การวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data Analytic) ของการเคลื่อนที่ของรถ (Vehicle Trajectory) โดยใช้ชุดคำสั่งในภาษา Python

BIG DATA ANALYTIC OF VEHICLE TRAJECTORY USING PYTHON LANGUAGE

จิรกฤต เถกิงผล[1] และ ชนินทร์พล ปถะคามินทร์[2]

[1,2]ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพมหานครฯ

E-mail address: [1]jirakrit.tkp@gmail.com, [2]poom5552010@gmail.com

บทคัดย่อ

โครงงานฉบับนี้มีวัตถุประสงค์เพื่อ (1) ศึกษาการใช้คอมพิวเตอร์โปรแกรมในการวิเคราะห์ข้อมูลขนาดใหญ่ (2) บ่งบอกถึงลักษณะพฤติกรรมการเคลื่อนที่ของรถแต่ละคัน ในช่วงเวลาต่าง ๆ ได้ (3) สร้างชุดคำสั่งอย่างง่ายเพื่อให้วิเคราะห์ข้อมูลขนาดใหญ่ได้อย่างรวดเร็ว รวมถึงแสดงผลข้อมูลความสัมพันธ์เชิงตำแหน่งและเวลาของเหตุการณ์ในรูปแบบกราฟฟิก โดยผู้วิจัยใช้ข้อมูลการเคลื่อนที่ของรถจากการจำลองโดยใช้โปรแกรม VISSIM มาใช้ในการวิเคราะห์ผ่าน Jupyter Notebook ซึ่งเป็นภาษา Python ที่ใช้วิเคราะห์ข้อมูลขนาดใหญ่ ผลการศึกษาทำให้ได้ชุดคำสั่งที่สามารถวิเคราะห์สภาพการจราจรได้ ทั้งในระดับมหภาค และจุลภาค ซึ่งทำให้ทราบถึงพฤติกรรมของการจราจรและพฤติกรรมของยานพาหนะที่ส่งผลให้เกิดกระแสการไหลของการจราจร ณ จุดนั้นๆ

คำสำคัญ: การเคลื่อนที่ของรถ; การเขียนคำสั่งทางคอมพิวเตอร์เพื่อวิเคราะห์ข้อมูล; การนำเสนอโดยใช้กราฟฟิก ; แบบจำลองการเคลื่อนที่ของรถ

Abstract

The objective of this project is to (1) study the use of computer programs in analyzing large-scale data, (2) indicate the movement behavior of each vehicle during different periods, and (3) create a simple command set for fast analysis of large-scale data, as well as displaying the positional and temporal relationships of events in a graphical presentation. The researcher used vehicle movement data obtained from VISSIM simulation and analyzed it through Jupyter Notebook, a Python language used for analyzing large-scale data. The study resulted in a set of commands that can analyze traffic conditions at both the Macroscopic and Microscopic levels, providing an understanding of overall traffic and vehicle behavior that influences the flow.

Key words: Vehicle Trajectory ; Coding for analyze data; Graphical presentation ; Vehicle trajectory simulations

1. บทนำ

ในปัจจุบันข้อมูลขนาดใหญ่ หรือ Big data เป็นที่กล่าวถึงและได้รับความนิยมมากในหมู่ผู้ใช้คอมพิวเตอร์ ไม่ว่าจะเป็นในหมู่โปรแกรมเมอร์ กลุ่มพนักงานบริษัท หรือแม้กระทั่งในองค์กรขนาดใหญ่ของรัฐบาล

ข้อมูลขนาดใหญ่ หรือ Big data นั้นเป็นข้อมูลที่มีขนาดใหญ่มากๆ อาจมีความซับซ้อนหรือไม่ก็ได้ และมีจุดเด่นที่ข้อมูลนั้นมีแนวโน้มที่จะเพิ่มจำนวนมากขึ้นเรื่อยๆ เนื่องจากได้รับข้อมูลจากแหล่งที่มาใหม่มากขึ้นเรื่อยๆตัวข้อมูลขนาดใหญ่ที่กล่าวมานี้เป็นข้อมูลที่มีขนาดใหญ่เกินกว่าที่ซอฟแวร์ทั่วไปในปัจจุบันจะสามารถจัดการได้ จุดเด่นของข้อมูลขนาดใหญ่คือผู้ใช้สามารถนำข้อมูลขนาดใหญ่นี้มาวิเคราะห์เพื่อศึกษาแนวโน้มเหตุการณ์ในอดีตที่ผ่านมา เพื่อนำไปวางแผนและคาดเดาเหตุการณ์ที่อาจจะเกิดขึ้นในอนาคต ส่งผลในการตัดสินใจทำสิ่งต่างๆ ในอนาคตเกิดประสิทธิภาพเพิ่มมากขึ้น

โครงงานนี้จึงมีเป้าหมายเพื่อที่จะใช้คอมพิวเตอร์โปรแกรมเขียนชุดคำสั่งอย่างง่ายเพื่อออกคำสั่งวิเคราะห์ข้อมูล และการแสดงผลโดยใช้กราฟฟิกเพื่อช่วยให้การวิเคราะห์ข้อมูลขนาดใหญ่ได้สะดวกและเห็นภาพมากขึ้นสามารถวิเคราะห์สภาพจราจรในสถานการณ์ต่างๆเช่นการเกิดอุบัติเหตุมีผลต่อการเคลื่อนที่ของรถอย่างไรนำไปสู่การวางแผนพัฒนาการจราจรและการจัดการต่อสถานการณ์ต่าง ๆ ต่อไป

2. วรรณกรรมที่เกี่ยวข้อง

2.1 ความหมายและความสำคัญของ Big Data

Big Data [1] เป็นคำศัพท์ใหม่ที่เพิ่งเริ่มใช้ในช่วงปี 1990 โดยมี John Mashey ผู้ที่ทำให้คำนี้เป็นที่รู้จักขึ้นมา ซึ่งหมายถึงข้อมูลที่มีปริมาณที่ใหญ่มาก โดยที่ซอฟท์แวร์รุ่นเก่าไม่สามารถประมวลผลได้ หรือสามารถประมวลผลได้แต่ใช้เวลานาน โดย Big Data จะมีทั้งข้อมูลที่มีโครงสร้าง กึ่งมีโครงสร้าง และไม่มีโครงสร้าง ขนาดของ Big Data ก็จะมีการเพิ่มขนาดของข้อมูลขึ้นไปเรื่อยๆอยู่ตลอดเวลา Big Data จะต้องมีลักษณะสำคัญ 4V ดังต่อไปนี้ ปริมาณ (Volume), ความหลากหลาย (Variety), ความเร็ว (Velocity), คุณภาพของข้อมูล (Veracity)

