รายงานสหกิจศึกษาฉบับสมบูรณ์

การออกแบบโปรแกรมประมวลผลข้อมูลสรุปงาน
ติดตั้งสถานีฐานโดยใช้ภาษาอาร์
Design of data analysis program about the base station using R language.

สลิตา อยู่เย็น

สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

การออกแบบโปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐานโดยใช้ภาษาอาร์

Design of data analysis program about the base station using R language.

สลิตา อยู่เย็น

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

Design of data analysis program about the base station using R language.

Sarita Yooyen

BACHELOR OF ENGINEERING IN TELECOMMUNICATION ENGINEERRING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2017

กิตติกรรมประกาศ

โครงงานสหกิจศึกษานี้สำเร็จลุล่วงไปได้ด้วยดีนั้น ผู้จัดทำโครงงานขอขอบพระคุณบริษัทหัวเว่ย เทคโนโลยี่ (ไทยแลนด์) จำกัด พี่พนักงานที่ปรึกษาคุณพิชิต เอกทะนง ที่ให้โอกาสเละให้การสนับสนุนการทำงานและโครงงานมาโดยตลอด ขอบคุณพี่ๆพนักงานทุกคน ที่คอยให้ความช่วยเหลือและช่วยสอนงานต่าง ๆ รวมถึงให้ข้อเสนอแนะที่เป็นประโยชน์ในการทำงาน

ขอขอบพระคุณอาจารย์นภัทร สระเอี่ยม อาจารย์ที่ปรึกษาโครงงานที่ได้กรุณาให้แนวคิด รวมถึงให้ข้อเสนอแนะ ช่วยแก้ไขข้อบกพร่องและติดตามโครงงานสหกิจศึกษามาโดยตลอดจนโครงงานสหกิจศึกษาเล่มนี้เสร็จสมบูรณ์

ขอขอบพระคุณเจ้าหน้าที่และบุคลากรสหกิจศึกษาทุกคน ที่ให้การสนับสนุนและคอยช่วยเหลือการทำงานของนักศึกษาสหกิจศึกษาเป็นอย่างดี

ขอบขอบคุณพ่อ แม่ พี่น้อง และเพื่อนทุกคนที่คอยช่วยเหลือ เป็นกำลังใจและให้การสนับสนุนในเรื่องต่างๆ ต่อผู้จัดทำโครงงานเสมอมา ขอบคุณที่รับฟังและเข้าใจเวลาเจออุปสรรคในการทำงานต่างๆ ผู้จัดทำโครงงานของกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

สลิตา อยู่เย็น

ผู้จัดทำโครงงาน

โครงงานสหกิจศึกษา การออกแบบโปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐานโดยใช้ภาษาอาร์

ชื่อ – สกุล นักศึกษา นางสาวสลิตา อยู่เย็น รหัสนักศึกษา 57011323

คณะ วิศวกรรมศาสตร์ **สาขาวิชา** วิศวกรรมโทรคมนาคม

ชื่อ – สกุล อาจารย์นิเทศ ผศ.ดร.นภัทร สระเอี่ยม

ชื่อ – สกุล ผู้นิเทศงาน นายพิชิต เอกทนง

ชื่อสถานประกอบการ บริษัท หัวเว่ย เทคโนโลยี่ (ไทยแลนด์) จำกัด

บทคัดย่อ

รายงานสหกิจศึกษาฉบับสมบูรณ์นี้เป็นส่วนหนึ่งของวิชาสหกิจศึกษา ที่กำหนดให้นักศึกษาทำโครงงานร่วมกับทางสถานประกอบการเป็นระยะเวลาหนึ่งภาคการศึกษา โดยบริษัทหัวเว่ยเทคโนโลยี่ (ไทยแลนด์) จำกัด เป็นบริษัทที่ให้บริการทั้งในด้านของอุปกรณ์เครือข่ายเน็ตเวิร์คโทรคมนาคม และยังมีบริการงานในด้านการติดตั้งเสาสัญญาณ ข้อมูลเกี่ยวการติดตั้งนี้เองต้องมีการสรุป และติดตามความคืบหน้าเกี่ยวกับโครงการตลอดเวลา โดยโครงงานฉบับนี้เป็นการศึกษาภาษาอาร์เพื่อนำมาใช้ในการออกแบบโปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐาน เนื่องจากการมองเห็นปัญหาของบริษัทในด้านการจัดทำรายงานสรุปผลข้อมูล ซึ่งข้อมูลที่ถูกนำมาวิเคราะห์และจัดทำเป็นกราฟแสดงผลนั้นจะใช้วิธีการจัดเรียงข้อมูลโดยพนักงานภายในบริษัท แต่เนื่องจากข้อมูลอยู่ในรูปแบบของ Big Data ขนาดใหญ่ที่ประกอบไปด้วยข้อมูลจำนวนมาก ทำให้การสรุปผลข้อมูลในแต่ละครั้งใช้เวลาที่นานและเป็นการเพิ่มงานให้กับพนักงานในบริษัท ด้วยเหตุนี้จึงทำการออกแบบโปรแกรมภาษาอาร์ให้สามารถนำมาช่วยในการจัดเรียงและแสดงผลข้อมูล โดยทำการออกแบบโปรแกรมเป็นแอพพลิเคชั่นเว็บที่สามารถโต้ตอบกับผู้ใช้งานได้ ซึ่งข้อมูลจะถูกแสดงผลตามที่ผู้ใช้งานต้องการ

Co-operative Title: Design of data analysis program about the base station using R language.

Student Intern Name: Miss Sarita Yooyen

Faculty: Engineering **Department:** Telecommunications Engineering

Advisor Name: Asst.Prof.Dr. Napat Sra-ium

Mentor Name: Mr. Pichit Ekthanong

Company: Huawei Technologies (Thailand) Co.,Ltd.

ABSTRACT

This cooperative education report is part of cooperative education. The students are required to work on a project for a period of one semester. Huawei Technologies (Thailand) company limited is a company that provides services in network equipment, telecommunications and also services in the installation of antenna. This installation information must be finalized and keep tracked the project progress at all time. This project is a R language study to be used in the design of the application program because of the company's problems in the preparation of a summary report. The information that is analyzed and graphed will be used by the internal staff. But since the data is in the form of large data, Big Data contains a lot of information. Make a summary of each time, take a long time and add work to the employees in the company. Because of this, the R language program can be used to organize and display information. The program is designed to be a web application that can interact with users. The data will be displayed according to user requirements.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		IVI
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์ของโครงงาน	2
	1.3 ขอบเขตของโครงงาน	2
	1.4 วิธีดำเนินงาน	3
	1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**5**
	2.1 เทคโนโลยีโทรศัพท์เคลื่อนที่	5
	2.2 การติดตั้งเสาสัญญาณโทรศัพท์เคลื่อนที่	11
	2.3 ภาษาอาร์	17
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**37**
	3.1 การออกแบบ	37
	3.2 เครื่องมือที่ใช้ในการทดลอง	40
	3.3 การจัดเก็บผลการทดลอง	41

สารบัญ (ต่อ)

		หน้า
บทที่ 4	**ผลการทดลอง**	**51**
	4.1 ผลการทดสอบการนำไฟล์เข้าโดยใช้ XLCONNECT	51
	4.2 ผลการทดสอบการใช้คำสั่ง SUBSET และ SUBSTR	52
	4.3 ผลการทดสอบการใช้คำสั่ง CBIND	53
	4.4 ผลการทดสอบการใช้คำสั่ง MERGE	53
	4.5 ผลการทดสอบการใช้คำสั่ง AGGREGATE	54
	4.6 ผลการทดสอบการพล็อตกราฟแท่ง	54
	4.7 ผลการทดสอบการพล็อตกราฟวงกลม	55
	4.8 ผลการทดสอบการสร้างแอพพลิเคชั่นเว็บจาก SHINY	56
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**61**
	5.1 สรุปผล	61
	5.2 ข้อเสนอแนะ	62
บรรณานุกรม		**63**
ภาคผนวก	ภาษาอาร์	**64**

สารบัญรูป

หน้า

รูปที่

2.1 วิวัฒนาการของโทรศัพท์มือถือ 6
2.2 โครงสร้างของระบบโทรศัพท์เซลลูลาร์ 9
2.3 การทาสีของเสาที่ระดับความสูงต่างๆ 11
2.4 เสาสัญญาณ ROOFTOP 12
2.5 เสา POLE 12
2.6 เสา GREEN FIELD 12
2.7 เสาสัญญาณ MONOPOLE 13
2.8 เสาสัญญาณ SELF-SUPPORT 13
2.9 เสาสัญญาณ GUYED 14
2.10 การทำงานระหว่างอุปกรณ์ภายในสถานีฐาน 14
2.11 BBU หรือ BASE BAND UNIT 15
2.12 RRU หรือ REMOTE RADIO UNIT 16
2.13 การเชื่อมต่ออุปกรณ์สถานีฐานไปยังจานสายอากาศ 16
2.14 ตู้โนดบีและอุปกรณ์ภายในตู้ 17
2.15 สัญลักษณ์ของภาษาอาร์ 17
2.16 หน้าต่างโปรแกรมภาษาอาร์ 18
2.17 หน้าต่างโปรแกรม RSTUDIO 18
2.18 การนำเข้าข้อมูลโดยวิธีการ IMPORT DATASET 24
2.19 การพล็อตประชันทุกเขตข้อมูลของข้อมูลดอกกุหลาบ 26
2.20 การพล็อตกระจายข้อมูลดอกกุหลาบ 26
2.21 โมเดลเชิงเส้นของความกว้างต่อความยาวกลีบดอก 27

สารบัญรูป (ต่อ)

2.22 โมเดลไม่เชิงเส้นของความกว้างต่อความยาวกลีบดอก 28

2.23 โมเดลจากสมการ $E^{X-(4/3)\Pi}$ 29

2.24 ข้อมูลดอกกุหลาบโดยแยกสีตามสปีชีส์ 30

2.25 กราฟแท่งแสดงการแจกแจงเกรดของนักเรียนโดยใส่สี 30

2.26 ฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบ 31

2.27 ฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบโดยขีดเส้นแบ่งช่วง 4 ครั้ง 31

2.28 ฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบโดยแบ่งขนาดช่วงไม่เท่ากัน 32

2.29 พล็อตกล่องความกว้างกลีบเลี้ยงดอกกุหลาบ 33

2.30 การพล็อตกราฟวงกลมแจกแจงระดับการศึกษา 33

2.31 กราฟพยากรณ์อุณหภูมิของเมืองนอตทิงแฮม 36

2.32 กราฟแผนที่โลกพร้อมตำแหน่งเกิดแผ่นดินไหว 36

3.1 แอพพลิเคชั่น SHINY ที่ทำการออกแบบ 39

3.2 ไอคอนภาษาอาร์ 40

3.3 หน้าต่างโปรแกรม RSTUDIO 40

3.4 ข้อมูล BIG DATA ที่ทำการดึงเข้าโปรแกรมอาร์ 41

3.5 ข้อมูลประกอบการใช้คำสั่ง SUBSET และ SUBSTR 42

3.6 ข้อมูลประกอบการใช้คำสั่ง CBIND 43

3.7 ข้อมูลประกอบการใช้คำสั่ง MERGE 43

3.8 ข้อมูลไฟล์ “DATASETNEW” ประกอบการใช้คำสั่ง AGGREGATE 44

3.9 ข้อมูลแสดงการแจกแจงจำนวนงานที่เวลาต่าง ๆ 45

3.10 ข้อมูลแสดงการแจกแจงจำนวนงานที่เวลาต่างๆ 45

3.11 รูปแบบของแอพพลิเคชั่นเว็บจากแพคเกจ SHINY 47

3.12 ชุดรูปแบบการรับข้อมูลจากผู้ใช้งาน 48

สารบัญรูป (ต่อ)

