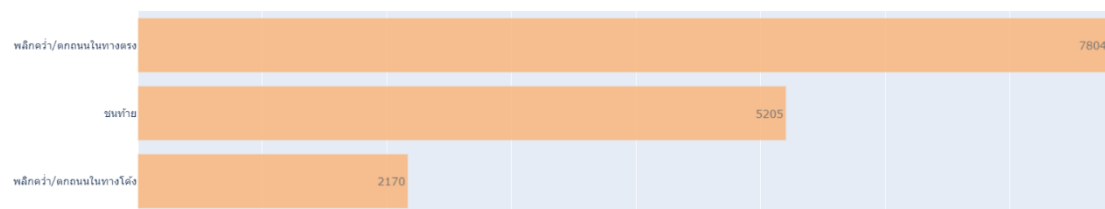
Accidents by Type of Road



รูปภาพที่ 63 จากกราฟจะเห็นว่าถนนหรือทางที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรกได้แก่ ทางตรง+ไม่มีความลาดชัน ทางโค้งกว้าง+ไม่มีความลาดชัน และ ทางโค้งกว้าง+ที่ลาดชัน โดยมีความถี่เป็น 13503, 1974 และ 805 ตามลำดับ

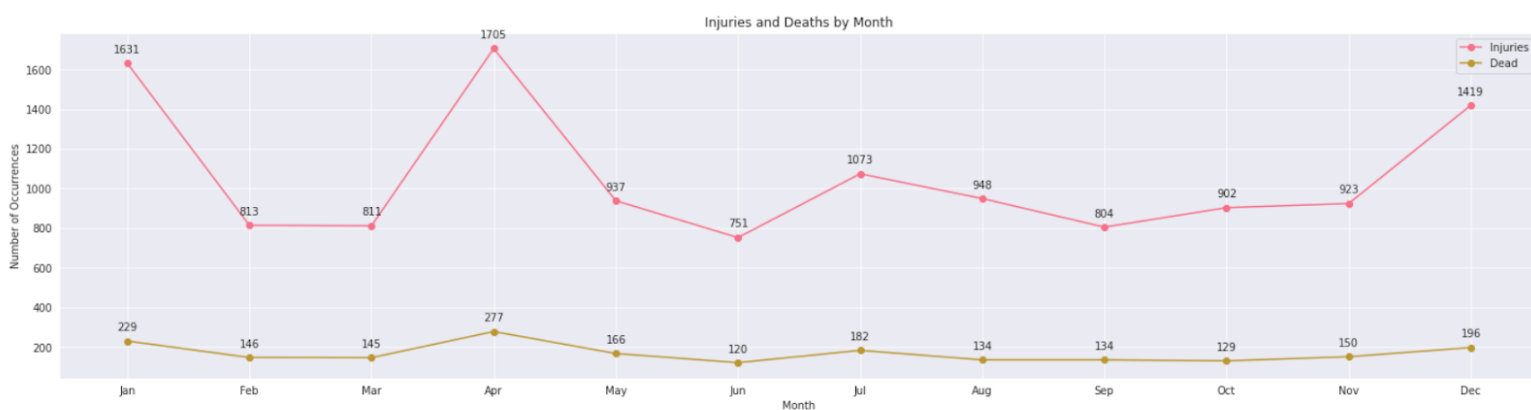
6.ลักษณะการเกิดอุบัติเหตุแบบใดที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรก

Accidents by Type of Accidents



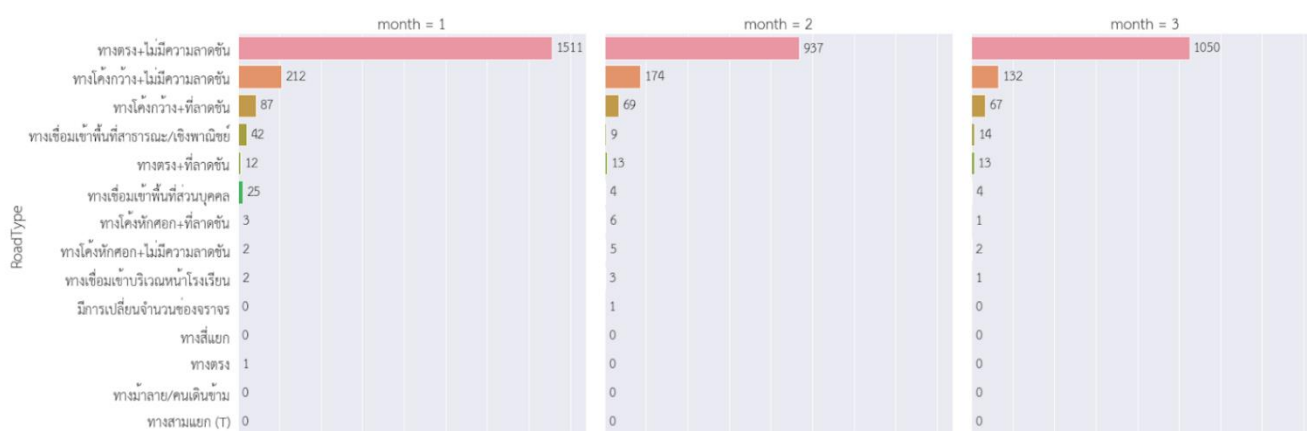
รูปภาพที่ 64 จากกราฟจะเห็นว่าถนนหรือทางที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรกได้แก่ พลิกคว่ำ/ตกถนนในทางตรง ชนท้าย และ พลิกคว่ำ/ตกถนนในทางโค้ง โดยมีความถี่เป็น 7804, 5205 และ 2170 ตามลำดับ