2.2 ความสามารถ และข้อมูลที่ได้จากโปรแกรมจำลองการเคลื่อนที่ของรถ VISSIM

PVT VISSIM เป็นโปรแกรมแบบจำลองการเคลื่อนที่ของการจราจรแบบ Microscopic Multi-Modal ความสามารถหลักของทางโปรแกรมมีดังนี้

2.2.1 มีจำลองการจราจรแบบ Microscopic

2.2.2 สามารถจำลองการจราจรได้หลายแบบ เช่น

2.2.3 ความมีปฏิสัมพันธ์ระหว่างยานพาหนะอื่นๆ ในแบบจำลอง

ในส่วนข้อมูลที่ได้จากโปรแกรม VISSIM จะประกอบไปด้วย หมายเลขยานพาหนะ, ช่วงเวลา, ความเร็ว, ความเร่ง, พิกัดของยานพาหนะ, หมายเลขช่องจราจร, หมายเลขถนน, ขนาดของยานพาหนะ

2.3 ความหมายของ Vehicle trajectory simulation

Vehicle trajectory simulation หรือการจำลองการเคลื่อนที่ของยานพาหนะ หมายถึงการสร้างแบบจำลองสถานการณ์การการเคลื่อนที่ของยานพาหนะในเวลาต่าง ๆ สามารถจำลองตำแหน่งของยานพาหนะ ความเร็ว ความเร่ง และไดนามิกอื่นๆ ขณะเดินทางบนถนนหรือทางหลวง นอกจากนั้นการจำลองสามารถคำนึงถึงปัจจัยต่าง ๆ เช่น การจราจร สภาพอากาศ สภาพถนน และพฤติกรรมของยานพาหนะคันอื่น ๆ

2.4 การเขียนชุดคำสั่งคอมพิวเตอร์ภาษา Python เพื่อทำการวิเคราะห์ ข้อมูลและแสดงผลด้วยโปรแกรม Jupyter notebook

Jupyter Notebook [2] คือเครื่องมือที่ใช้เขียน code เป็นที่นิยมมากในกลุ่มคนที่ทำงานด้าน Data Science เหมาะกับการทำงานเกี่ยวกับการจัดการข้อมูลเป็นจำนวนมาก พร้อมทำการรายงานผลที่ได้ดำเนินการ เนื่องจากมี Library ที่สำคัญในการจัดการข้อมูล

2.5 การวิเคราะห์การเคลื่อนที่ของยานพาหนะ เพื่อความปลอดภัยบนถนน

มีงานวิจัยที่เกี่ยวข้องเป็นของ Bernard Jacob และ Eric Violette [4] ซึ่งศึกษาประโยชน์ของการวิเคราะห์การเคลื่อนที่ของยานพาหนะ, ข้อจำกัดและมุมมองในการพัฒนา ภายในงานวิจัยได้มีกรณีศึกษา เช่นศึกษาผลกระทบจาก Infrastructure ในฝรั่งเศสในปีค.ศ. 2011 จมีการเก็บข้อมูลการจราจรด้วยกล้องวิดีโอ แล้วนำไปวิเคราะห์ จากการวิเคราะห์พบว่ามีการเคลื่อนที่ที่ไม่พึงประสงค์อยู่ จึงเกิดการตั้งคำถามต่อ infrastructure ว่ามีส่วนทำให้เกิดพฤติกรรมนั้นหรือไม่ และอย่างไร ซึ่งนำไปสู่การแก้ไขในภายหลัง

จากกรณีศึกษาพบว่าการวิเคราะห์การเคลื่อนที่ของยานพาหนะทำให้ได้ข้อมูลที่มีรายละเอียดเฉพาะต่อกรณีนั้น ๆ ซึ่งทำให้แก้ไขปัญหาได้อย่างตรงจุดมากขึ้น

2.6 การประเมินความเสี่ยงต่อการชนท้าย โดยใช้ข้อมูลการเคลื่อนที่ของยานพาหนะ

งานวิจัยของ Cheol Oh, Taejin Kim [5] โดยข้อมูลการเคลื่อนที่ของยานพาหนะในงานวิจัยนี้ได้มาจากซอฟต์แวร์ Premier Pro 2.0 โดยดึงทุก ๆ 1/10 วินาที

ผลสรุปของวิจัยเป็นระบบเฝ้าระวังการจราจร ตรวจจับและติดตามยานพาหนะได้จากวิดีโอ ซึ่งช่วยให้วิเคราะห์สภาพการไหลของจราจรได้ละเอียดมากขึ้น การศึกษานี้เสนอวิธีประเมินความเสี่ยงต่อการชนท้ายแบบเรียลไทม์ สามายานพาหนะรับรู้สภาพกระแสการจราจรที่เป็นอันตราย นอกจากนี้ยังมีการวิจัยเพิ่มเติมเกี่ยวกับความเสี่ยงในการเปลี่ยนเลนเพื่อมาประกอบการคำนวณ

แต่ถึงอย่างนั้นในส่วนท้ายของงานวิจัยได้ให้ข้อเสนอว่าการสร้างแบบจำลองเพื่อให้เชื่อถือได้มากขึ้นควรใช้ชุดข้อมูลที่ใหญ่ขึ้นในการสร้างแบบจำลอง

3. ระเบียบวิธีวิจัย

3.2.1 ศึกษาค้นคว้าทฤษฎีต่าง ๆ และงานวิจัยที่เกี่ยวข้อง

3.2.2 ศึกษาการใช้โปรแกรมคอมพิวเตอร์

3.2.3 พิจารณาข้อมูล และจัดการข้อมูล

3.2.4 ดำเนินการเขียนโปรแกรมเพื่อวิเคราะห์ข้อมูล

3.2.5 ดำเนินการเขียนโปรแกรมเพื่อแสดงผลในรูปแบบ กราฟฟิก

3.2.6 สรุปและอภิปรายผล

4. ผลการดำเนินงานวิจัย

4.1 ผลลัพธ์ในการวิเคราะห์ในระดับมหภาค (Macroscopic)