3.13	การจัดเก็บไฟล์รูปที่ใช้แสดงบนแอพพลิเคชั่นเว็บ	49
4.1	การทดสอบการนำไฟล์เข้าโดยใช้ XLCONNECT	51
4.2	การทดสอบคำสั่ง SUBSTR	52
4.3	การทดสอบคำสั่ง SUBSET	52
4.4	การทดสอบคำสั่ง CBIND	53
4.5	การทดสอบคำสั่ง MERGE	53
4.6	การใช้คำสั่ง AGGREGATE คำนวณค่าต่ำสุดและค่าสูงสุด	54
4.7	การทดสอบการพล็อตกราฟแท่ง	55
4.8	การทดสอบการพล็อตกราฟแท่งแบบ 2 กราฟ	55
4.9	การทดสอบการพล็อตกราฟวงกลม	56
4.10	การทดสอบการพล็อตกราฟวงกลมแบบ 4 กราฟ	56
4.11	การทดสอบการสร้างกล่องที่มีตัวเลือกให้เลือก	57
4.12	การทดสอบการสร้างชุดปุ่มวิทยุ	57
4.13	การทดสอบการสร้างการอัพโหลดไฟล์นามสกุล .CSV	58
4.14	ทดสอบการแสดงข้อมูลเวลา	58
4.15	ทดสอบการแสดงข้อมูลที่ผู้ใช้ทำเลือก	59
4.16	การทดลองแสดงกราฟแท่งบนหน้าต่างแอพพลิเคชั่นเว็บ	59
4.17	การทดลองแสดงกราฟวงกลมบนหน้าต่างแอพพลิเคชั่นเว็บ	60

บทที่ 1
บทนำ

1.1. ความเป็นมาและความสำคัญ

บริษัทหัวเหว่ยเทคโนโลยี (Huawei Technologies) เป็นบริษัทผู้ผลิตอุปกรณ์เครือข่ายและอุปกรณ์โทรคมนาคมสัญชาติจีน มีสำนักงานใหญ่อยู่ที่เมืองเซินเจิ้น มณฑลกวางตุ้ง สาธารณรัฐประชาชนจีน ปัจจุบันเป็นผู้ผลิตอุปกรณ์เครือข่ายและการโทรคมนาคมรายใหญ่ที่สุดของประเทศจีน ทั้งยังเป็นผู้ผลิตอุปกรณ์และติดตั้งงานโครงสร้างพื้นฐานของระบบโทรศัพท์เคลื่อนที่รายใหญ่อันดับหนึ่งของโลก บริษัทหัวเหว่ยมีธุรกิจหลักแบ่งออกเป็นสามกลุ่มใหญ่ ๆ ได้แก่กลุ่มเครือข่ายโทรคมนาคม กลุ่มผลิตภัณฑ์และบริการสำหรับภาคธุรกิจและองค์กร และกลุ่มอุปกรณ์สื่อสาร นอกจากนี้ หัวเหว่ยยังมีศูนย์วิจัยและพัฒนาเทคโนโลยีอีกกว่า 20 แห่งทั่วโลก

ในปัจจุบันหัวเหว่ยได้ขยายพื้นที่ให้บริการออกไปยังประเทศต่าง ๆ หนึ่งในนั้นรวมถึงประเทศไทย โดยหัวเหว่ยได้เข้ามามีบทบาททั้งในด้านของเครือข่ายโทรคมนาคมและบริการที่เกี่ยวข้อง ระบบฮาร์ดแวร์และซอฟต์แวร์เครือข่ายสำหรับลูกค้าองค์กร ตลอดจนถึงสมาร์ทโฟนและแท็บเล็ตคอมพิวเตอร์ หัวเว่ยเทคโนโลยีไทยแลนด์ประกอบด้วยแผนกต่าง ๆ มากมาย โดยงานหลักของบริษัทจะเกี่ยวข้องกับการติดตั้งเครือข่ายโทรคมนาคมให้กับผู้ให้บริการความถี่คลื่นสัญญาณโทรศัพท์ภายในประเทศ ไม่ว่าจะเป็น DTAC AIS และ TRUE ทำให้เกิดข้อมูลที่ทางบริษัทจำเป็นต้องเก็บค้นคว้า และรวบรวมเพื่อทำเป็นรายงานสรุปถึงข้อมูลที่สำคัญต่าง ๆตลอดเวลาเช่น ความคืบหน้าของแต่ละโครงการเป็นต้น ข้อมูลต่าง ๆ เหล่านี้จะถูกนำเสนอต่อผู้บริหารเพื่อเป็นการประกอบการในการคิดวิเคราะห์วางแผน และแก้ปัญหา เกี่ยวกับความคืบหน้าของโครงการที่ทำอยู่ ซึ่งพบว่าข้อมูลต่าง ๆมีจำนวนมากเนื่องจากมีการติดตั้งเสาสัญญาณทุกเครือข่ายตลอดทั่วทั้งประเทศไทย โดยเมื่อมีข้อมูลเพิ่มเข้ามา จะต้องทำการวิเคราะห์และจัดเรียงข้อมูลใหม่เสมอ ซึ่งก่อให้เกิดปัญหาเนื่องจากรายงานสรุปผลการทำงานของแต่ละโครงการ จำเป็นต้องใช้เวลาที่นานในการจัดเรียงข้อมูลและแสดงผล นอกจากนี้ข้อมูลอาจเกิดข้อผิดพลาดเนื่องจากผู้จัดทำรายงาน จากปัญหานี้เองจึงเกิดเป็นโครงงานออกแบบโปรแกรมช่วยวิเคราะห์ข้อมูลโดยใช้ภาษาอาร์ที่เป็นภาษาโปรแกรมมิ่งเหมาะสำหรับการนำมาวิเคราะห์ข้อมูลทางสถิติและการแสดงผลข้อมูลต่าง ๆ ทั้งยังช่วยสรุปข้อมูลให้เข้าใจถึงปัญหามากขึ้น ตลอดจนเพื่อความสะดวกรวดเร็วในการจัดเรียงข้อมูลและการวิเคราะห์ข้อมูล นอกจากนี้ยังช่วยประหยัดเวลาให้กับองค์กรอีกด้วย

1.2. วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาและออกแบบโปรแกรมช่วยวิเคราะห์ข้อมูลโดยใช้ภาษาอาร์เพื่อประกอบการตัดสินใจของผู้บริหารในการวางแผนการทำงาน ดูความคืบหน้าของโครงการ ดูประสิทธิภาพของพนักงานภายในองค์กร ตลอดจนแก้ปัญหาการทำงานที่เกิดขึ้น

2) ช่วยลดค่าใช้จ่ายของบริษัทในการจ้างพนักงานวิเคราะห์ข้อมูล รวมถึงช่วยประหยัดเวลาในการจัดเรียงและแสดงผลข้อมูล ทำให้สามารถแก้ปัญหาและหาข้อสรุปเกี่ยวกับในการทำงานได้อย่างรวดเร็ว

3) เพื่อลดปัญหาเกี่ยวกับภาระงานให้กับพนักงานในองค์กร

1.3. ขอบเขตของโครงงาน

1) การออกแบบโปรแกรมวิเคราะห์ข้อมูลโดยใช้ภาษาอาร์เพื่อประกอบการตัดสินใจของผู้บริหาร

2) เขียนโปรแกรมประมวลผลข้อมูลและจัดเรียงข้อมูลให้อยู่ในรูปแบบที่ต้องการ ก่อนนำไปแสดงผลข้อมูล

3) สร้างแอพพลิเคชั่นในการแสดงผลและสรุปข้อมูลที่สำคัญตามความต้องการของบริษัท

1.4. วิธีดำเนินงาน

1) ศึกษาข้อมูลและเรียนรู้กระบวนการทำงานภายในบริษัทหัวเว่ยเทคโนโลยี่ (ไทยแลนด์) จำกัด แผนก PMO (Project Management Office)

2) ลงพื้นที่ทำงานเพื่อเข้าใจกระบวนการทำงานจริงภายในไซต์งานการติดตั้งเสาสัญญาณ เพื่อให้เกิดความเข้าใจต่อรูปแบบงานมากขึ้น

3) ศึกษากระบวนการมาตรฐานการติดตั้งเสาสัญญาณโทรศัพท์ที่ความถี่ต่าง ๆ และมาตรฐานการตรวจรับงานของลูกค้าตามแต่ละประเภทของงาน

4) ศึกษาความปลอดภัยภายในไซต์งานการติดตั้งเสาสัญญาณ การตรวจสอบเครื่องมือและอุปกรณ์ความปลอดภัยต่าง ๆภายในบริเวณสถานที่การทำงานให้อยู่ในมาตรฐานของบริษัท

5) ศึกษาข้อมูลอุปกรณ์ในการติดตั้งเสาสัญญาณ และหน้าที่ของแต่ละตัวอุปกรณ์ในงานติดตั้ง

6) ศึกษารูปแบบการทำรายงาน ความสำคัญของรายงาน ตลอดจนวิเคราะห์ความต้องการของผู้บริหารเกี่ยวกับการนำเสนอข้อมูลในที่ประชุม

7) รวบรวมข้อมูลและวิเคราะห์ปัญหาการทำงานภายในบริษัท

8) ศึกษาการเขียนโปรแกรมภาษาอาร์

9) ออกแบบรูปแบบโปรแกรม และข้อมูลที่ต้องการนำเสนอ

10) เขียนโปรแกรมโดยใช้ภาษาอาร์ในวิเคราะห์ข้อมูล สรุปผลข้อมูล และแสดงข้อมูลเพื่อนำเสนอต่อผู้บริหาร

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถใช้ความรู้ภาษาอาร์ในการวิเคราะห์ข้อมูล และเขียนโปรแกรมเพื่อช่วยในการประมวลผลและแสดงผลของข้อมูลได้

2) ได้เรียนรู้ระบบการทำงานในกระบวนติดตั้งเสาสัญญาณโทรศัพท์เคลื่อนที่ของผู้ให้บริการภายในประเทศ ตลอดถึงมาตราฐานความปลอดภัยภายในไซต์งานการติดตั้งเสาสัญญาณ

3) ได้เรียนรู้รูปแบบในการทำรายงาสรุปความสำคัญของงาน เพื่อวิเคราะห์ถึงปัญหาที่เกิดขึ้น และการนำเสนองานต่อผู้บริหาร

4) สามารถลดค่าใช้จ่ายและเวลาของบริษัทในวิเคราะห์ข้อมูล

5) ได้เรียนรู้กระบวนการทำงานต่าง ๆของการทำงานภายในบริษัทหัวเว่ย เทคโนโลยี่ (ไทยแลนด์) จำกัด แผนก PMO ((Project Management Office) และได้ทราบถึงบทบาท หน้าที่ และความสำคัญของงานในแผนกงานอื่นเช่นแผนก RF (Radio frequency) เป็นต้น