7.เดือนใดที่มีจำนวนผู้เสียชีวิตและผู้บาดเจ็บรวมสูงที่สุด

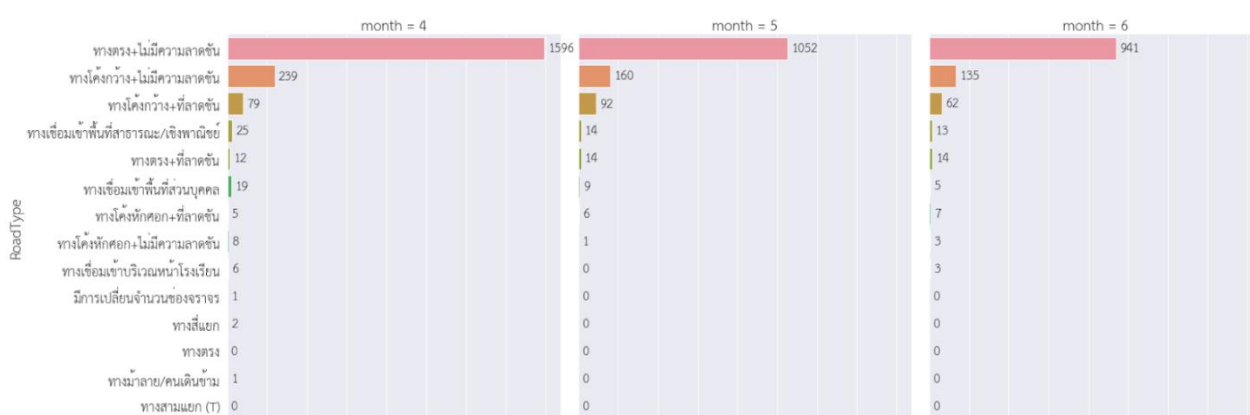


รูปภาพที่ 65 สรุปได้ว่าเดือนเมษายน มียอดผู้เสียชีวิตรวมจากอุบัติเหตุสูงมากที่สุด โดยมียอดผู้เสียชีวิตรวมอยู่ที่ 277 คนและเดือนเมษายน มียอดผู้บาดเจ็บรวมจากอุบัติเหตุมากที่สุดเช่นกัน โดยมียอดผู้บาดเจ็บรวมอยู่ที่ 1705 คน

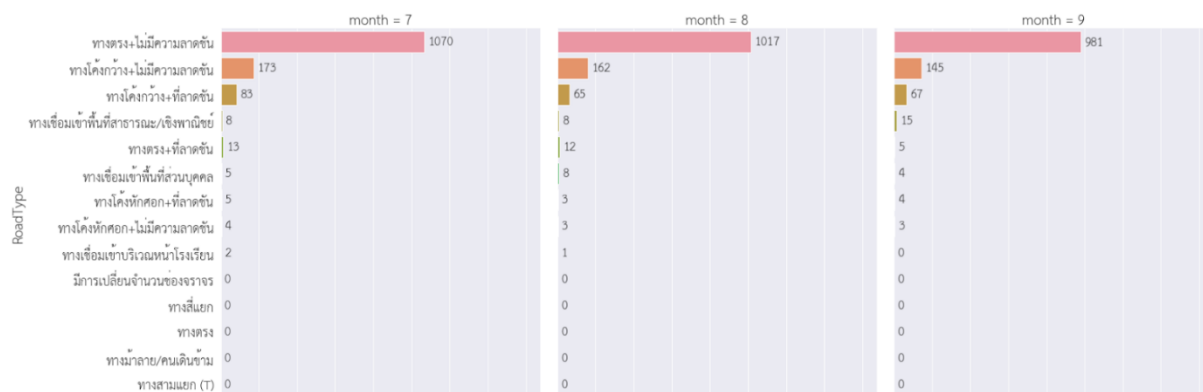
8.ประเภทของทางบนถนนแบบใดที่มีความถี่ของการเกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับแรกในแต่ละเดือน



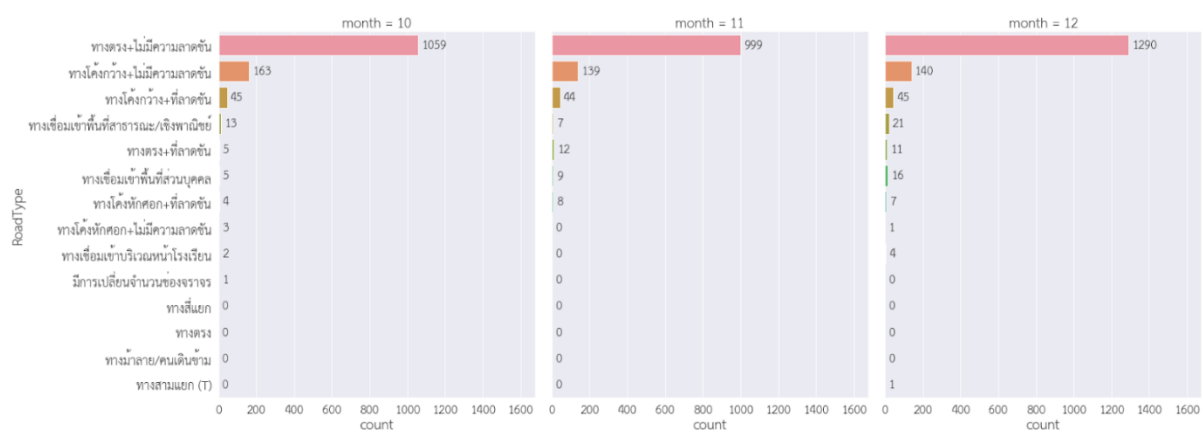
รูปภาพที่ 66 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 1 – 3