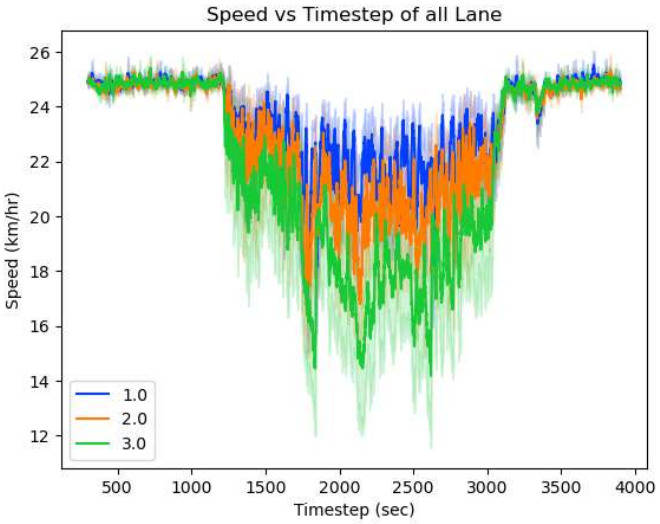
4.1.1 การแสดงผลความเร็วของยานพาหนะ

การแสดงผลความเร็ว ณ เวลาต่างๆ

สามารถใช้คำสั่ง

Trafficspeed_per_time_alllane(data_name)

ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ data_name คือชื่อของข้อมูลการเคลื่อนที่ของยานพาหนะที่สนใจ หลักการทำงานคือนำข้อมูลมาทำการแสดงผลกราฟฟิกแบบเส้นระหว่างความสัมพันธ์ของความเร็วของยานพาหนะกับ เวลาต่างๆ



รูปที่ 4.1.1.1.2 ผลลัพธ์จากคำสั่ง

Trafficspeed_per_time_alllane("CAV50Incident.csv")

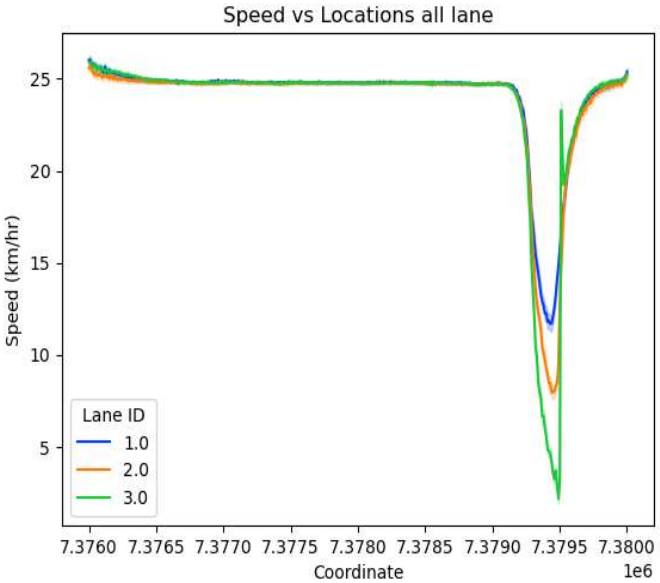
สามารถวิเคราะห์ความเร็วการเคลื่อนที่ของยานพาหนะ ณ เวลาต่างๆ โดยสังเกต ความสัมพันธ์ความเร็ว(แสดงในแกนตั้ง)และช่วงเวลา(แสดงในแกนนอน)

การแสดงผลความเร็ว ณ ตำแหน่งต่างๆ

สามารถใช้คำสั่ง

Trafficspeed_per_section_alllane(data_name,time_s,time_f)

ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ data_name คือชื่อของข้อมูลการเคลื่อนที่ของยานพาหนะ ที่สนใจ, time_s คือเวลาเริ่มต้น, time_f คือเวลาสิ้นสุดที่สนใจ หลักการทำงาน คือ ทำการกรองข้อมูลให้เหลือเฉพาะช่วงเวลาที่สนใจและทำการแสดงผลกราฟฟิกแบบเส้นระหว่างความสัมพันธ์ของความเร็วเฉลี่ยทุกช่วงเวลาที่สนใจกับตำแหน่งต่างๆ



รูปที่ 4.1.1.2.2 ผลลัพธ์จากคำสั่ง

Trafficspeed_per_section(CAV50Incident.csv,1500,3000)

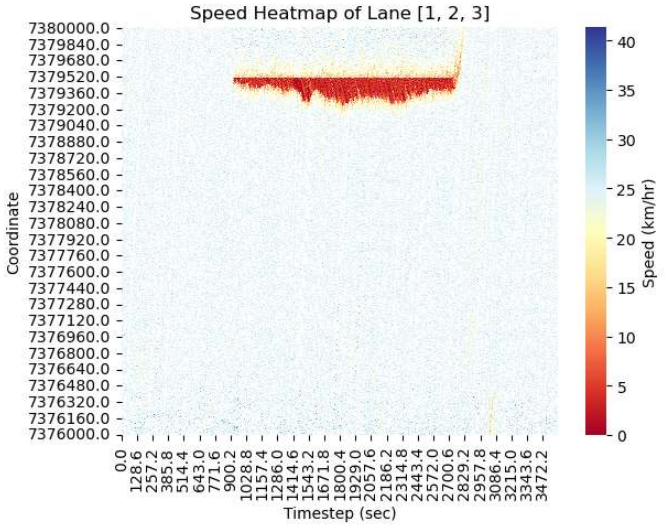
สามารถวิเคราะห์ความเร็วการเคลื่อนที่ยานพาหนะ ณ ตำแหน่งต่างๆ โดยสังเกตความสัมพันธ์ ความเร็ว(แสดงในแกนตั้ง) และตำแหน่ง(แสดงในแกนนอน)

การแสดงผลความเร็ว ณ ตำแหน่งและเวลาต่างๆ

สามารถใช้คำสั่ง

heatmap_trafficspeed(data_name,lane_id)

ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ data_name คือชื่อของข้อมูลการเคลื่อนที่ของยานพาหนะที่สนใจ ,lane_id คือหมายเลขช่องจราจรที่สนใจ โดยจะกรอกเป็นรูปแบบของ หลักการทำงาน คือ ทำการกรองข้อมูลให้เหลือเฉพาะช่องจราจรที่สนใจ จากนั้นแสดงผลแบบ 3 ตัวแปรขึ้นมาและทำการแสดงผล กราฟฟิก แบบแผนภาพความร้อน(Heatmap)ระหว่างความสัมพันธ์ของความเร็วกับช่วงเวลาและตำแหน่งต่างๆ



รูปที่ 4.1.1.3.2 ผลลัพธ์จากคำสั่ง

heatmap_trafficspeed("CAV50Incident.csv",[1,2,3])

สามารถวิเคราะห์ความเร็วการเคลื่อนที่ยานพาหนะ ณ ช่วงเวลาและตำแหน่งต่างๆ โดยสังเกตุความสัมพันธ์ความเร็ว(แสดงในรูปแบบของสี ซึ่งจะมีความสัมพันธ์ของสีและค่าความเร็วตามที่แสดงในหลอดด้านขวาของแผนภาพ) กับช่วงเวลา(แสดงในแกนนอน) และตำแหน่ง(แสดงในแกนตั้ง)