6) ได้ทักษะในการติดต่อประสานงาน การใช้ภาษาภายในองค์กร และการทำงานร่วมกับผู้อื่น

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

บริษัทหัวเว่ยเทคโนโลยี่ไทยแลนด์จำกัด เป็นบริษัทที่ให้บริการทั้งในด้านของอุปกรณ์เครือข่ายเน็ตเวิร์คโทรคมนาคม ระบบฮาร์ดแวร์และซอฟต์แวร์เครือข่ายสำหรับลูกค้าองค์กร ตลอดจนถึงสมาร์ทโฟนและแท็บเล็ตคอมพิวเตอร์ นอกจากนี้ยังมีบริการงานในด้านการติดตั้งเสาสัญญาณ การติดตั้ง RRU (Remote Radio Unit) และการเปลี่ยนแบตเตอรี่ให้กับตู้โนดบี(ตู้ใส่อุปกรณ์การติดตั้งเสาสัญญาณ) โดยในปัจจุบันการติดตั้งเสาสัญญาณโทรศัพท์เคลื่อนที่ในประเทศไทยมีการเติบโตมากขึ้น ส่งผลให้มีการบริการติดตั้งเสาสัญญาณให้กับเครือข่าย TRUE DTAC และ AIS ทั่วประเทศ ข้อมูลเกี่ยวการติดตั้งนี้เองต้องมีการสรุป และติดตามความคืบหน้าเกี่ยวกับโครงการตลอดเวลาเพื่อนำเสนอให้กับผู้บริหารในการแก้ปัญหาการทำงานที่เกิดขึ้นเช่น การทำงานที่ล่าช้าหรือการส่งงานที่ล่าช้าของผู้รับเหมาตลอดจนช่วยดูประสิทธิภาพการตรวจรับงานของพนักงานภายในองค์กรอีกด้วย ซึ่งขั้นตอนการสรุปข้อมูลนี้จำเป็นต้องมีการจัดเรียงข้อมูล ตลอดจนแสดงผลของข้อมูลออกมา โดยปกติจะใช้วิธีการสรุปข้อมูลโดยใช้พนักงานในองค์กรเป็นคนจัดเรียงข้อมูลก่อนแสดงผล ซึ่งทำให้เสียเวลามากเนื่องจากมีหลายโครงการที่กำลังดำเนินการ อีกทั้งยังต้องสรุปงานแต่ละโครงการตลอดเวลาเป็นรายสัปดาห์ รายเดือน หรือตั้งแต่เริ่มทำโครงการ ซึ่งประกอบไปด้วยข้อมูลจำนวนมาก การสรุปและวิเคราะห์ข้อมูลในส่วนนี้จึงทำให้บริษัทเกิดความล่าช้าในการแก้ไขปัญหาเกี่ยวกับงานที่เกิดขึ้น และเป็นการสร้างภาระงานให้กับพนักงานในองค์กร ดังนั้นจึงเกิดเป็นโครงงานการสรุปข้อมูลเกี่ยวกับกับติดตั้งเสาสัญญาณขึ้น โดยมีการนำภาษาอาร์ ภาษาโปรแกรมมิ่งที่พัฒนามาจากภาษาเอสมีจุดเด่นในเรื่องการวิเคราะห์ทางสถิติและแสดงผลออกมาเป็นกราฟได้ มาช่วยในการวิเคราะห์และแสดงผลของข้อมูล

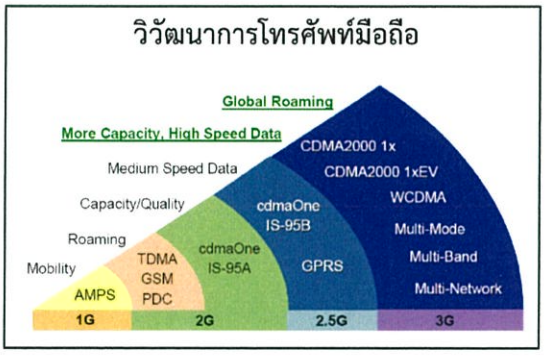
2.1 เทคโนโลยีโทรศัพท์เคลื่อนที่

เครือข่ายโทรศัพท์มือถือเป็นเทคโนโลยีเครือข่ายไร้สายประเภทหนึ่งที่ได้รับความนิยมอย่างมากจากผู้ใช้ทั่วโลก เนื่องจากมีความสะดวกสบายในการใช้งาน ยุคแรกๆ นั้นจะเป็นการใช้สำหรับสื่อสารด้วยเสียงสนทนาในรูปแบบสัญญาณแอนะลอก (Analog Signal) เพียงอย่างเดียว จนกระทั่งพัฒนามาเป็นการรับส่งข้อมูลต่าง ๆ ในรูปแบบดิจิทัล (Digital Signal) เพื่อเพิ่มความสามารถในการ

รองรับการส่งข้อมูล อื่นๆ เช่น ข้อความหรือ text ไฟล์เพลง รูปภาพหรือคลิปวิดีโอ และปัจจุบันสามารถเชื่อมต่อเข้ากับอินเทอร์เน็ตทำให้สามารถรับส่งอีเมล เข้าดูเว็บไซต์ต่างๆ ได้ จุดด้อยของเครือข่ายโทรศัพท์ คือ แบนด์วิธต่ำแต่ในปัจจุบันได้มีการพัฒนาเพื่อขยายแบนด์วิธเพิ่มมากขึ้นตาม ความต้องการของผู้ใช้ตามยุคต่างๆ ตอบสนองผู้ใช้บริการได้เป็นจำนวนมากขึ้น

2.1.1 วิวัฒนาการโทรศัพท์เคลื่อนที่

วิวัฒนาการของระบบโทรศัพท์มือถือนั้นมักนิยมเรียกชื่อหรือแบ่งเป็นยุคๆ โดยในยุคแรกเรียกว่า ยุค 1G และมีการลำดับตัวเลขเพิ่มขึ้นเรื่อยๆ โดยปัจจุบันอยู่ในช่วงยุค 3G แต่ละยุคของโทรศัพท์มือถือมีระยะเวลาโดยเฉลี่ยราว 10 ปี วิวัฒนาการของโทรศัพท์ในแต่ละยุคแสดงดังรูปที่ 2.1



รูปที่ 2.1 วิวัฒนาการของโทรศัพท์มือถือ

1. การสื่อสารโทรศัพท์เคลื่อนที่ในยุค 1 (First Generation: 1G)

การสื่อสารโทรศัพท์เคลื่อนที่ในยุค 1 เป็นยุคแรกของการพัฒนาระบบโทรศัพท์แบบเซลลูลาร์ การรับส่งสัญญาณใช้วิธีการมอดูเลตสัญญาณแบบแอนะลอก (Analog) เข้าช่องสื่อสารโดยใช้การแบ่งความถี่ ด้วยวิธีการนี้มีข้อจำกัดในเรื่องจำนวนช่องสัญญาณ และการใช้ไม่เต็มประสิทธิภาพจึงติดขัดเรื่องการขยายจำนวนเลขหมาย และการขยายแถบความถี่

2. การสื่อสารโทรศัพท์เคลื่อนที่ยุค 2 (Second Generation: 2G)

การสื่อสารโทรศัพท์เคลื่อนที่ยุค 2 เป็นยุคที่พัฒนาต่อมาโดยการเข้ารหัสสัญญาณเสียง ด้วยวิธีการบีบอัดสัญญาณเสียงในรูปแบบดิจิทัล ซึ่งในยุคนี้เองเป็นยุคที่เริ่มทำให้เราเริ่มที่จะสามารถใช้งานทางด้านข้อมูลได้ นอกเหนือจากการใช้งานเสียงเพียงอย่างเดียว ยุคนี้เป็นยุคที่เกิดเทคโนโลยี GPRS (General Packet Radio Service) ซึ่งตามหลักการแล้วเทคโนโลยี GPRS นี้สามารถส่งข้อมูลได้ที่ความเร็วสูงสุดถึง 115 กิโลบิตต่อวินาที แต่ในการใช้งานจริงความเร็วของ GPRS จะถูกจำกัดให้อยู่ที่ประมาณ 40 กิโลบิตต่อวินาทีเท่านั้น และเป็นช่วงที่เริ่มมีการใช้เทคโนโลยี EDGE (Enhanced Data rates for Global Evolution) EDGE ซึ่งถือเป็นเทคโนโลยีต่อยอดของ GPRS

3. การสื่อสารโทรศัพท์เคลื่อนที่ยุค 3 (Third Generation: 3G)

การสื่อสารโทรศัพท์เคลื่อนที่ยุค 3 เป็นยุคที่มีการสร้างระบบใหม่ให้รองรับระบบเก่าได้เรียกว่า UMTS (Universal Mobile Telecommunication Systems) โดยมุ่งหวังว่าการเข้าถึงเครือข่ายแบบไร้สายสามารถกระทำได้ด้วยอุปกรณ์หลากหลาย เช่น จากคอมพิวเตอร์หรือจากเครื่องใช้ไฟฟ้าอื่นๆ สำหรับเทคโนโลยีในระบบ 3G ที่ใช้ในประเทศไทยนั้นแต่เดิมแบ่งออกได้เป็น 2 ประเภทคือ มาตรฐาน WCDMA และมาตรฐาน CDMA2000

4. การสื่อสารโทรศัพท์เคลื่อนที่ยุค 4 (Forth Generation: 4G)

เทคโนโลยี 4G คือเครือข่ายไร้สายความเร็วสูงในการสื่อสารแบบไร้สาย โดยเครือข่ายยุคที่ 4 จะสามารถส่งผ่านข้อมูลด้วยระดับความเร็วสูงถึง 100 เมกะบิตต่อวินาที ซึ่งความเร็วของ 4G นี้สามารถให้บริการรับชมรายการโทรทัศน์ผ่านทางโทรศัพท์เคลื่อนที่ได้ นอกจากเรื่องความเร็วนั้น 4G ยังมีความสามารถรองรับการปรับเครือข่ายให้ใช้เป็นบริเวณพื้นที่กว้าง หรือทำเป็นเครือข่ายขนาดเล็กๆ แบบ Wireless LAN ได้ด้วย

ระบบโทรศัพท์เคลื่อนที่ยุคที่ 1 ถึง 4 มีความแตกต่างกันที่อัตราความเร็วในการรับส่งข้อมูลซึ่งจะมีความเร็วเพิ่มสูงขึ้น สามารถใช้ส่งได้ทั้ง

สัญญาณเสียงและสัญญาณข้อมูล ตลอดจนสัญญาณภาพนิ่งและภาพวิดีโอได้อย่างรวดเร็ว ผ่านเทคโนโลยีการเข้าถึงทั้งแบบ TDMA, FDMA, CDMA และ WCDMA