รูปภาพที่ 67 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 4 – 6

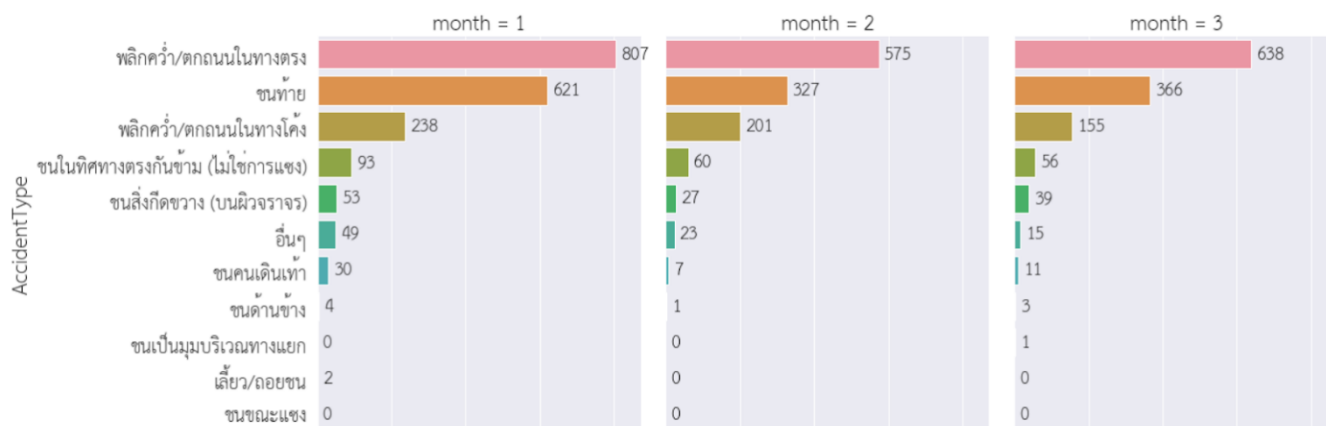


รูปภาพที่ 68 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 7 – 9

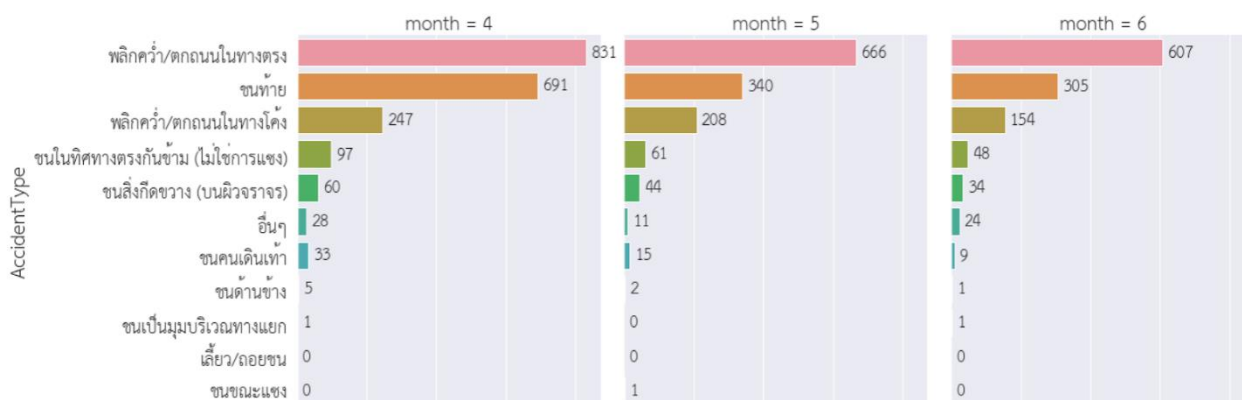


รูปภาพที่ 69 กราฟความถี่ของการเกิดอุบัติเหตุของเดือน 10 – 12

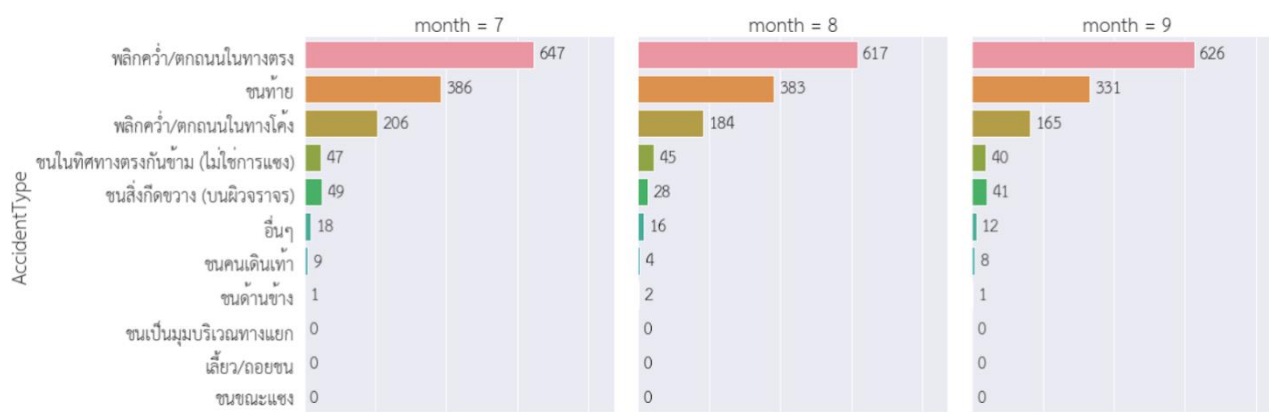
9.ลักษณะการเกิดอุบัติเหตุที่มีความถี่ในการเกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับในแต่ละเดือน



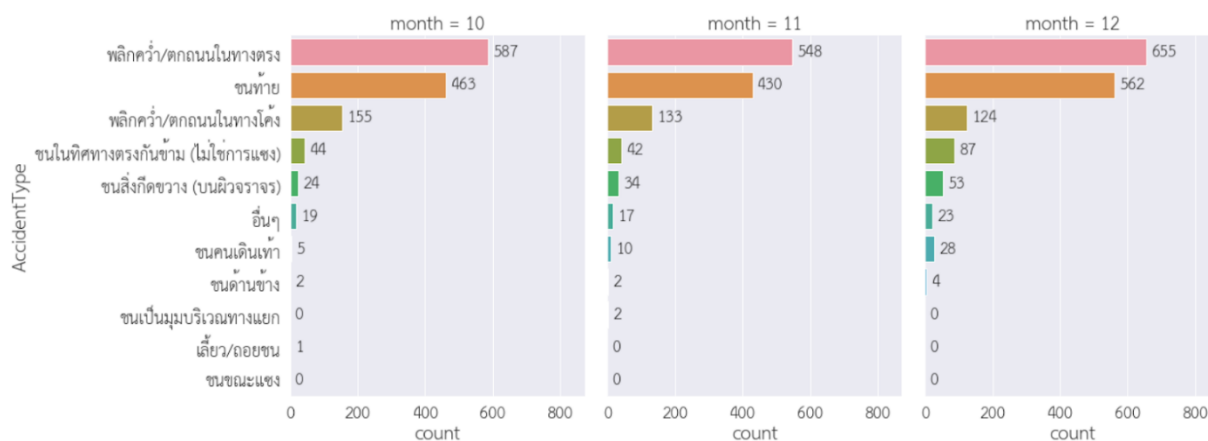
รูปภาพที่ 70 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 1 – 3