4.1.2 การแสดงผลความหนาแน่นของยานพาหนะ ณ เวลาต่างๆ

จะต้องมีการเตรียมข้อมูลผ่านทางคำสั่ง

Density_period_alllane(data_name,t1,period,t2,l1,l2) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ data_name คือชื่อของข้อมูลการเคลื่อนที่ของยานพาหนะที่สนใจ, t1 คือเวลาเริ่มต้น, period คือความกว้างของช่วงระยะเวลา, t2 คือเวลาสิ้นสุด, l1 คือช่วงพิกัดของถนนเริ่มต้น และ l2 คือช่วงพิกัดของถนนสิ้นสุด หลักการทำงานของคำสั่งชุดนี้คือ ทำการคำนวณความหนาแน่นของยานพาหนะในช่วงพิกัดถนนที่สนใจ โดยจะคำนวนเก็บเป็นข้อมูล Dataframe ตั้งแต่เวลาเริ่มต้นที่สนใจ จนถึงเวลาสิ้นสุดที่สนใจ โดยชุดคำสั่งจะมีการคำนวณสมการในการหาความหนาแน่นผ่านสมการ

$$D = \frac{m}{l} \times 1{,}000 \qquad (4.1.2)$$

D หมายถึง ความหนาแน่นของยานพาหนะ

m หมายถึง จำนวนยานพาหนะในช่วงพิกัดที่สนใจ

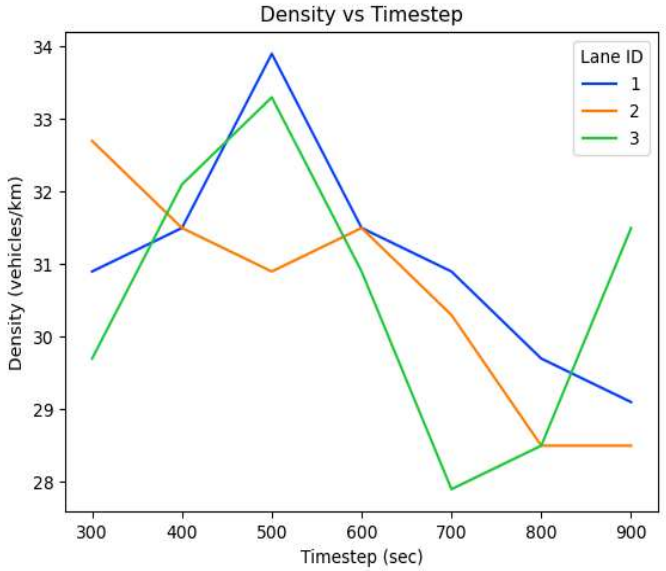
l หมายถึง ความยาวของถนนในพิกัดที่สนใจ

การแสดงผลความหนาแน่น ณ เวลาต่างๆ

สามารถใช้คำสั่ง

density_plot_alllane(data_name,t1,period,t2,l1,l2)

ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ data_name คือชื่อของข้อมูลการเคลื่อนที่ของยานพาหนะที่สนใจ, t1 คือเวลาเริ่มต้น, period คือความกว้างของช่วงระยะเวลา, t2 คือเวลาสิ้นสุด, l1 คือช่วงพิกัดของถนนเริ่มต้น และ l2 คือช่วงพิกัดของถนนสิ้นสุดที่ หลักการทำงานคือนำข้อมูลการเลื่อนที่ของยานพาหนะที่ยังไม่ทำการวิเคราะห์ความหนาแน่น ไปคำนวณความหนาแน่นผ่านทางชุดคำสั่งในรูปที่ 4.1.2.0 หลังจากนั้นนำข้อมูลที่ผ่านการประมลผล แล้วทำการแสดงผลกราฟฟิกแบบเส้นระหว่างความสัมพันธ์ของความหนาแน่นกับเวลาต่างๆ



รูปที่ 4.1.2.2 ผลลัพธ์จากคำสั่ง

density_plot_alllane('CAV50Incident.csv',300,100,900,7378000,7379650)

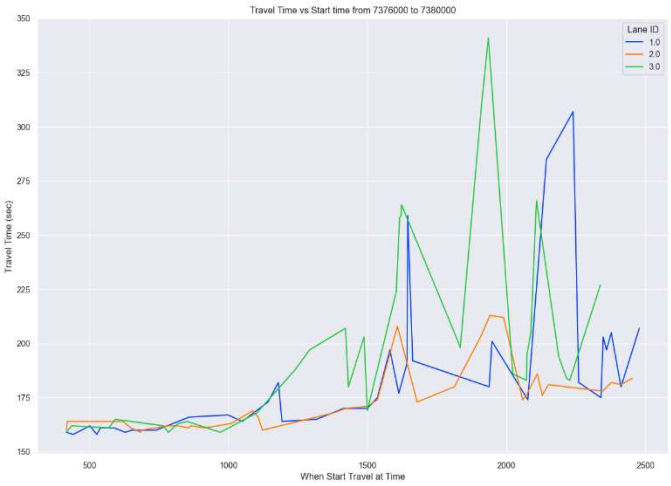
สามารถวิเคราะห์ความหนาแน่นการเคลื่อนที่ยานพาหนะ ณ ช่วงเวลาต่างๆ โดยสังเกตุความสัมพันธ์ ความหนาแน่น(แสดงในแกนตั้ง) กับช่วงเวลา(แสดงในแกนนอน)

4.1.3 การแสดงผลระยะเวลาที่ใช้เดินทางของยานพาหนะเมื่อเริ่มออกเดินทาง ณ เวลาต่างๆ จากจุดที่สนใจจุดแรกไปยังจุดที่สนใจจุดที่สอง

สามารถใช้คำสั่ง

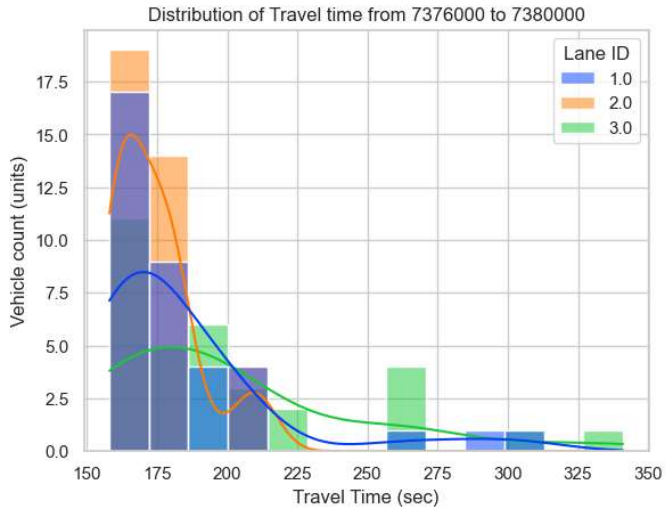
Travel_time_from_atob(data_name,t1,t2,l1,l2)

ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ data_name คือชื่อของข้อมูลการเคลื่อนที่ของยานพาหนะที่สนใจ, t1 คือเวลาเริ่มต้น, period คือความกว้างของ ช่วงระยะเวลาที่, t2 คือเวลาสิ้นสุด, l1 คือช่วงพิกัดของถนนเริ่มต้น และ l2 คือช่วงพิกัดของถนนสิ้นสุด หลักการทำงานคือกรองข้อมูลให้เหลือเพียงช่วงเวลาที่สนใจ และทำการเก็บรายชื่อ ยานพาหนะที่ผ่านจุดพิกัดสิ้นสุดที่สนใจ (l2) และทำการตรวจสอบย้อนกลับ ไปยังจุดพิกัดเริ่มต้นที่สนใจ (l1) หากรายชื่อยานพาหนะปรากฏทั้งในพิกัด l1 และ l2 จะทำการบันทึกรายชื่อยานพาหนะนั้นไปยังชุดข้อมูลใหม่ โดยจะทำการบันทึกชื่อยานพาหนะ, ช่วงเวลาที่ผ่านพิกัด l1, ช่วงเวลาที่ผ่านพิกัด l2 และ หมายเลขช่องจราจร และทำการคำนวณหาระยะเวลาที่ใช้เดินทางของยานพาหนะจากผลต่างของช่วงเวลาที่ผ่านพิกัด l1 และพิกัด l2



รูปที่ 4.1.3.2 ผลลัพธ์จากคำสั่ง Travel_time_from_atob('CAV50Incident.csv',400,2700,7376000,7380000)

นอกจากนี้ยังสามารถเลือกให้มีการแสดงผลถึงการกระจายของระยะเวลาที่ใช้เดินทางของยานพาหนะ โดยใช้คำสั่ง dis_travel_time(data_name,t1,t2,l1,l2) โดยที่พารามิเตอร์ที่ทำการใส่เข้าไปจะเป็นพารามิเตอร์เดียวกับคำสั่ง Travel_time_from_atob(data_name,t1,t2,l1,l2) โดยที่หลักการทำงานของคำสั่งนี้จะเพิ่ม การแสดงผลข้อมูลแบบ กราฟฟิกการกระจายของระยะเวลาที่ใช้เดินทางของยานพาหนะ



รูปที่ 4.1.3.4 ผลลัพธ์จากคำสั่ง dis_travel_time('CAV50Incident.csv',400,2700,7376000,7380000)

สามารถวิเคราะห์ข้อมูลระยะเวลาที่ยานพาหนะใช้ในการเดินทาง รูปที่ 4.1.3.2 โดยสังเกต ความสัมพันธ์ระยะเวลาที่ใช้ในการเดินทาง(แสดงในแกนตั้ง) กับช่วงเวลาที่เริ่มต้นเดินทาง (แสดงในแกนนอน)โดยที่สีของกราฟเส้นตรงแต่ละสีจะแสดงถึงระยะเวลาที่ยานพาหนะใช้ใน การเดินทางในแต่ละหมายเลขช่องจราจร

4.2 ผลลัพธ์ในการวิเคราะห์ในระดับจุลภาค (Microscopic)

4.2.1 ตรวจสอบรถคันหน้า และรถคันหลังของยานพาหนะคันที่สนใจ

สามารถใช้คำสั่งGetdataframe(dataframe).around(target = ‘k’, vid = v, t = t) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ k คือเป้าหมายประกอบไปด้วย ‘font’ และ ‘back’ ซึ่งหมายถึงคันหน้าและหลังตามลำดับ, v คือไอดีของยานพาหนะ และ t คือเวลาอ้างอิง หลักการทำงานคือโปรแกรมจะตรวจสอบจากค่า ‘Front X’ ที่มีค่ามากกว่าหรือน้อยกว่า คันที่สนใจน้อยที่สุด คือมีตำแหน่งไกลหรือกว่าคันที่สนใจน้อยที่สุด บน ‘Link ID’ และ ‘Lane ID’ เดียวกัน

Vehicle ID	Timestep	Link ID	Lane ID	Front X	Front Y	Rear X	Rear Y	Length	Width	Speed	Acceleration	Front Z	Rear Z	Unnamed: 14
169.0	309.8	1.0	2.0	7379500.0	-3398470.0	7379490.0	-3398470.0	4.76	2.06911	24.7671	-0.246022	0.0	0.0	NaN
	309.4	1.0	2.0	7379490.0	-3398470.0	7379480.0	-3398470.0	4.76	2.06911	24.8255	-0.055221	0.0	0.0	NaN
	309.0	1.0	2.0	7379480.0	-3398470.0	7379470.0	-3398470.0	4.76	2.06911	24.8496	-0.075443	0.0	0.0	NaN
	308.6	1.0	2.0	7379470.0	-3398470.0	7379460.0	-3398470.0	4.76	2.06911	24.8819	-0.097734	0.0	0.0	NaN
	308.2	1.0	2.0	7379460.0	-3398470.0	7379450.0	-3398470.0	4.76	2.06911	24.9234	-0.114752	0.0	0.0	NaN
	307.8	1.0	2.0	7379450.0	-3398470.0	7379440.0	-3398470.0	4.76	2.06911	24.9688	-0.101504	0.0	0.0	NaN
	307.4	1.0	2.0	7379440.0	-3398470.0	7379430.0	-3398470.0	4.76	2.06911	25.0074	-0.081690	0.0	0.0	NaN
	307.0	1.0	2.0	7379430.0	-3398470.0	7379420.0	-3398470.0	4.76	2.06911	25.0383	-0.064543	0.0	0.0	NaN
	306.6	1.0	2.0	7379420.0	-3398470.0	7379410.0	-3398470.0	4.76	2.06911	25.0537	0.187676	0.0	0.0	NaN
	306.2	1.0	2.0	7379410.0	-3398470.0	7379400.0	-3398470.0	4.76	2.06911	24.9386	0.664229	0.0	0.0	NaN
	305.8	1.0	2.0	7379400.0	-3398470.0	7379390.0	-3398470.0	4.76	2.06911	24.5835	0.855151	0.0	0.0	NaN
	305.4	1.0	2.0	7379390.0	-3398470.0	7379380.0	-3398470.0	4.76	2.06911	24.3285	-0.310793	0.0	0.0	NaN
	305.0	1.0	2.0	7379380.0	-3398470.0	7379370.0	-3398470.0	4.76	2.06911	24.5276	-0.684655	0.0	0.0	NaN