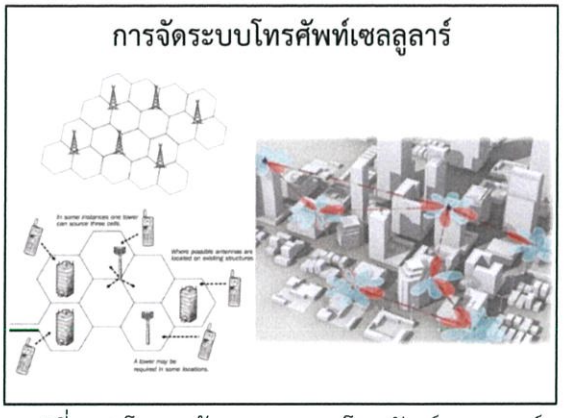
2.1.2 โทรศัพท์เคลื่อนที่ (Cellular Telephone)

วิวัฒนาการของเทคโนโลยีที่ก่อให้เกิดเป็นระบบโทรศัพท์เคลื่อนที่ เริ่มต้นจากวิทยุโทรศัพท์เคลื่อนที่ (Mobile Radiotelephone) แต่ในเวลานั้นวิทยุโทรศัพท์เคลื่อนที่ยังคงใช้งานได้แต่เฉพาะในกลุ่มจำกัด โดยใช้การส่งแบบเอเอ็ม (AM) ซึ่งมีข้อจำกัดอยู่หลายประการเช่น ข้อจำกัดด้านเครือข่ายของการสื่อสาร เนื่องจากช่องสัญญาณแต่ละช่องอนุญาตให้มีผู้ใช้บริการได้เพียงคนเดียว และข้อเสียประการสำคัญคือมีความจุของช่องสัญญาณต่ำ จึงได้มีผู้พัฒนาระบบเพื่อให้มีความจุของสัญญาณสูงขึ้น โดยใช้เทคโนโลยีแบบเอฟเอ็ม (FM) ซึ่งสามารถช่วยเพิ่มจำนวนช่องสัญญาณใช้ในการติดต่อสื่อสารได้มากขึ้น และนอกจากนั้นยังได้แบ่งช่องว่างความถี่ในพื้นที่บริการให้ใช้งานหลายช่องความถี่ และได้ได้นำความถี่ที่ใช้งานในพื้นที่อื่นกลับมาใช้งานใหม่ในอีกพื้นที่หนึ่ง จึงสามารถใช้ช่องสัญญาณเพื่อติดต่อสื่อสารได้อย่างมีประสิทธิภาพ ในปี พ.ศ. 2490 ห้องทดลองเบลล์ได้จดสิทธิบัตรของระบบนี้และได้พัฒนาระบบวิทยุโทรศัพท์เคลื่อนที่จนกลายมาเป็นระบบโทรศัพท์แบบรวงผึ้ง (Cellular Mobile Telephone System) หรือเรียกสั้นๆ ว่า โทรศัพท์เคลื่อนที่ (Cellular Telephone) โครงสร้างของระบบประกอบด้วยอุปกรณ์ทวนสัญญาณ (Repeater) จำนวนมากประกอบกันเครือข่าย โดยพื้นที่ให้บริการทั้งหมดจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า เซลล์ (Cell) แต่ละเซลล์เล็กๆจะมีรัศมีและจัดสรรความถี่ใช้งานเฉพาะเซลล์พ่วงต่อกันเป็นแบบรวงผึ้ง เนื่องจากพื้นที่ให้บริการมีขนาดเล็กจึงไม่จำเป็นต้องใช้เครื่องส่งที่กำลังส่งสูงๆ สามารถนำความถี่ซ้ำๆ ไปใช้งานได้เพิ่มมากขึ้น

2.1.2.1 การจัดการระบบโทรศัพท์เคลื่อนที่

ระบบโทรศัพท์เคลื่อนที่ได้รับการพัฒนาขึ้นมาให้เหมาะสมกับการใช้งานได้ในปัจจุบัน สามารถครอบคลุมพื้นที่ให้บริการได้กว้างมากขึ้น ขยายขอบเขตการให้การบริการได้ต่อเนื่องตามความต้องการ ในเขตพื้นที่มีประชากรหนาแน่น ซึ่งจะมีความต้องการใช้โทรศัพท์เคลื่อนที่จำนวนมากก็ต้องออกแบบ

ให้มีจำนวนเซลล์เป็นจำนวนมากขึ้นเพื่อรับรองอัตราใช้บริการแบบทราฟฟิก (Traffic) ที่เพิ่มขึ้น ส่วนในเขตที่มีประชากรไม่หนาแน่นนั้นจะมีความต้องการใช้โทรศัพท์เคลื่อนที่มีจำนวนน้อยกว่า ก็จะออกแบบให้เซลล์มีขนาดใหญ่ขึ้น แต่ละเซลล์ที่ติดกันจะใช้ย่านความถี่ที่แตกต่างกันเพื่อการนำความถี่กลับมาใช้อีก (Frequency Reuse) โดยไม่ให้เกิดสัญญาณรบกวนแทรกสอด (Interference) แต่หากต้องการใช้โทรศัพท์เคลื่อนที่เพิ่มขึ้นก็จะแบ่งจำนวนเซลล์ออกแบบเป็นเซลล์ย่อย (Cell Splitting) ให้มากขึ้นตามต้องการได้ ทำให้ระบบโทรศัพท์เคลื่อนที่สามารถนำความถี่กลับมาใช้งานอย่างมีประสิทธิภาพ ระบบโทรศัพท์เคลื่อนที่ใช้หลักการติดตั้งเครื่องรับส่งวิทยุ (Transceiver) ที่กำลังส่งต่ำๆ จำนวนมากกระจายเป็นจุดๆ ไปทั่วพื้นที่ให้บริการ จุดที่ติดตั้งเครื่องรับส่งวิทยุเหล่านี้เรียกว่า สถานีฐาน (Base Station : BS) หรืออาจเรียกว่าที่ตั้งเซลล์ (Cell Site) ซึ่งทำหน้าที่เป็นศูนย์กลางเซลล์ ลักษณะโครงสร้างระบบโทรศัพท์เคลื่อนที่ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 โครงสร้างของระบบโทรศัพท์เซลลูลาร์

โครงสร้างพื้นฐานของระบบโทรศัพท์เคลื่อนที่แต่ละเซลล์จะเชื่อมต่อถึงกัน ซึ่งแต่ละเซลล์มีรัศมีครอบคลุมกว้างหรือแคบนั้นขึ้นอยู่กับความหนาแน่นของความต้องใช้บริการโทรศัพท์เคลื่อนที่ ในพื้นที่มีความหนาแน่นไม่มากจำนวนเซลล์จะครอบคลุมพื้นที่ได้เป็นบริเวณกว้าง แต่ถ้ามีความต้องใช้

โทรศัพท์เคลื่อนที่หนาแน่นมากจำนวนเซลล์จะมีขนาดลดลงเพื่อเพิ่มการใช้ความถี่ซ้ำให้มากขึ้นและจะต้องเพิ่มสถานีฐานมากขึ้นตามไปด้วย แต่เครื่องส่งของแต่ละเซลล์จะลดกำลังส่งลงให้ครอบคลุมอยู่เฉพาะพื้นบริการเท่านั้น ระบบโทรศัพท์เคลื่อนที่นั้นต้องนำความถี่ที่ใช้แล้วกลับมาใช้อีกในเซลล์ต่างๆ ที่อยู่ห่างไกลออกไป

2.1.2.2 โครงสร้างพื้นฐานระบบโทรศัพท์เคลื่อนที่

โครงสร้างเบื้องต้นของระบบโทรศัพท์เซลลูลาร์ ประกอบด้วยส่วนประกอบหลัก ๆ คล้ายกัน โดยส่วนประกอบที่สำคัญได้แก่ เครื่องโทรศัพท์เซลลูลาร์ สถานีฐาน ชุมสายโทรศัพท์เซลลูลาร์ สายอากาศ และสายส่งสัญญาณ

1) เครื่องโทรศัพท์เซลลูลาร์ (Telephone) ประกอบด้วย หน่วยควบคุม ตัวรับส่งคลื่นวิทยุ ระบบสายอากาศ และแบตเตอรี่ ตัวเครื่องโทรศัพท์เซลลูลาร์สามารถตรวจสอบระบบสัญญาณในบริเวณที่เครื่องอยู่ได้ มีโปรแกรมติดต่อกับสถานีฐานและชุมสายโทรศัพท์เซลลูลาร์

2) SIM หรือ ซิมการ์ดเป็นสมาร์ทการ์ดที่จัดเก็บข้อมูล เกี่ยวกับผู้ใช้และสมุดโทรศัพท์

3) สถานีฐาน (Base Station Subsystem) เป็นสถานีประจำเซลล์ระบบโทรศัพท์เซลลูลาร์ทำหน้าที่เชื่อมต่อสัญญาณระหว่างตัวเครื่องโทรศัพท์เซลลูลาร์กับชุมสายโทรศัพท์เซลลูลาร์ ภายในสถานีฐานประกอบด้วย หน่วยควบคุมตัวประมวลผลสัญญาณคลื่นวิทยุ ตัวรับเครื่องวิทยุ ระบบสายอากาศและแหล่งจ่ายไฟ

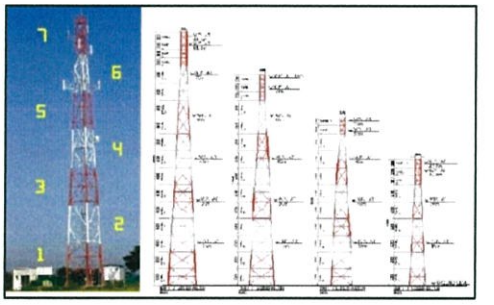
4) ชุมสายโทรศัพท์เซลลูลาร์หรือเครือข่ายแบ็คโบน (Network and Switch Subsystem) เป็นศูนย์กลางการควบคุมการทำงานของระบบควบคุมทั้งสถานีฐาน และ ตัวเครื่องเซลลูลาร์ มีการต่อเชื่อมระบบกับชุมสายโทรศัพท์ต่างๆ ทำหน้าที่ช่วยในการควบคุม ขั้นตอนการโทรศัพท์ การเก็บข้อมูลการโทรศัพท์ การเปลี่ยนแปลงการ

เชื่อมต่อเซลล์ และ ตัดต่อระบบโทรศัพท์เซลลูลาร์กับระบบโทรศัพท์อื่นๆ

2.2 การติดตั้งเสาสัญญาณโทรศัพท์เคลื่อนที่

2.2.1 เสาสัญญาณโทรคมนาคม (Telecommunication Towers)

เสาสัญญาณโทรคมนาคมมีการใช้งานในหลายภาคส่วนที่แตกต่างกันเช่นมือถือ, อินเทอร์เน็ต, โทรทัศน์, การเดินเรือและการส่งสัญญาณวิทยุ เสาสัญญาณโทรคมมีอุปกรณ์รับและส่งสัญญาณชนิดต่างๆ ถูกติดตั้งบนยอดเสา เพื่อให้สัญญาณกระจายและครอบคลุมให้ได้พื้นที่มากขึ้น โดยอุปกรณ์ติดตั้งเป็น Mobile Antenna, Microwave Antenna เป็นหลักและเนื่องจากเสามีความสูงจึงต้องมีการทาสีแดง-ขาว ให้สังเกตได้ง่ายเพื่อความปลอดภัยด้านการบิน โดยทาเป็น 7 ชั้น แดง-ขาวสลับกันไปเฉลี่ยตามความสูงของเสา ดังรูปที่ 2.3