รูปภาพที่ 71 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 4 – 6

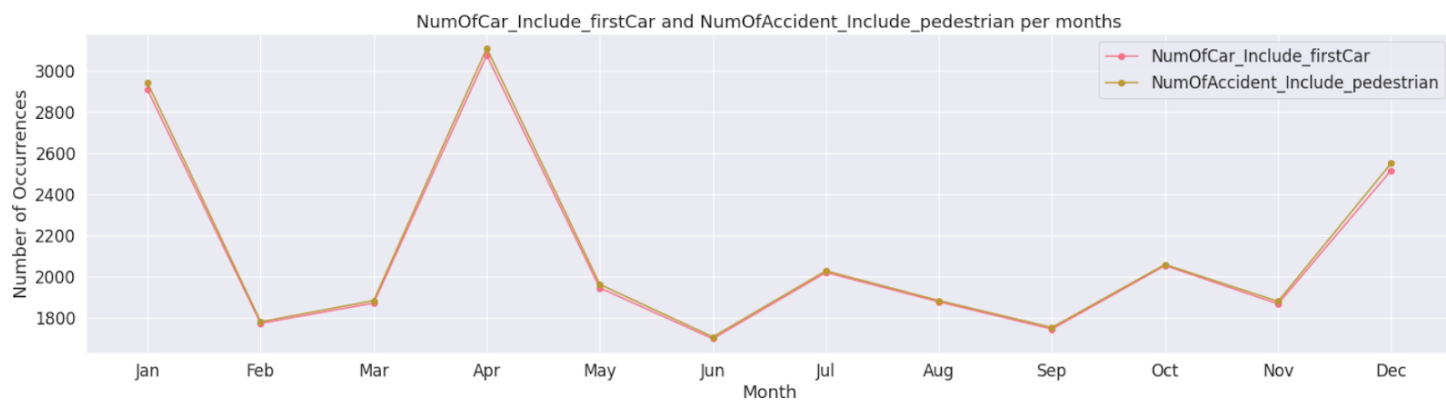


รูปภาพที่ 72 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 7 – 9



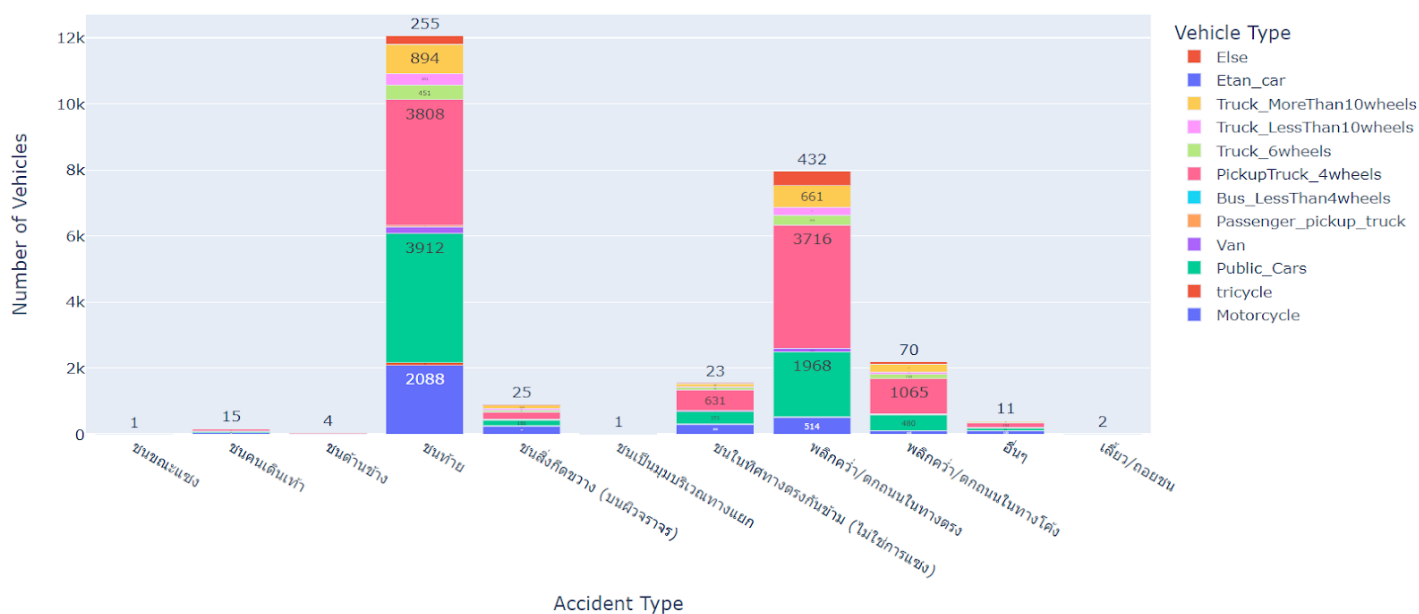
รูปภาพที่ 73 กราฟความถี่ของลักษณะการเกิดอุบัติเหตุในเดือน 10 – 12

10.เดือนใดที่มีจำนวนรถที่เกิดเหตุทั้งหมดมากที่สุด



รูปภาพที่ 74 จากกราฟสรุปได้ว่าเดือนเมษายนมีจำนวนรถที่เกิดเหตุมากที่สุดทั้งแบบรวมคันแรกและแบบรวมคนเดินเท้า