รูปที่ 4.2.1.2 ผลลัพธ์จากคำสั่ง Getdataframe(df3).around(target = 'front',vid = 167,t = '305:310')

สามารถตรวจสอบรถคันหน้า และรถคันหลังของยานพาหนะคันที่สนใจ ในเวลาที่สนใจได้

4.2.2 ตรวจสอบระยะห่างจากคันหน้า และรถคันหลังของยานพาหนะคันที่สนใจ

สามารถใช้คำสั่ง Find(dataframe).distance(target = ‘k’, vid = v, t = t) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ k คือเป้าหมายประกอบไปด้วย ‘font’ และ ‘back’ ซึ่งหมายถึงคันหน้าและหลังตามลำดับ, v คือไอดีของยานพาหนะที่ต้องการนำมาสร้างกราฟ และ t คือเวลาอ้างอิง หลักการทำงานคืออันดับแรกต้องตรวจว่ารถคันเป้าหมายคือคันใด โดยใช้คำสั่งจาก 4.1.1 จากนั้นนำค่า ‘Front X’ มาคำนวณกับรถคันที่สนใจ

```
Out[10]: [65.24000000022352,
          55.24000000022352,
          55.24000000022352,
          55.24000000022352,
          55.24000000022352,
          55.24000000022352,
          55.24000000022352,
          55.24000000022352,
          55.24000000022352,
          45.24000000022352,
          45.24000000022352,
          45.24000000022352,
          45.24000000022352]
```

รูปที่ 4.2.2.2 ผลลัพธ์จากคำสั่ง Find(df3).distance(target = 'front',vid = 167,t = '305:310')

สามารถตรวจสอบระยะห่างจากรถคันหน้า และรถคันหลังถึงรถคันที่สนใจ ในเวลาที่สนใจได้ ซึ่งจะให้ค่าเป็นระยะทาง สามารถนำไปใช้เพื่อคำนวณค่าต่าง ๆ ต่อได้

4.2.3 Headway ในแต่ละหน้าตัด

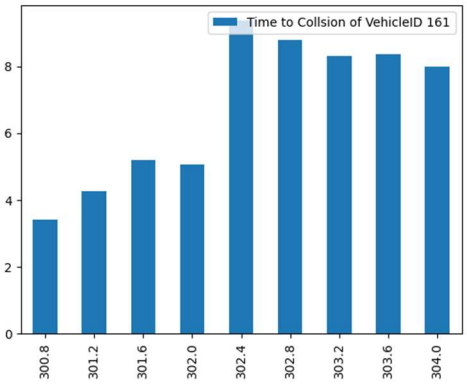
สามารถใช้คำสั่ง Plot(dataframe).headway(section,time) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ dataframe คือ DataFrame ของข้อมูลที่จะทำการคำนวณ, section คือตำแหน่งที่ต้องการคิด headway และ time คือช่วงเวลาที่ต้องการคำนวณ หลักการทำงานคือหลังจากกำหนด section แล้ว โปรแกรมจะทำการลูปข้อมูล เมื่อยานพาหนะคันไหนมี ‘Front X’ ผ่าน section แล้ว ก็จะเก็บค่า ‘Timestep’ ณ ตอนที่ผ่านไว้ และเมื่อครบตามที่กำหนดแล้วจึงนำมาสร้างกราฟ

รูปที่ 4.2.3.2 ผลลัพธ์จากคำสั่ง Plot(df3).headway(7378383,1,'300:350')

สามารถดู Headway แต่ละหน้าตัด ในแต่ละเลนได้ ซึ่งระยะห่าง ของจุดก็แสดงออกถึง headway และยังเชื่อมโยงไปถึง space - time diagram ได้ เนื่องจากจุดเวลานี้เป็นส่วนหนึ่งของ space - time diagram

4.2.4 Time to Collision

สามารถใช้คำสั่ง Plot(dataframe).ttc(vehicleID,time) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ dataframe คือ DataFrame ของข้อมูลที่จะทำการคำนวณ, vehicleID คือไอดีของยานพาหนะที่ต้องการคำนวณ Time to Collision (TTC) และ time คือ ช่วงเวลาที่สนใจ หลักการทำงานคือโปรแกรมจะลูปข้อมูลมีการตรวจสอบการวิ่งของยานพาหนะคันที่สนใจ และคันหน้า ณ เวลาหนึ่ง ๆ ซึ่งหากการวิ่ง ณ เวลานั้น สามารถทำให้เกิดการชนได้ก็จะคิด TTC จากสมการเส้นตรง เมื่อครบตามที่กำหนดแล้วจึงนำมาสร้างกราฟ



รูปที่ 4.2.4.2 ผลลัพธ์จากคำสั่ง Plot(df3).ttc(161,'300:304')

4.2.5 ตรวจสอบพฤติกรรมการเปลี่ยนเลนของยานพาหนะ สามารถใช้คำสั่ง Getdataframe(dataframe).changinglane(time) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ dataframe คือ DataFrame ของข้อมูลที่จะทำการคำนวณ และ time คือเวลาที่สนใจ หลักการทำงานคือโปรแกรมจะลูปข้อมูลยานพาหนะแต่ละคัน เมื่อข้อมูล ‘Lane ID’ มีการเปลี่ยนแปลงจากลูปก่อนหน้า หรือก็คือมีการเปลี่ยนเลน ก็จะบันทึกข้อมูลในแถวนั้นไว้

Vehicle ID	Timestep	Link ID	Lane ID	Front X	Front Y	Rear X	Rear Y	Length	Width	Speed	Acceleration	Front Z	Rear Z
161.0	302.4	1.0	3.0	7379270.0	-3398470.0	7379270.0	-3398470.0	4.012	1.85222	28.8528	-0.168514	0.0	0.0
	315.2	1.0	2.0	7379600.0	-3398470.0	7379600.0	-3398470.0	4.012	1.85222	24.9613	-0.450426	0.0	0.0
163.0	329.0	1.0	1.0	7379990.0	-3398480.0	7379990.0	-3398480.0	4.012	1.85222	25.7860	0.861497	0.0	0.0
165.0	302.6	1.0	1.0	7379190.0	-3398470.0	7379180.0	-3398470.0	4.644	1.99871	26.3299	1.231020	0.0	0.0
	308.2	1.0	2.0	7379340.0	-3398470.0	7379340.0	-3398480.0	4.644	1.99871	30.2744	1.062490	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
382.0	393.8	1.0	2.0	7376820.0	-3398460.0	7376820.0	-3398460.0	4.610	1.94931	23.8511	1.887710	0.0	0.0
	399.8	1.0	3.0	7376920.0	-3398460.0	7376920.0	-3398460.0	4.610	1.94931	18.8404	2.323710	0.0	0.0
383.0	395.4	1.0	3.0	7376830.0	-3398460.0	7376830.0	-3398460.0	4.012	1.85222	24.2303	-0.487404	0.0	0.0
384.0	397.8	1.0	2.0	7376860.0	-3398460.0	7376850.0	-3398460.0	4.760	2.06911	20.6250	-1.208170	0.0	0.0
386.0	400.0	1.0	3.0	7376680.0	-3398460.0	7376670.0	-3398460.0	4.610	1.94931	17.7050	-6.917300	0.0	0.0