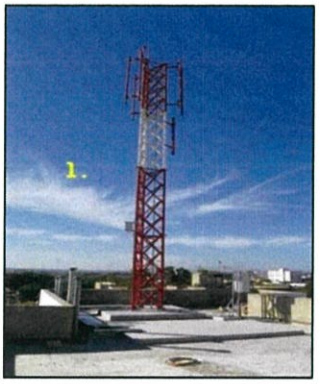


รูปที่ 2.3 การทาสีของเสาที่ระดับความสูงต่างๆ

ชนิดของเสาแบ่งตามพื้นที่ติดตั้งและลักษณะของเสา

1) Rooftop Tower: เสาตั้งบนดานฟ้าหรือหลังคาของตึก ส่วนมากตั้งค่อนข้างกึ่งกลางดานฟ้าตึก เสาเดียวได้หลาย sector antenna จะเจอเสาลักษณะนี้ในเขตเมืองเพราะประหยัดพื้นที่ตั้งเสา พื้นที่ครอบคลุมสัญญาณไม่มากนัก เสา Rooftop แสดงดังรูปที่ 2.4

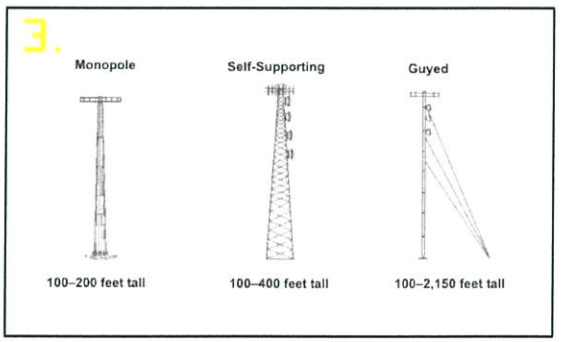
2) Angular Tower / Pole Tower: เสาตั้งบนมุมของตึก ส่วนมากหนึ่งเสาจะได้หนึ่งเซกเตอร์ของสายอากาศ จะเจอเสาลักษณะนี้ในเขตเมือง เพราะประหยัดพื้นที่ตั้งเสา พื้นที่ครอบคลุมสัญญาณไม่มากนัก เสา Pole แสดงดังรูปที่ 2.5



รูปที่ 2.4 เสาสัญญาณ Rooftop

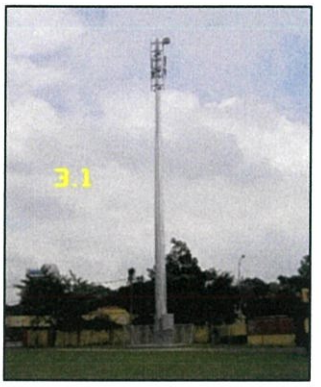
รูปที่ 2.5 เสา Pole

3) Ground Base Tower / On Ground Tower / Green field Tower: เสาที่ตั้งบนพื้นโลกปกติ เสาพวกนี้มีความสูงตั้งแต่ 30 - 650 เมตรเจอได้ทั่วไปมีลักษณะเสาดังรูปที่ 2.6 พื้นที่ครอบคลุมสัญญาณของเสากว้าง 1-3 กิโลเมตร ตามความสูงของเสา ส่วน Ground tower จะแบ่งตามลักษณะเสาได้ดังนี้



รูปที่ 2.6 เสา Green field

3.1) Monopole เป็นเสาขนาดเล็ก 30-60 เมตร แบบที่ตั้งพื้นปกติไม่นิยมในประเทศไทย เป็นเสาต้นเดี่ยวๆ มีบันไดให้ปีนด้านข้างเสา เสา Monopole แสดงดังรูปที่ 2.7

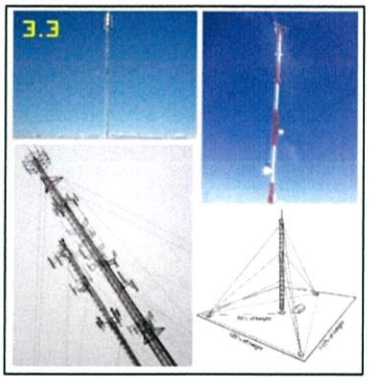


รูปที่ 2.7 เสาสัญญาณ Monopole

3.2) Self-Support เป็นเสาที่นิยมให้ในไทย มีขนาด 30-120 เมตร มีหลายแบบตามการใช้งาน มีทั้ง 3ขา 4ขา ดังรูปที่ 2.8 มีบันไดให้ปีนด้านในเสา และมีแพรตฟอร์มให้พักเป็นระยะ มีความแข็งแรงมั่นคงมาก

3.3) Guyed เป็นเสาที่นิยมให้ในไทย มีขนาด 30-650 เมตร เสาขนาดเล็กมีสายสลิงยึดเสาเอาไว้ให้มั่นคง สามารถตั้งแบบเตี้ยๆ 10-20 เมตร โดยตั้งบนตึกหรือตั้งเป็นเสาที่สูงมากๆได้ เสา Guyed แสดงดังรูปที่ 2.9

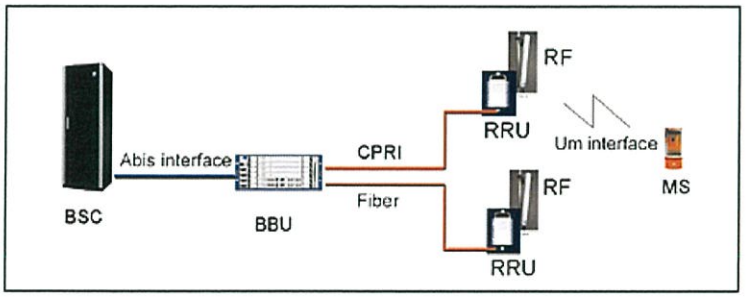
รูปที่ 2.8 เสาสัญญาณ Self-Support



รูปที่ 2.9 เสาสัญญาณ Guyed

2.2.2 อุปกรณ์ภายในตู้สถานีฐาน

ในสถาปัตยกรรมของสถานีฐานอุปกรณ์ส่งสัญญาณความถี่วิทยุส่วนใหญ่จะถูกติดตั้งอยู่ในตู้คาบิเนทบริเวณฐานของเสาส่ง การส่งสัญญาณจะทำผ่านสาย feeder ที่เป็นเคเบิลโคแอกเชียลเพื่อส่งสัญญาณอะนาลอกและสายไฟที่ส่งกระแสไฟเลี้ยงไปยังชุดขยายกำลัง ก่อนจะเชื่อมต่อไปยังจานสายอากาศแบบพาสซีฟด้วยสายโคแอกเชียลระยะสั้น แสดงการทำงานระหว่างอุปกรณ์ภายในสถานีฐานดังรูปที่ 2.10



รูปที่ 2.10 การทำงานระหว่างอุปกรณ์ภายในสถานีฐาน

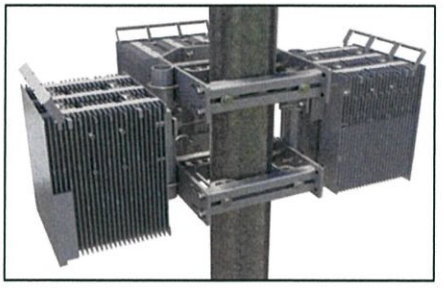
1) BBU หรือ Base Band Unit ทำหน้าที่เป็น Modem สื่อสารกับชุดเครื่องส่งวิทยุหรือ RRU และควบคุมการทำงานของ RRU จะติดตั้งอยู่ที่พื้นล่าง มีหน้าที่ดำเนินการกับฟังก์ชันวิทยุบนโดเมนสัญญาณดิจิตอล อุปกรณ์ส่วนนี้จะต่อเข้ากับเครือข่ายแบ็กฮอลล์ผ่านระบบสื่อสารด้วยแสงหรือระบบไมโครเวฟเป็นอุปกรณ์ที่มีพอร์ตส่งต่อไปที่ RRU บนเสา หรือ RRU ที่ติดตั้งห่างไกลออกไปได้

โดยปกติ BBU หนึ่งตัวจะควบคุม RRU ได้สามหรือหกตัว BBU มีอีกชื่อเรียกว่า Master Unit มีขนาด 2U หรือ 3U จะต่อเข้ากับ Switch IP LAN เพื่อส่งกลับข้อมูลไปยังชุมสายเพื่อกระจายไปยังสถานีอื่นๆการ์ดที่ใส่เข้าไปใน BBU ของเครือข่าย 3G และ 4G จะไม่เหมือนกัน แต่บางผู้ผลิตจะผลิตให้ BBU ของ 3G จะสามารถพัฒนาเป็นเครื่อข่าย 4G ได้โดยการเปลี่ยนเฉพาะการด์ใน BBU ไม่จำเป็นต้องเปลี่ยนทั้ง BBU ตัวอย่าง BBU แสดงดังรูปที่ 2.11

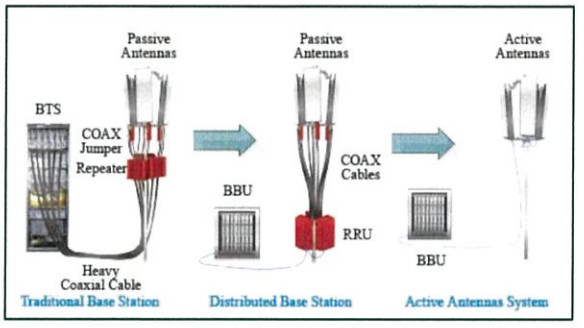
รูปที่ 2.11 BBU หรือ Base Band Unit

2) RRU หรือ Remote Radio Unit จะติดตั้งอยู่ถัดจากจานสายอากาศ มีหน้าที่ดำเนินการกับฟังก์ชันที่เกี่ยวข้องกับความถี่วิทยุบนโดเมนสัญญาณอะนาลอก มีการแปลงสัญญาณดิจิตอลไปเป็นอะนาลอกเพื่อขับจานสายอากาศแบบพาสซีฟผ่านสายเคเบิลโคแอกเชียล ดังนั้นสายที่ลากจากด้านล่างขึ้นไปยังเสาส่งจึงประกอบด้วยเคเบิลใยแก้วนำแสงกับตัวนำทางไฟฟ้า ซึ่งโดยปกติจะถูกห่อหุ้มรวมกันอยู่ภายในเคเบิลที่มีการป้องกันอีกชั้นหนึ่ง มีหน้าที่แปลงสัญญาณจาก Modem หรือ BBU ส่วนมากจะติดตั้งอยู่บนเสาสูงเพื่อออกอากาศความถี่วิทยุที่ความถี่มือถือ 850MHz, 900MHz, 1800MHz หรือ 2100MHz และเชื่อมต่อกับเสาอากาศเพื่อส่งสัญญาณต่อไปที่โทรศัพท์มือถือ และรับสัญญาณโทรศัพท์มือถือกลับมาที่ BBU เพื่อส่งต่อข้อมูลไปยังชุมสาย ปัจจุบันจะติดตั้งอยู่ใกล้กับเสาอากาศมีลักษณะเป็นกล่องสี่เหลี่ยม และมีครีบระบายความร้อน ดังรูปที่ 2.12 นอกจากนี้ด้วยเทคโนโลยีล่าสุด ผู้ผลิตได้มีการพัฒนาจานสายอากาศจากแบบพาสซีฟไปเป็นแบบแอกทีฟด้วยการรวม RRU เข้าไปในจาน