11. จากลักษณะการเกิดอุบัติเหตุแต่ละแบบส่วนใหญ่เกิดกับยานพาหนะประเภทใด

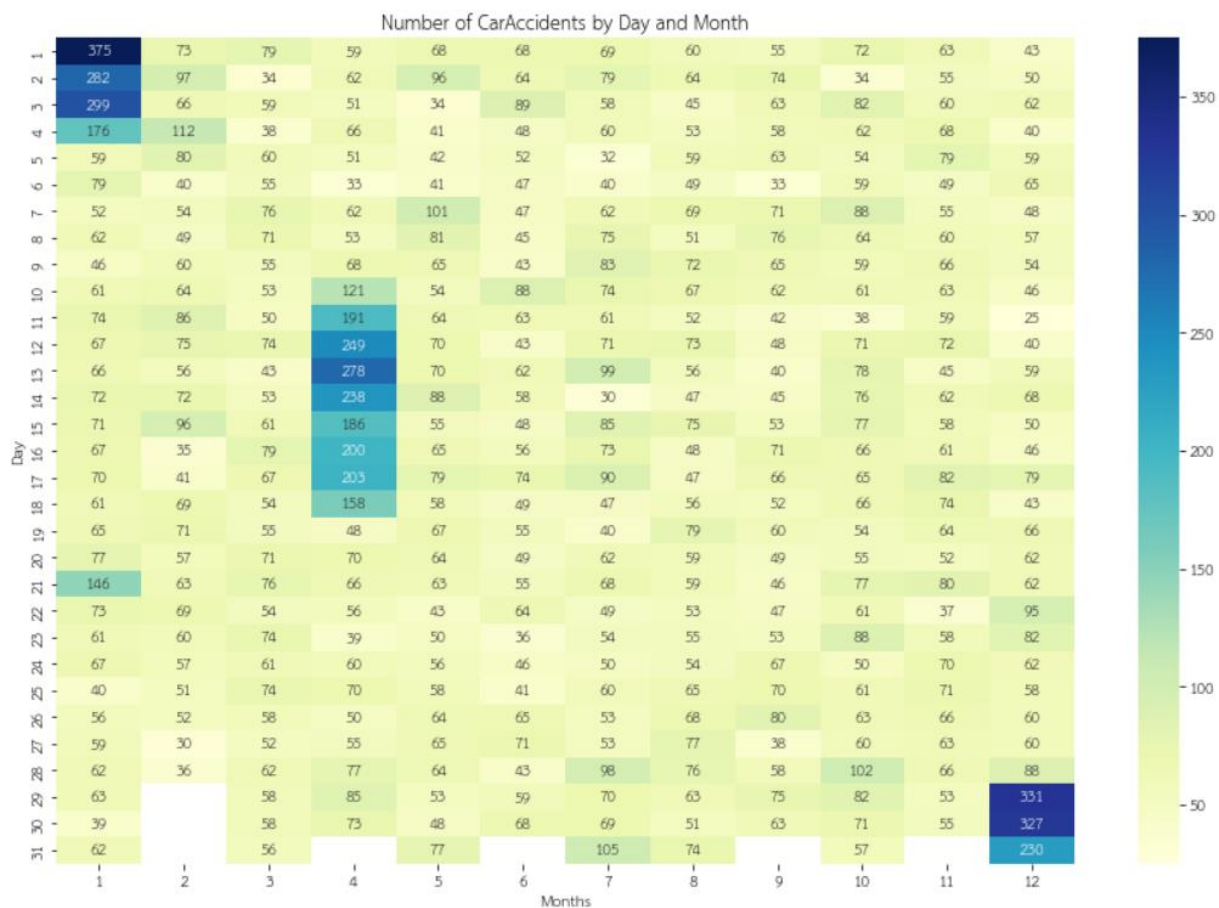


รูปภาพที่ 75 จากกราฟสังเกตได้ว่าลักษณะการเกิดอุบัติเหตุส่วนใหญ่จะเกิดกับยานพาหนะ

ประเภทรถยนต์นั่งส่วนบุคคล/รถยนต์นั่งสาธารณะ และ รถปิคอัพบรรทุก 4 ล้อ

เช่น ลักษณะอุบัติเหตุประเภทชนท้ายเกิดกับยานพาหนะประเภทรถยนต์นั่งส่วนบุคคล/รถยนต์นั่งสาธารณะมากที่สุด ในขณะที่ลักษณะพลิกคว่ำ/ตกถนนในทางตรงจะเกิดกับรถปิคอัพบรรทุก 4 ล้อมากที่สุด

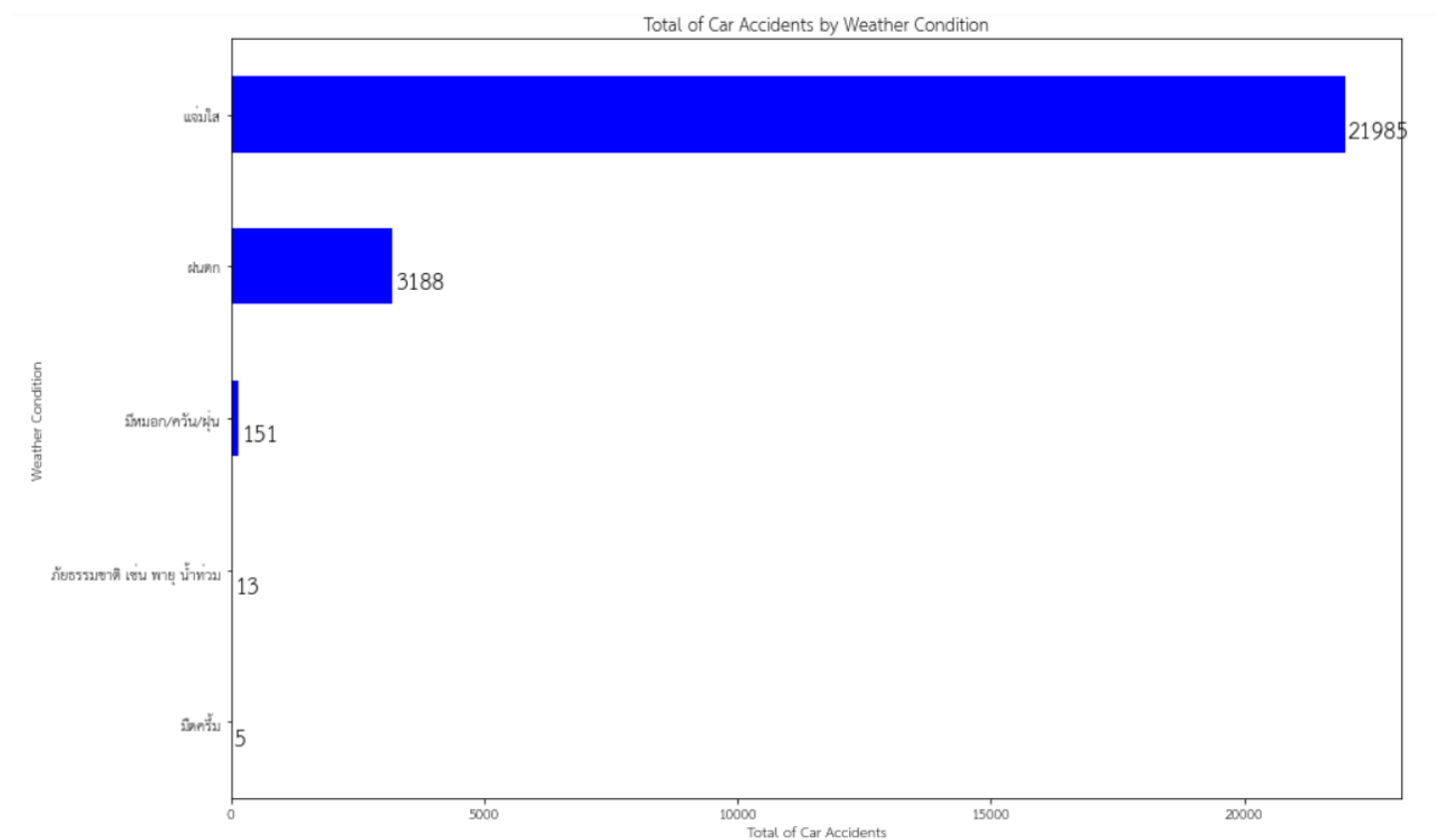
12. ช่วงวันใดควรระมัดระวังการเดินทางโดยพิจารณาจากจำนวนรถที่เกิดเหตุต่อวันและเดือน