รูปที่ 4.2.5.2 ผลลัพธ์จากคำสั่ง Getdataframe(df3).changinglane('300:400')

สามารถทราบถึงพฤติกรรมการเปลี่ยนเลนของรถแต่ละคันได้ ซึ่งจะนำเสนอในรูปแบบของ DataFrame ของยานพาหนะ ณ เวลาที่เปลี่ยนเลน

4.2.6 Space - Time Diagram

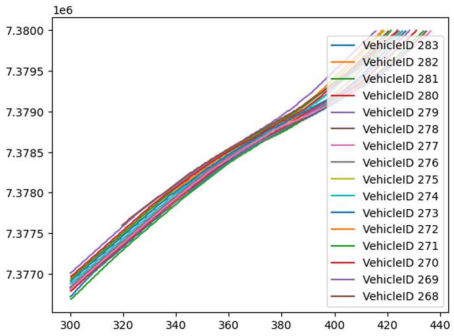
สามารถใช้คำสั่ง Plot(dataframe).spacetime(kind = ‘k’, vid = v, t = t) ซึ่งความหมายแต่ละพารามิเตอร์ได้แก่ v คือไอดีของยานพาหนะที่ต้องการนำมาสร้างกราฟ, t คือเวลาอ้างอิงที่จะใช้ในการสร้างกราฟ และ k คือรูปแบบของข้อมูลที่จะนำไปสร้างกราฟ ประกอบไปด้วย

4.2.6.1 kind = ‘all’ ใช้ข้อมูลยานพาหนะทุกคันในการสร้างกราฟ

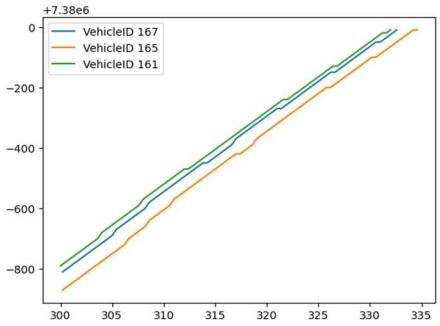
4.2.6.2 kind = ‘isolate’ สร้างกราฟจากยานพาหนะคันที่สนใจเท่านั้น

4.2.6.3 kind = ‘frontandback’ สร้างกราฟจากยานพาหนะคันที่สนใจ พร้อมทั้งยานพาหนะคันหน้าและหลังยานพาหนะคันที่สนใจในเวลาอ้างอิง

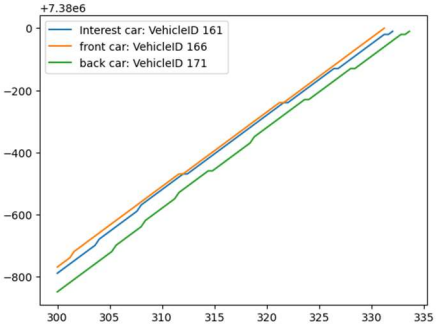
4.2.6.4 kind = ‘changinglane’ สร้างกราฟจากยานพาหนะคันที่สนใจ ซึ่งกำลังจะเปลี่ยนเลน โดยยานพาหนะคันหน้าจะเป็นยานพาหนะในเลนเริ่มต้น และยานพาหนะคันหลังจะเป็นยานพาหนะในเลนเป้าหมาย หลักการทำงานคือโปรแกรมจะลูปข้อมูล เก็บค่า ‘Front X’ และค่า ‘Timestep’ ไว้ตาม kind ที่กำหนด เมื่อครบแล้วจึงสร้างกราฟ



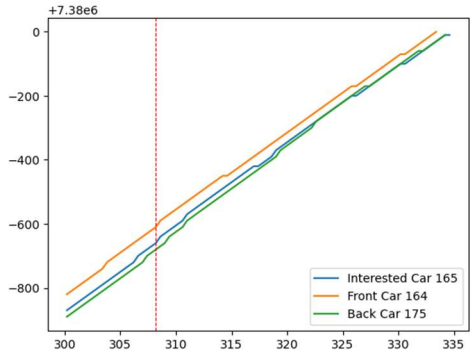
รูปที่ 4.2.6.2 ผลลัพธ์จากคำสั่ง Plot(df2[20000:25000]).spacetime(kind = 'all')



รูปที่ 4.2.6.3 ผลลัพธ์จากคำสั่ง Plot(df3).spacetime(kind = 'isolate',vid = [161,165,167])



รูปที่ 4.2.6.4 ผลลัพธ์จากคำสั่ง Plot(df3).spacetime(kind = 'frontandback',vid = 161, t = 302)



รูปที่ 4.2.6.5 ผลลัพธ์จากคำสั่ง Plot(df2).spacetime(kind = 'changinglane',vid = 165, t = 308.2)

Space - Time Diagram มีแกนนอนคือเวลา และแกนตั้งคือระยะทางโดยมีความชัน ของกราฟคือความเร็ว สามารถทำให้ทราบถึงพฤติกรรมการเคลื่อนที่ของยานพาหนะได้ หรือการดู พฤติกรรมของยานพาหนะคันต่าง ๆ ก็ทำให้ทราบถึงสาเหตุ หรือผลที่ตามมาของพฤติกรรมนั้น ๆ ได้