สายอากาศ ทำให้การเชื่อมโยงจาก BBU ที่อยู่ด้านล่างของเสาด้วยเคเบิลใยแก้วนำแสงกับสายไฟมีการต่อตรงไปยังจาน สำหรับการเชื่อมต่ออุปกรณ์สถานีฐานไปยังจานสายอากาศแบบต่าง ๆ จะแสดงได้ดังรูปที่ 2.13



รูปที่ 2.12 RRU หรือ Remote Radio Unit



รูปที่ 2.13 การเชื่อมต่ออุปกรณ์สถานีฐานไปยังจานสายอากาศ

3) ตู้โนดบี (NodeB) เป็นตู้สำหรับการทำงานทางด้านคลื่นความถี่โดยเฉพาะ เป็นตู้ที่ใช้ใส่อุปกรณ์ที่สถานีฐาน ปัจจุบันมีมากมายหลายประเภทตามลักษณะของงาน ภายในตู้โนดบีจะประกอบไปด้วย BBU, Battery, Circuit Breaker, Rectifier โมดูล, Alarm, IPRAN และ DCDU โดยตู้โนดบีแสดงดังรูปที่ 2.14

รูปที่ 2.14 ตู้โนดบีและอุปกรณ์ภายในตู้

2.3 ภาษาอาร์

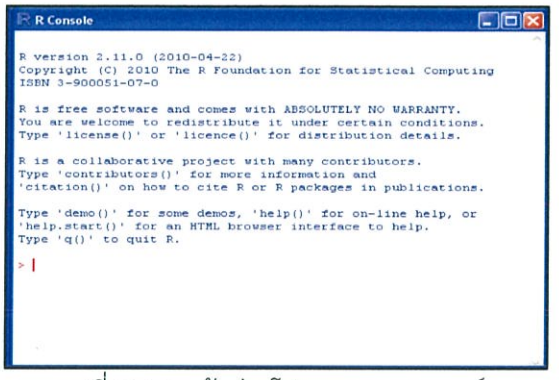
อาร์ (R) หรือ ภาษาอาร์ เป็นภาษาโปรแกรมบนคอมพิวเตอร์สำหรับวิเคราะห์ทางด้านสถิติภาษาอาร์เป็นที่นิยมใช้ในหมู่นักสถิติในด้านการพัฒนาซอฟต์แวร์สถิติ และการวิเคราะห์ข้อมูล ภาษาอาร์เป็นซอฟต์แวร์ฟรีในรูปแบบของโอเพนซอร์ซซึ่งสามารถนำไปใช้งานได้ฟรี และแจกจ่ายได้ฟรี โดยได้รับแรงบันดาลใจมาจากภาษาเอส มีจุดประสงค์สำหรับการคำนวณและนำเสนอกราฟิกทางสถิติ ถูกสร้างและเผยแพร่เป็นครั้งแรกในปี 1993 โดย Ross Ihaka และ Robert Gentleman สัญลักษณ์ของภาษาอาร์แสดงดังรูป 2.15

รูปที่ 2.15 สัญลักษณ์ของภาษาอาร์

ด้วยความเฉพาะทาง ภาษาอาร์จึงแตกต่างจากภาษาโปรแกรมสำหรับงานทั่วไป (general-purpose programming language) อย่างเช่น C, Python ซึ่งเอกลักษณ์สำคัญอย่างหนึ่งของภาษาอาร์คือการเป็นภาษาโปรแกรมเชิงอาร์เรย์ (array language) ที่เน้นการคำนวณข้อมูลพร้อมกันเป็นกลุ่ม ใช้นำมาประมวลผลข้อมูล (Big data) ในเชิงสถิติ

2.3.1 การติดตั้งโปรแกรมอาร์

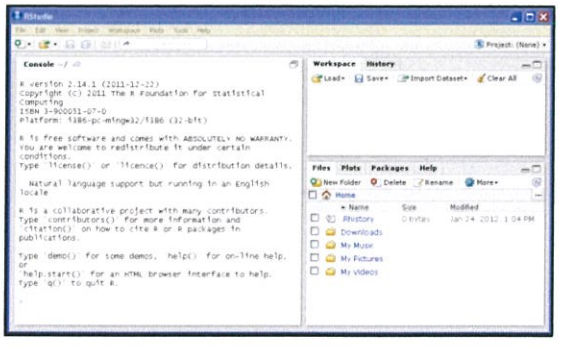
โปรแกรมอาร์สามารถดาวน์โหลดได้ฟรีจาก http://cran.r-project.org เมื่อติดตั้งเสร็จแล้วเรียกโปรแกรมอาร์ขึ้นมาจะเห็นเป็นหน้าต่าง R console ดังรูปที่ 2.16 โดยเครื่องหมาย “>” แสดงว่าพร้อมรับคำสั่ง



รูปที่ 2.16 หน้าต่างโปรแกรมภาษาอาร์

2.3.2 การติดตั้ง Rstudio

การใช้งานภาษาอาร์ผ่าน R console เป็นการสั่งงานผ่านทาง command line หากต้องการใช้งานภาษาอาร์ในลักษณะที่มีหน้าจอต่างๆ ให้ พร้อมสำหรับการแก้ไขชุดคำสั่งต่างๆหน้าจอ แสดงผล หน้าจอแสดงตัวแปรต่างๆ สามารถติดตั้ง Rstudio เพิ่มเติมได้ การใช้ Rstudio จะช่วยอำนวยความสะดวกในเวลาที่ต้องใช้ชุดคำสั่งเดิมหลายๆครั้งสามารถนำชุดคำสั่งทั้งหมดเก็บเป็นไฟล์โปรแกรมอาร์ และเรียกมาใช้งานผ่าน Rstudio ได้ทีละคำสั่งได้ โดยเมื่อทำการติดตั้งเสร็จแล้วจะแสดงหน้าต่างโปรแกรม Rstudio ดังรูป 2.17



รูปที่ 2.17 หน้าต่างโปรแกรม Rstudio

ภาษาอาร์มีการแบ่งประเภทของข้อมูลออกเป็น 2 ประเภทคือข้อมูลเดี่ยวและข้อมูลชุด

2.3.3 ประเภทข้อมูลเดี่ยว

ข้อมูลเดี่ยวคือข้อมูลขนาดเล็กที่สุดที่ไม่สามารถแบ่งย่อยลงไปได้อีก ไม่สามารถนำไปใช้คำนวณทางสถิติได้แบ่งออกเป็น 3ประเภท

1) ข้อมูลตัวเลข เนื่องจากภาษาอาร์เป็นภาษาที่เน้นการนำไปใช้งานทางสถิติเป็นหลัก การเก็บตัวเลขแบบจุดทศนิยมลอยตัว ในภาษาอาร์เรียกข้อมูลชนิดนี้ว่า numeric อย่างไรก็ตามภาษาอาร์ยังมีข้อมูลตัวเลขอีกสองประเภทให้เลือกใช้เพิ่ม เติมตามความเหมาะสมเช่นกัน ซึ่งก็คือ

1.จำนวนเต็ม (integer) สร้างได้โดยเขียนห้อยตัว L ไว้ข้างหลังตัวเลข เช่น 100L

2.จำนวนเชิงซ้อน (complex) สร้างได้โดยเขียนห้อยตัว i ไว้ข้างหลังตัวเลข เช่น 1i, 3+4i

ลำดับของตัวเลขในภาษาอาร์ เรียงจากต่ำไปสูงได้แก่ จำนวนเต็ม จำนวนทศนิยม และจำนวนเชิงซ้อน หากมีการใช้งานตัวเลขคละประเภทกัน อาร์จะแปลงตัวเลขที่อยู่ในชั้นต่ำกว่าขึ้นไปเป็นชั้นที่สูงกว่าเสมอ

2) ข้อมูลตัวอักษร ภาษาอาร์ไม่แบ่งแยกข้อมูลตัวอักษรเป็นหลายระดับ แต่จะมองว่าตัวอักษรชุดหนึ่งไม่ว่าจะยาวเท่าใดคือหนึ่งข้อความเสมอ และเรียกข้อมูลชนิดนี้ว่า character การประกาศข้อความสามารถใช้ได้ทั้งอัญประกาศเดี่ยว (') หรืออัญประกาศคู่ (") ล้อมรอบข้อความที่ต้องการ

3) ข้อมูลตรรกะ การแสดงข้อมูลตรรกะ (logical) ว่าถูกหรือผิด จะใช้คำว่า TRUE กับ FALSE ตามลำดับ ไม่สามารถใช้ตัวเลขทดแทนได้ นอกจากนี้สามารถป้อนแค่ตัวอักษรตัวแรก (T,F) เข้าไปได้เช่นกัน ภาษาอาร์ยังมีข้อมูลตรรกะแบบพิเศษอีกตัวหนึ่งเพิ่มเข้ามา คือ ข้อมูลว่าง (NA) ซึ่งจะเกิดขึ้นเมื่อพยายามเข้าถึงข้อมูลในตำแหน่งที่ไม่มีข้อมูล หรือพยายามคำนวณค่าทั้งที่ไม่มีข้อมูลอยู่

2.3.4 ประเภทข้อมูลชุด

ข้อมูลชุดคือข้อมูลที่เก็บรวบรวมข้อมูลเดี่ยวหลายๆ ตัวมาไว้ในที่เดียวกัน จึงทำให้สามารถนำไปวิเคราะห์เชิงสถิติต่อได้

1) เวกเตอร์

ภาษาอาร์ให้ความสำคัญกับการคำนวณตัวเลขหลายตัวพร้อมกันเป็นอย่างมาก โดยข้อมูลเดี่ยวทุกชนิดที่สร้างขึ้นในภาษาอาร์จะถูกห่อหุ้มด้วยสิ่งที่เรียกว่าเวกเตอร์ (vector) ซึ่งเทียบในภาษาอื่นได้ว่าเป็นอาเรย์ที่มีความยาวไม่จำกัด แม้ว่าข้อมูลชุดนั้นจะมีความยาวเพียงแค่หนึ่งหน่วย การนำเวกเตอร์หลายชุดมารวมกันจะเป็นการนำเวกเตอร์มาต่อกันเสมอ ซึ่งจะได้ผลลัพธ์เป็นเวกเตอร์เพียงชุดเดียวที่ไล่สมาชิกจากเวกเตอร์ตัวแรกไปจนจบ แล้วจึงไล่สมาชิกจากเวกเตอร์ตัวถัดไปเรื่อยๆ โดยฟังก์ชันที่ทำหน้าที่ประกาศเวกเตอร์หรือนำเวกเตอร์มาต่อกันคือ c() ดังตัวอย่างโปรแกรมอาร์ต่อไปนี้

```
> y <- c(1,2,3,4)
> y
[1] 1 2 3 4
```

2.3.5 ตัวแปรและประเภทข้อมูล

แม้ภาษาอาร์จะทำให้สามารถดัดแปลงและสร้างประเภทข้อมูลชนิดใหม่ๆ ขึ้นมาได้ แต่โดยส่วนใหญ่จะมีข้อมูลพื้นฐานเพียงไม่กี่ประเภทเท่านั้น การทดสอบว่าข้อมูลดังกล่าวถูกจัดอยู่ในประเภทใด สามารถใช้ฟังก์ชัน class() เพื่อตรวจสอบประเภทของข้อมูลนั้นได้ เช่น

```
> class("hello")
[1] "character"
```