รูปภาพที่ 76 จากกราฟแบบ Heatmap สังเกตได้ว่า ช่วงเดือนที่ 1 ในวันที่ 1-4, เดือนที่ 4 วันที่ 10-18 และ เดือนที่ 12 ช่วงวันที่ 29-31 มีจำนวนรถที่เกิดเหตุเป็นจำนวนมากกว่าช่วงอื่นๆ

ดังนั้นสรุปได้ว่าช่วงเวลาข้างต้นมีโอกาสดังกล่าวเกิดอุบัติเหตุเป็นจำนวนมาก ดังนั้นควรระวังช่วงวันดังกล่าว

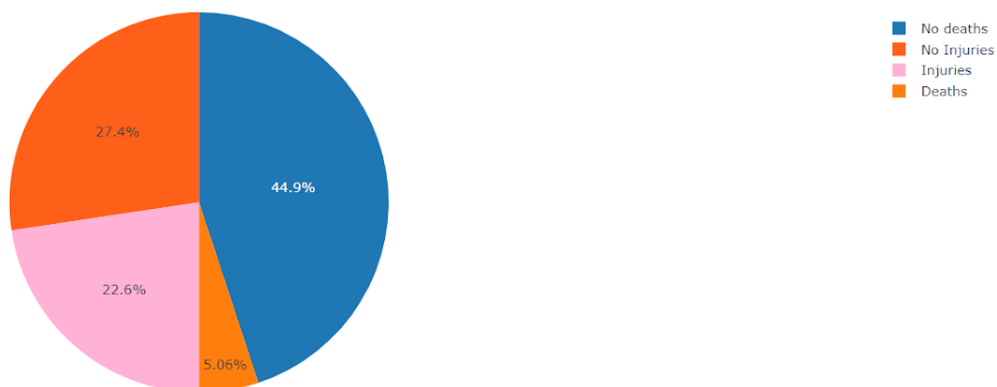
13.อุบัติเหตุที่เกิดในสภาพอากาศแบบใดบ้าง โดยเทียบกับผลรวมของยานพาหนะที่เกิดเหตุ



รูปภาพที่ 77 จากกราฟสรุปได้ว่า จำนวนของรถที่เกิดอุบัติเหตุส่วนใหญ่เกิดในสภาพอากาศที่แจ่มใส
รองลงมาคือ ฝนตก

14. สัดส่วนความถี่ของผู้ที่บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บ, ไม่เสียชีวิตเลยเป็นอย่างไร

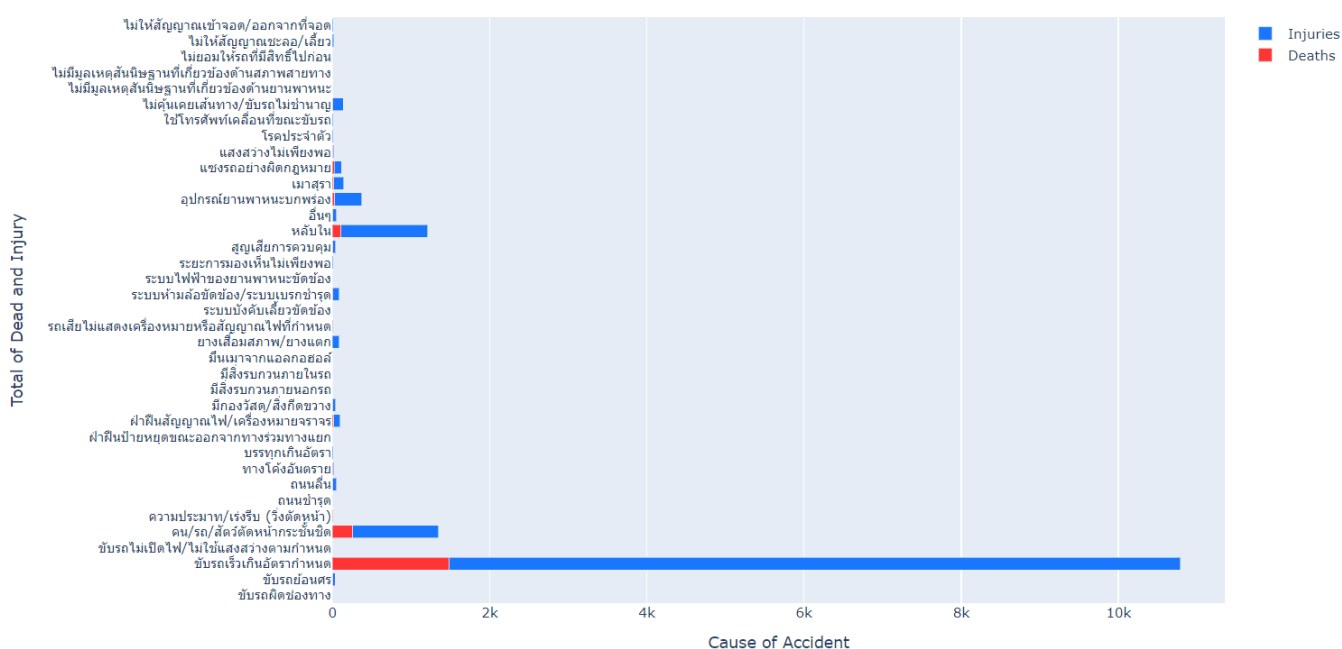
Percentage of Accidents with and Without Deaths and Injuries