4.3 แนวทางในการนำชุดคำสั่งทางคอมพิวเตอร์ไปใช้งานกับข้อมูลการจราจรชุดอื่นๆ

4.3.1 แนวทางในการนำชุดคำสั่งระดับมหภาค

เริ่มต้นด้วยการใช้คำสั่ง heatmap_trafficspeed เพื่อศึกษาภาพรวมของกระแสการไหล สามารถใช้คำสั่ง Trafficspeed_per_time_alllane เพื่อศึกษาความเร็วของยานพาหนะในแต่ละช่วงเวลาในแต่ละช่องจราจร และคำสั่ง Trafficspeed_per_section_alllane เพื่อศึกษาความเร็วของยานพาหนะในแต่ละหน้าตัดของถนนในช่วงเวลาที่สนใจในแต่ละช่องจราจร ในการใช้คำสั่ง Trafficspeed_per_time_alllane และ Trafficspeed_per_section_alllane จะสามารถรู้ถึงความแตกต่างของยานพาหนะในแต่ละช่องจราจรได้อย่างชัดเจน

เมื่อทราบถึงปัญหาในการจราจรจากคำสั่งด้านนบนแล้ว สามารถใช้คำสั่ง density_plot_alllane เพื่อศึกษาถึงความหนาแน่นของยานพาหนะในแต่ละช่อง และอีกหนึ่งคำสั่งทางคอมพิวเตอร์ที่สามารถนำมาประยุกต์ในการศึกษากระแสจราจรในภาพรวมได้คือ Travel_time_from_atob ซึ่งสามารถศึกษาถึงระยะเวลาที่ใช้ในการเดินทางจากจุดที่สนใจสองจุดในแต่ละเวลาเริ่มเดินทางที่แตกต่างกัน

4.3.2 แนวทางในการนำชุดคำสั่งระดับจุลภาค

เมื่อได้ช่วงเวลาที่เกิดการจราจรที่เป็นปัญหา สามารถดึงข้อมูลที่อยู่ในช่วงเวลานั้นเพื่อหาว่ายานพาหนะคันใดเป็นคันแรกที่ความเร็วลดต่ำลง จากนั้นใช้คำสั่ง Plot(dataframe).spacetime(kind = 'frontandback', vid = v, t = t) สร้าง space-time diagram ของยานพาหนะคันนั้นกับคันรอบ ๆ ได้เพื่อดูว่าเกิดการเคลื่อนที่ที่ไม่พึงประสงค์ในช่วงเวลานั้นหรือไม่

จากนั้นใช้คำสั่ง Plot(dataframe).ttc(vehicleID,time) จะสามารถสร้างกราฟ Time to Collision ของคันที่ความเร็วลดต่ำลง หากมีค่า TTC ที่เข้าใกล้ 0 ก็มีความเป็นไปได้สูงที่จะเกิดการชน หรือสามารถใช้คำสั่ง Plot(dataframe).headway(section,time) สร้างกราฟ headway หากจุดอยู่ที่ตำแหน่งเดียวกันในเวลาเดียวกันก็มีความเป็นไปได้สูงที่จะเกิดการชน ต่อจากนั้นใช้คำสั่ง Getdataframe(dataframe).around(target = 'k', vid = v, t = t) เพื่อตรวจดูยานพาหนะคันหน้าของคันที่ความเร็วลดต่ำลง เพื่อดูสถานะ ณ เวลานั้น ๆ เช่นช่องการจราจร พิกัด ความเร็ว และความเร่ง หากมีพิกัดอยู่จุดเดียวกันก็อาจจะสรุปได้ว่าเกิดการชน

ยิ่งไปกว่านั้นยังสามารถดึงข้อมูลรถที่ทำการเปลี่ยนเลนจากคำสั่ง Getdataframe(dataframe).changinglane(time) หากรถคันหน้านั้นอยู่ในรถที่ทำการเปลี่ยนเลน ณ เวลานั้นก็อาจสรุปได้ว่าสาเหตุของการจราจรที่ผิดปกติมาจากการเปลี่ยนเลนที่ไม่พึงประสงค์ สามารถสร้าง space-time diagram เพื่อตรวจสอบสภาพแวดล้อมในช่วงเปลี่ยนเลนได้เช่นกันผ่านคำสั่ง Plot(dataframe).spacetime(kind = 'changinglane', vid = v, t = t)

5. สรุปผลการวิจัย

โครงงานนี้มีวัตถุประสงค์เพื่อเขียนชุดคำสั่งในภาษา Python เพื่อวิเคราะห์ข้อมูลการ เคลื่อนที่ของรถ ซึ่งเป็นข้อมูลขนาดใหญ่และแสดงผลในรูปแบบกราฟฟิกเพื่อให้เกิดความเข้าใจ ต่อผลการวิเคราะห์

จากการวิจัย การใช้คำสั่งทางคอมพิวเตอร์และแสดงผลออกมาในรูปแบบกราฟฟิก ทำให้เราสามารถวิเคราะห์ข้อมูลได้ง่ายขึ้น สามารถที่จะใช้ชุดคำสั่งร่วมกันเพื่อให้ได้ถึงข้อมูลและแสดงออกถึงสภาพจราจรทั้งในระดับมหภาค (Macroscopic) และระดับจุลภาค (Microscopic) ชุดคำสั่งสำหรับวิเคราะห์ผลในระดับมหภาคช่วยให้สามารถเข้าใจข้อมูลในภาพรวม และในการตรวจสอบสภาพ แวดล้อมโดยละเอียดของยานพาหนะแต่ละคันสามารถที่จะใช้ชุดคำสั่งสำหรับวิเคราะห์ผลในระดับจุลภาค การใช้ชุดคำสั่งทั้งสองระดับร่วมกันสามารถทำให้กระแสจราจรที่มีปัญหา และสาเหตุของปัญหาได้อย่างชัดเจน

ในงานวิจัยนี้ได้ทำการนำเสนอแนวทางการใช้งานชุดคำสั่งเบื้องต้นเพื่อเป็นแนวทางให้ผู้ใช้งานที่มีความสนใจ สามารถนำไปใช้งานได้อย่างมีประสิทธิภาพและแสดงข้อมูลที่ต้องการได้อย่างครบถ้วนสมบูรณ์

เอกสารอ้างอิง

[1] 1stcraft.com. (2563). Big Data คืออะไร อธิบายแบบเข้าใจง่าย https://1stcraft.com/what-is-big-data/#0-ทำความรู้จักกับ-big-data

[2] Ragnar Corporation. (2563). JUPYTER NOTEBOOK คือ เครื่องมือสำหรับ DATA SCIENCE. https://blog.ragnar.co.th/blog/what-is-jupyter-notebook

[3] Bernard Jacob, Eric Violette. (2012, July 14). Vehicle Trajectory Analysis: An Advanced Tool for Road Safety. https://www.sciencedirect.com/science/article/pii/S1877042812028984

[4] Cheol Oh, Taejin Kim. (2010, June 7). Estimation of rear-end crash potential using vehicle trajectory data. https://www.sciencedirect.com/science/article/pii/S0001457510001454