การถามประเภทข้อมูลของเวกเตอร์ด้วยฟังก์ชัน class() จะไม่รับคำตอบกลับมาว่าคือเวกเตอร์ แต่จะได้รับคำตอบว่าสมาชิกทุกตัวในเวกเตอร์นี้เป็นข้อมูลเดี่ยวประเภทใด นี่ทำให้เมื่อนำเวกเตอร์ของข้อมูลคละชนิดกันมารวมกัน สมาชิกบางตัวจะถูกแปลงประเภทข้อมูลขึ้นไปเป็นแบบที่สามัญกว่าทันที

```
> z <- c(1,2,3,4)
> class(z)
[1] "numeric"
```

การเข้าถึงสมาชิกในเวกเตอร์ สามารถใช้สัญลักษณ์วงเล็บปีกแข็งล้อมรอบตัวเลขตำแหน่งที่ต้องการ โดยอาร์จะเริ่มนับสมาชิกตัวแรกด้วยเลขหนึ่ง (one-based indexing) และการเข้าถึงสมาชิกตัวที่สูงเกินกว่าขนาดของเวกเตอร์จะได้ผลลัพธ์เป็นข้อมูลว่าง เช่นนี้

```
> z <- c(1,2,3,4)
> z[3]
[1] 3
> z[10]
 [1] NA
```

นอกจากนี้ยังสามารถเข้าถึงข้อมูลย่อยพร้อมๆ กันหลายตัวได้ โดยส่งเวกเตอร์ของตำแหน่งทั้งหมดที่ต้องการเข้าไป ซึ่งสามารถเขียนย่อช่วงตำแหน่งที่ติดกันโดยใช้สัญลักษณ์โคลอน (:) หรือจะส่งเวกเตอร์ตรรกะเพื่อบอกว่าจะเลือกสมาชิกตัวไหนบ้างก็ได้ ตามตัวอย่างต่อไปนี้

```
> z[1:3]
[1] 1 2 3
```

การคำนวณทางคณิตศาสตร์ระหว่างเวกเตอร์ จะเป็นการจับคู่สมาชิกแต่ละตัวในเวกเตอร์แล้วคำนวณตามเครื่องหมายคณิตศาสตร์ที่ให้ไว้ ถ้าเวกเตอร์มีขนาดไม่เท่ากัน เวกเตอร์ตัวที่สั้นกว่าจะถูกนำมาใช้จับคู่ซ้ำเรื่อยๆ ดังตัวอย่างต่อไปนี้

```
> z <- c(1,2,3,4)
> z - 1
[1] 0 1 2 3
> z * c(1,-1)
[1]  1 -2  3 -4
```

2.3.6 ประกาศตัวแปร

การประกาศตัวแปรในภาษาอาร์จะใช้สัญลักษณ์ที่แตกต่างจากภาษาโปรแกรมโดยทั่วไป ซึ่งก็คือสัญลักษณ์ลูกศรซ้าย “<-“ ดังเช่นตัวอย่างการประกาศตัวแปรนี้

```
> x <- 42
> y <- "Hello"
```

นอกจากนี้ในภาษาอาร์ยังสามารถใช้สัญลักษณ์เท่ากับ (=) เพื่อประกาศตัวแปรได้เช่นกัน ความแตกต่างคือสัญลักษณ์เท่ากับ (=) ถูกออกแบบมาเพื่อใช้ประกาศพารามิเตอร์/อาร์กิวเมนต์ของฟังก์ชันเป็นหลัก การใช้สัญลักษณ์ลูกศรซ้าย“<-“ เพื่อประกาศพารามิเตอร์/อาร์กิวเมนต์ฟังก์ชันอาจทำให้โปรแกรมทำงานผิดพลาดได้ ทั้งนี้ในภาษาอาร์ยังสามารถให้ใช้สัญลักษณ์จุด . เป็นส่วนผสมในชื่อตัวแปรได้ สามารถขอดูตัวแปรทั้งหมดที่สร้างขึ้นผ่านฟังก์ชัน ls() และลบตัวแปรได้ผ่านฟังก์ชัน rm()

2.3.7 ข้อมูลชุดสำหรับคำนวณทางคณิตศาสตร์

หากต้องการเก็บข้อมูลที่เทียบเท่ากับอาเรย์สองมิติในภาษาอื่น ภาษาอาร์จะเรียกข้อมูลชนิดนั้นว่าเมทริกซ์ (matrix) การจะสร้างข้อมูลชนิดนี้ได้ต้องเตรียมข้อมูลให้อยู่ในรูปของเวกเตอร์ก่อน แล้วจึงเปลี่ยนประเภทข้อมูลเป็นเมทริกซ์ ดังนี้

```
> z <- c(1,2,3,4)
> A <- matrix(z, nrow=2, ncol=2)
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> B <- matrix(z, nrow=2, ncol=2, byrow=TRUE)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

เมื่อต้องการเข้าถึงสมาชิกในข้อมูลประเภทเมทริกซ์ จะต้องเขียนสัญลักษณ์ลูกน้ำกำกับด้วยดังนี้

```
> A[1,2]
[1] 3
> B[,2]
[1] 2 4
```

การดำเนินการทางคณิตศาสตร์โดยใช้เครื่องหมายธรรมดาทั่วไป (+,-,*,/) จะเป็นการนำสมาชิกในข้อมูลทั้งสองชุดมาจับคู่กันตัวต่อตัวแล้วดำเนินการทางคณิตศาสตร์แบบธรรมดาเท่านั้น หากต้องการดำเนินการทางคณิตศาสตร์แบบเมทริกซ์ ต้องใช้เครื่องหมายคณิตศาสตร์สำหรับเมทริกซ์โดยเฉพาะ เช่น การคูณเมทริกซ์ (%*%) ดังตัวอย่างต่อไปนี้

```
> A %*% B
     [,1] [,2]
[1,]   10   14
[2,]   14   20
```

2.3.8 กรอบข้อมูล

กรอบข้อมูล (data frame) คือข้อมูลชุดแบบตารางเช่นเดียวกับตารางในฐานข้อมูล กล่าวคือ แต่ละหลักของตารางจะบอกว่าหลักนี้ใช้เก็บข้อมูลเรื่องใด ส่วนแต่ละแถวของตารางก็คือข้อมูลหนึ่งระเบียน (record) วิธีสร้างกรอบข้อมูล เริ่มจากการเตรียมข้อมูลแต่ละหลักด้วยเวกเตอร์ โดยตรวจสอบให้แน่ใจว่าเวกเตอร์ทุกตัวมีความยาวเท่ากัน แล้วจึงนำมารวมกันด้วยฟังก์ชัน data.frame() ดังนี้

```
> handle <- c("Sheldon","Leonard","Penny")
> year <- c(1980,1980,1985)
> iq <- c(187,173,NA)
> big.bang <- data.frame(handle, yob=year, iq)
```

```
> big.bang
   handle  yob  iq
1 Sheldon 1980 187
2 Leonard 1980 173
3  Penny 1985  NA
```

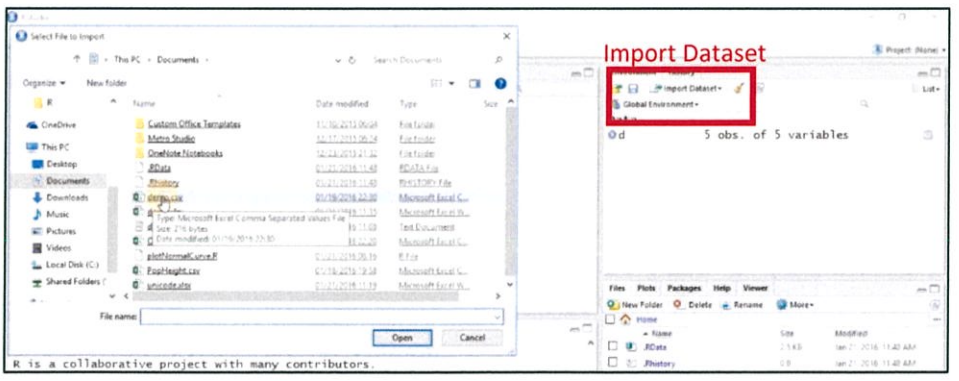
เนื่องจากรูปแบบการเก็บข้อมูลนี้ มีลักษณะคล้ายกันกับการเก็บโดยเมทริกซ์ การเข้าถึงข้อมูลแต่ละแถวสามารถทำได้ในทำนองเดียวกันกับเมทริกซ์ และ การเข้าถึงข้อมูลแต่ละหลักสามารถใช้เครื่องหมาย $ แล้วตามด้วยชื่อของเขตข้อมูลที่ต้องการ

2.3.9 การนำเข้าข้อมูล

การนำเข้าข้อมูลมาเป็นกรอบข้อมูล (data frame) ต้องเป็นไฟล์นามสกุล .CSV (Comma-Separated Value) จึงจะสามารถอ่านไฟล์เข้ามาได้ โดยใช้คำสั่ง read.csv(ตำแหน่งที่ตั้งของไฟล์) ดังตัวอย่าง