รูปภาพที่ 78 สรุปได้ว่าอุบัติเหตุส่วนใหญ่ที่เกิดไม่มีผู้เสียชีวิตถึง 44.9% รองลงมาไม่มีผู้บาดเจ็บถึง 27.1% มีผู้บาดเจ็บ 22.6% และน้อยที่สุดคือมีผู้เสียชีวิต 5.06%

15. เทียบสาเหตุการเกิดอุบัติเหตุแต่ละแบบ ว่ามีผลรวมของผู้เสียชีวิตกับผู้บาดเจ็บเป็นอย่างไร

Total of Deaths and Injuries Accidents by Cause



รูปภาพที่ 79 จากกราฟสรุปได้ว่า สาเหตุของการเกิดอุบัติเหตุส่วนใหญ่มีผู้บาดเจ็บมากกว่าผู้เสียชีวิต

ตัวอย่างเช่น สาเหตุที่เป็นขับรถเร็วเกินกำหนดสรุปได้ว่า มีความถี่ของผู้บาดเจ็บเป็นส่วนใหญ่

รองลงมาคือมีผู้เสียชีวิต

บทที่ 5

สรุปผลการดำเนินงาน และข้อเสนอแนะ

5.1 วัตถุประสงค์ของโครงการ

1. เพื่อหาว่าเดือนใดที่มีอัตราการเกิดอุบัติเหตุมากที่สุดและน้อยที่สุด และมีค่าเฉลี่ยเท่าใด
2. เพื่อหาว่าวันไหนของเดือนที่มีอัตราการเกิดอุบัติเหตุมากที่สุด และน้อยที่สุดมีค่าเฉลี่ยเท่าใด
3. เพื่อหาช่วงเวลาที่เกิดอุบัติเหตุมากที่สุด และน้อยที่สุด
4. เพื่อหา 5 อันดับแรกของจังหวัดที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
5. เพื่อหา 3 อันดับแรกของถนนหรือทางที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
6. เพื่อหา 3 อันดับแรกของลักษณะการเกิดอุบัติเหตุที่มีจำนวนการเกิดอุบัติเหตุมากที่สุด
7. เพื่อหาว่าเดือนใดที่มีจำนวนผู้เสียชีวิต และผู้บาดเจ็บรวมจากอุบัติเหตุที่มากที่สุด
8. เพื่อหาว่า 3 อันดับแรกของประเภทรถที่ทำให้เกิดอุบัติเหตุมากที่สุดในแต่ละเดือน
9. เพื่อหาว่า 3 อันดับของลักษณะการเกิดอุบัติเหตุที่มากที่สุดในแต่ละเดือน
10. เพื่อหาว่าเดือนใดที่มีจำนวนรถที่เกิดอุบัติเหตุมากที่สุด
11. เพื่อหาว่าจากลักษณะการเกิดอุบัติเหตุแต่ละแบบมักเกิดกับยานพาหนะประเภทใด
12. เพื่อหาว่าช่วงวันไหนของเดือนที่ควรระมัดระวังในการเดินทาง
13. เพื่อหาว่าอุบัติเหตุเกิดขึ้นนั้นเกิดในสภาพอากาศลักษณะใดบ้าง
14. เพื่อหาสัดส่วนของอัตราผู้บาดเจ็บอย่างน้อย 1 คน, ผู้เสียชีวิตอย่างน้อย 1 คน, ผู้ที่ไม่บาดเจ็บและไม่เสียชีวิตเลย
15. เพื่อนำมาเปรียบเทียบสาเหตุการเกิดอุบัติเหตุแต่ละแบบว่าผลรวมของผู้เสียชีวิตเป็นอย่างไรมาก

5.2 เครื่องมือที่ใช้ในโครงการ

- Google Colab ใช้ในการเขียนโปรแกรมภาษา Python
- ไลบรารีต่างๆของ Python ตัวอย่างเช่น Numpy matplotlib Pandas เป็นต้น

5.3 สรุปผลการดำเนินงาน

- เดือนเมษายนคือ เดือนที่มีความถี่ของการเกิดอุบัติเหตุมากที่สุด รองลงมาคือ เดือนมกราคม และเดือนธันวาคม โดยมีความถี่คือ 1993,1897 และ 1536
- เดือนมิถุนายนมีความถี่ของการเกิดอุบัติเหตุที่น้อยที่สุด โดยมีความถี่เป็น 1183
- ค่าเฉลี่ยของความถี่ในการเกิดอุบัติเหตุแต่ละเดือนเป็น 1403.83

- วันที่ 1 ของทุกเดือนมีความถี่ในการเกิดอุบัติเหตุมากที่สุด รองลงมาเป็นวันที่ 2 และ 13 โดยมีความถี่เป็น 733, 658 และ 620 ตามลำดับ
- วันที่ 6 ของทุกเดือนมีความถี่ในการเกิดอุบัติเหตุน้อยที่สุด โดยมีความถี่เป็น 404
- ค่าเฉลี่ยของความถี่ในการเกิดอุบัติเหตุแต่ละวันของทุกเดือนเป็น 543.42
- เวลา 15 นาฬิกามีความถี่ของจำนวนการเกิดอุบัติเหตุมากที่สุด โดยมีความถี่เป็น 974
- ช่วงเวลา 3 นาฬิกามีความถี่ของจำนวนการเกิดอุบัติเหตุน้อยที่สุด โดยมีความถี่เป็น 482
- จังหวัดที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 5 อันดับแรก คือกรุงเทพมหานคร ชลบุรี นครราชสีมา เชียงใหม่ และ สุพรรณบุรี โดยมีความถี่เป็น 1332, 1056 762 639 และ 494 ตามลำดับ
- ถนนหรือทางที่มีความถี่ในการเกิดอุบัติเหตุมากที่สุด 3 อันดับแรกได้แก่ 1.ทางตรง+ไม่มีความลาดชัน 2.ทางโค้งกว้าง+ไม่มีความลาดชัน และ 3.ทางโค้งกว้าง+ที่ลาดชันโดยมีความถี่เป็น 13503, 1974 และ 805 ตามลำดับ
- ลักษณะการเกิดอุบัติเหตุที่มีความถี่มากที่สุด 3 อันดับแรกได้แก่ 1.พลิกคว่ำ/ตกถนนในทางตรง 2.ชนท้าย และ 3.พลิกคว่ำ/ตกถนนในทางโค้ง โดยมีความถี่เป็น 7804, 5205 และ 2170 ตามลำดับ
- เดือนเมษายน มียอดผู้เสียชีวิตรวมจากอุบัติเหตุสูงมากที่สุด มียอดผู้เสียชีวิตรวมอยู่ที่ 277 คน
- เดือนเมษายน มียอดผู้บาดเจ็บรวมจากอุบัติเหตุมากที่สุดเช่นกัน โดยมียอดผู้บาดเจ็บรวมอยู่ที่ 1705 คน
- ประเภทของทางบนถนนที่ทำให้เกิดอุบัติเหตุขึ้นมากที่สุด 3 อันดับแรกในแต่ละเดือนนั้นไปในทิศทางเดียวกัน โดย 3 อันดับแรกได้แก่ 1.ทางตรง+ไม่มีความลาดชัน 2.ทางโค้งกว้าง+ไม่มีความลาดชัน 3.ทางโค้งกว้าง+ที่ลาดชัน
- ลักษณะการเกิดอุบัติเหตุที่เกิดขึ้นมากที่สุด 3 อันดับแรกในแต่ละเดือนนั้นไปในทิศทางเดียวกัน โดย 3 อันดับแรกได้แก่ 1.พลิกคว่ำ/ตกถนนในทางตรง 2.ชนท้าย 3.พลิกคว่ำ/ตกถนนในทางโค้ง
- เดือนเมษายนมีจำนวนรถที่เกิดเหตุมากที่สุดทั้งแบบรวมคันแรกและแบบรวมคนเดินเท้า
- สรุปได้ว่าลักษณะการเกิดอุบัติเหตุส่วนใหญ่จะเกิดกับยานพาหนะประเภทรถยนต์นั่งส่วนบุคคล/รถยนต์นั่งสาธารณะ และ รถจักรยานยนต์ทุก 4 ล้อ เช่นลักษณะอุบัติเหตุประเภทชนท้ายเกิดกับยานพาหนะประเภทรถยนต์นั่งส่วนบุคคล/รถยนต์นั่งสาธารณะมากที่สุด ในขณะที่ลักษณะพลิกคว่ำ/ตกถนนในทางตรงจะเกิดกับรถจักรยานยนต์ทุก 4 ล้อ มากที่สุด
- ช่วงเดือนที่ 1 ในวันที่ 1-4, เดือนที่ 4 วันที่ 10-18 และเดือนที่ 12 ช่วงวันที่ 29-31 มีจำนวนรถที่เกิดเหตุเป็นจำนวนมากกว่าช่วงอื่นๆ ดังนั้นสรุปได้ว่าช่วงเวลาข้างต้นมีโอกาสดังกล่าวเกิดอุบัติเหตุเป็นจำนวนมาก ควรระวังช่วงวันดังกล่าว
- สรุปได้ว่า จำนวนของยานพาหนะรวมที่เกิดอุบัติเหตุส่วนใหญ่เกิดในสภาพอากาศที่แจ่มใส รองลงมาคือ ฝนตก และมีหมอก/ควัน/ฝุ่น ตามลำดับ

- สรุปได้ว่า สาเหตุของการเกิดอุบัติเหตุส่วนใหญ่มีผู้บาดเจ็บมากกว่าผู้เสียชีวิต
ตัวอย่างเช่น สาเหตุที่เป็นข้อบกพร่องเกินกำหนดมีความถี่ของผู้บาดเจ็บเป็นส่วนใหญ่องลงมาคือมีผู้เสียชีวิต

5.4 ข้อเสนอแนะ

จากการวิเคราะห์ข้อมูลการเกิดอุบัติเหตุดังกล่าว ทำให้ทราบถึงปัจจัยการเกิดอุบัติเหตุแล้ว
ทั้งนี้ผลลัพธ์ที่ได้เป็นเพียงข้อมูลที่ได้จากการวิเคราะห์ข้อมูลข้างต้น เนื่องจาก อุบัติเหตุอาจจะเกิดตอนไหนก็ได้
ซึ่งนอกเหนือจากปัจจัยของข้อมูลที่มีข้างต้นแล้ว ผู้อ่านควรพึงระมัดระวังในการเดินทางตลอดเวลาไม่ว่าเวลาใด

บรรณานุกรม

- [1] สำนักงานนโยบายและแผนการขนส่งและจราจร สำนักความปลอดภัย (2563). การวิเคราะห์สถานการณ์อุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2562 ค้นเมื่อ 10 กุมภาพันธ์ 2563, จาก [รายงานการวิเคราะห์สถานการณ์อุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2562 \(otg.go.th\)](http://otg.go.th)
- [2] กรมป้องกันและบรรเทาสาธารณภัย (2563). ข้อมูลการเกิดอุบัติเหตุทางถนน ค้นเมื่อ 10 กุมภาพันธ์ 2563, จาก [ข้อมูลการเกิดอุบัติเหตุทางถนน - Open Government Data of Thailand](http://otg.go.th)
- [3] กระทรวงคมนาคม (2563). ข้อมูลอุบัติเหตุบนโครงข่ายถนน ค้นเมื่อ 11 กุมภาพันธ์ 2563, จาก <https://datagov.mot.go.th/dataset/roadaccident>
- [4] สำนักงานนโยบายและแผนการขนส่งและจราจร สำนักความปลอดภัย (2564). การวิเคราะห์สถานการณ์อุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2564 ค้นเมื่อ 11 กุมภาพันธ์ 2563, จาก [รายงานการวิเคราะห์สถานการณ์อุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2564 \(otg.go.th\)](http://otg.go.th)