```
> data <- read.csv("C:/Users/computer /Documents/Demo.csv")
```

หรืออาจจะคลิกที่ import dataset แล้วกดเลือกไฟล์ นามสกุล .csv ดังรูปที่ 2.18



รูปที่ 2.18 การนำเข้าข้อมูลโดยวิธีการ Import Dataset

นอกจากนี้อาจเลือกลบแถวที่ข้อมูลขาดหายหรือมีข้อมูลเป็น NA ทิ้ง เพื่อความสะดวกในการคำนวณข้อมูลในภายหลัง ซึ่งทำได้โดยใช้คำสั่ง na.omit() นอกจากนี้หากต้องการอ่านไฟล์ที่เป็นไฟล์นามสกุลเอกเซลสามารถทำได้โดยการติดตั้ง XLConnect แพจเกจเพื่ออ่านไฟล์ได้ด้วย

2.3.10 การคำนวณข้อมูลทางสถิติเบื้องต้น

เมื่อต้องการเริ่มวิเคราะห์ข้อมูลชุดหนึ่งๆ วิธีการที่มีประสิทธิภาพคือการดูข้อสรุปที่ถูกสรุปออกมาจากข้อมูลชุดนั้น โดยหลักๆ แล้วอาจสนใจค่าสรุปโดยดูที่ ตำแหน่งข้อมูล ซึ่งตำแหน่งข้อมูลจะแสดงว่าข้อมูลดังกล่าวมีตำแหน่งสำคัญๆ ใดบ้าง เช่น ค่ามากที่สุดหรือค่าน้อยสุด ค่ากลางข้อมูล และควอร์ไทล์ โดยค่ากลางข้อมูลคือค่าที่แสดงว่าข้อมูลชุดนั้นมีค่าลู่เข้าสู่ตรงกลางที่เท่าใด ซึ่งอาจแบ่งย่อยได้เป็นอีก 3 ประเภท คือ

1ค่าเฉลี่ย (mean) หาได้จากการนำทุกค่าในข้อมูลมาบวกรวมกัน แล้วจึงหารด้วยจำนวนข้อมูลที่ใช้

2) มัธยฐาน (median) หาได้จากการเรียงข้อมูลจากน้อยไปมากตามลำดับ แล้วดูว่าที่ตำแหน่งตรงกลางของข้อมูลมีค่าเป็นเท่าใด

3) ฐานนิยม (mode) หาได้จากการนับว่าค่าใดปรากฏซ้ำมากที่สุดในข้อมูล

ในภาษาอาร์สามารถเรียกฟังก์ชัน summary() เพื่อดูภาพรวมตำแหน่งข้อมูลต่างๆ ได้

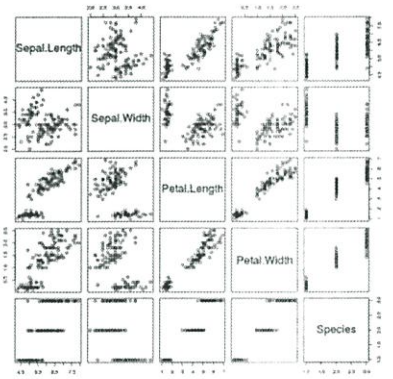
2.3.11 พล็อตข้อมูล

จุดแข็งสำคัญของภาษาที่เน้นการใช้งานเชิงสถิติอย่างอาร์คือ การพล็อตข้อมูลขึ้นมา การพล็อตข้อมูลนอกจากจะช่วยให้เห็นภาพว่าข้อมูลโดยรวมมีลักษณะอย่างไรแล้ว ยังช่วยให้ไม่เข้าใจข้อมูลผิดไปเพียงเพราะเลือกดูแค่ค่าเฉลี่ยที่สรุปมาจากข้อมูลทั้งหมดเท่านั้น

2.3.11.1 พล็อตประชันทุกเขตข้อมูล คือการมองภาพความสัมพันธ์ทั้งหมดอย่างคร่าวๆ เป็นการนำข้อมูลทุกๆ คู่ในแต่ละหลักของกรอบข้อมูลมาประชันกัน แล้วพล็อตแต่ละคู่ของข้อมูลแบบกระจาย (scatter plot) การพล็อตนี้สามารถทำได้ผ่านฟังก์ชัน plot() ตัวอย่างต่อไปนี้เมื่อมีข้อมูลดอกกุหลาบสามารถแสดงการพล็อตประชันทุกเขตข้อมูลดังรูปที่ 2.19

```
> Rose
    Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
1            5.1         3.5          1.4         0.2           setosa
...
150          5.9         3.0          5.1         1.8        virginica
```

> plot(Rose)

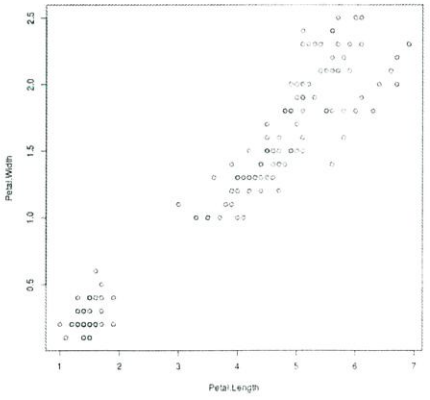


รูปที่ 2.19 การพล็อตประชันทุกเขตข้อมูลของข้อมูลดอกกุหลาบ

2.3.11.2 พล็อตกระจาย (Scatter Plot)

จากรูปที่ 2.19 หากต้องการให้ความสนใจที่ความยาวกลีบดอก (Petal.Length) ต่อความกว้างกลีบดอก (Petal.Width) จะเลือกพล็อตกระจายโดยใช้เขตข้อมูลเพียง 2 เขตนี้เท่านั้นแสดงการพล็อตกระจายดังรูปที 2.20 พร้อมทั้งสามารถระบุชื่อกราฟและชื่อแกน x และแกน y ด้วยคำสั่ง > plot(dataX, dataY, title="The title", xlab="X-axis label", ylab= "Y-axis label") ดังตัวอย่างต่อไปนี้

> plot(Petal.Width ~ Petal.Length)



รูปที 2.20 การพล็อตกระจายข้อมูลดอกกุหลาบ โดยสนใจความกว้างต่อความยาวกลีบดอก

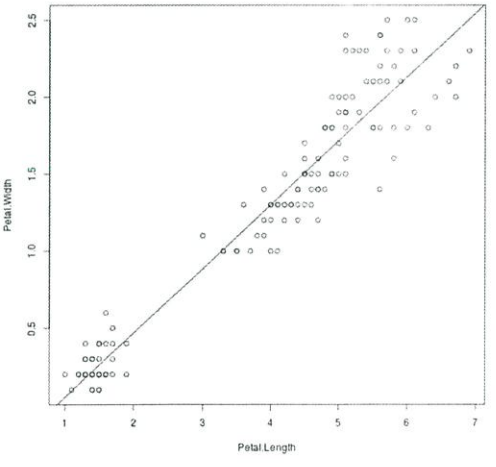
2.3.11.3 การลากโมเดลเชิงเส้น

จากรูปที่ 2.20 ถ้าต้องการหาโมเดลเชิงเส้นของข้อมูลดังกล่าว เพื่อทำนายแนวโน้มว่าหากมีข้อมูลใหม่ๆ เพิ่มขึ้นมา ข้อมูลเหล่านั้นจะเกิดบริเวณใด การคำนวณดังกล่าวสามารถทำได้ผ่านฟังก์ชัน lm() ดังนี้

```
> lm(Petal.Width ~ Petal.Length)
Call:
lm(formula = Petal.Width ~ Petal.Length, data =Rose)
Coefficients:
 (Intercept)  Petal.Length
     -0.3631         0.4158
```

หากต้องการลากเส้นดังกล่าวลงไปบนพล็อต ต้องมีภาพของการพล็อตเดิมอยู่ก่อนแล้ว หลังจากนั้นจึงเรียกฟังก์ชัน abline() เพื่อลากเส้นตรงจากสัมประสิทธิ์ที่คำนวณได้ จะแสดงการลากโมเดลเชิงเส้นดังรูปที่ 2.21 ดังนี้

```
> plot(Petal.Width ~ Petal.Length)
> abline(lm(Petal.Width ~ Petal.Length))
```



รูปที่ 2.21 โมเดลเชิงเส้นของความกว้างต่อความยาวกลีบดอก

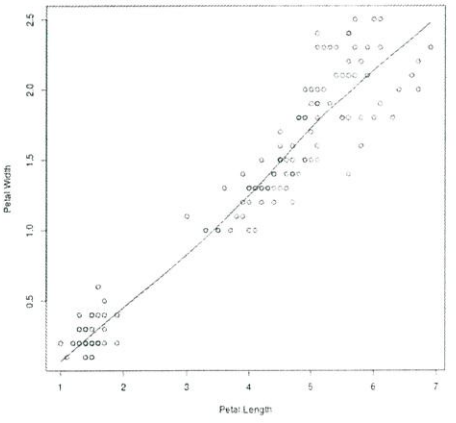
2.3.11.4 ลากโมเดลไม่เชิงเส้น

หากข้อมูลที่สนใจมีแนวโน้มว่าจะไม่เป็นเชิงเส้น ภาษาอาร์ก็มีวิธีการคำนวณโมเดลแบบต่างๆ ให้เลือกใช้ได้หลายวิธี สำหรับวิธีที่ง่ายต่อการเรียกใช้งานที่สุด คือการประมาณค่าถดถอยแบบท้องถิ่น (local regression หรือ LOESS) ซึ่งจะสร้างโมเดลสมการพหุนามโดยอาศัยข้อมูลแต่ละส่วนในบริเวณใกล้เคียงมาปรับค่า สามารถเรียกดูผลลัพธ์จุดตัดต่างๆ บนเส้นโค้งของโมเดลที่คำนวณได้ดังนี้

```
> loess.smooth(Petal.Length, Petal.Width)
$x
 [1] 1.000000 1.120408 1.240816 1.361224 1.481633
...
$y
 [1] 0.06892099 0.11545407 0.16200259 0.20869061 0.25544166
...
```

ในการพล็อตเส้นโค้งนี้ จะเปลี่ยนมาเรียกผ่านฟังก์ชัน lines() ที่จะลากเส้นตรงเชื่อมระหว่างจุดตัดต่างๆ แต่เนื่องจากมีจุดตัดที่คำนวณไว้เป็นจำนวนมาก จึงได้ผลลัพธ์ออกมาเปรียบเสมือนเส้นโค้งดังรูปที่ 2.22

```
> plot(Petal.Width ~ Petal.Length)
> lines(loess.smooth(Petal.Length, Petal.Width))
```

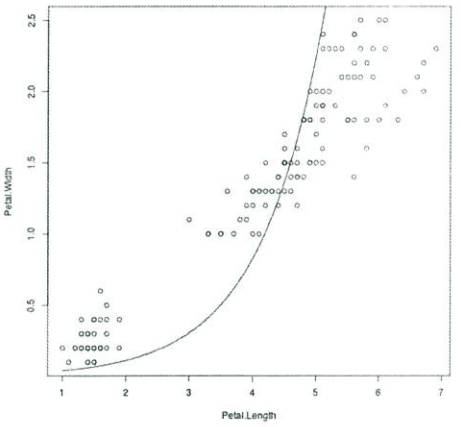


รูปที่ 2.22 โมเดลไม่เชิงเส้นของความกว้างต่อความยาวกลีบดอก

2.3.11.5 ลากโมเดลอิสระจากสมการที่กำหนดเอง

นอกจากการลากโมเดลเชิงเส้นและไม่เป็นเชิงเส้นแล้ว ยังสามารถนำสมการไปพล็อตทดสอบกับข้อมูลจริงได้ แสดงการลากโมเดลอิสระจากสมการที่กำหนดเองดังรูปที่ 2.23

```
> plot(Petal.Width ~ Petal.Length)
> curve(exp(x-4/3*pi), add=TRUE)
```

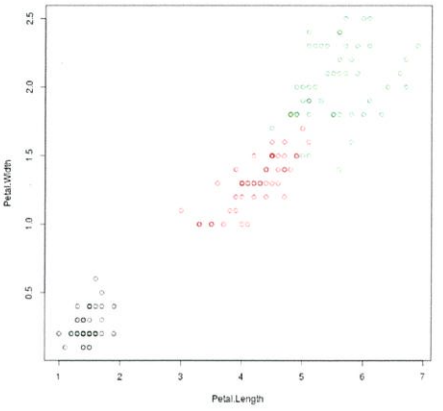


รูปที่ 2.23 โมเดลจากสมการ $e^{x-(4/3)\pi}$

2.3.11.6 แต้มสีแยกกลุ่มข้อมูล

จากข้อมูลตั้งต้นของดอกกุหลาบ นอกจากจะมีข้อมูลเชิงตัวเลขที่บ่งบอกลักษณะดอกไม้แล้ว ยังมีข้อมูลการแบ่งกลุ่มดอกไม้ตามสปีชีส์อีกด้วย โดยเพิ่มการแบ่งกลุ่มเข้าไปอีกแกนหนึ่ง เช่น แต้มสีจุดของแต่ละกลุ่มให้แตกต่างกัน ดังรูปที่ 2.24 ข้อมูลดอกกุหลาบโดยแยกสีตามสปีชีส์ ซึ่งสามารถสั่งได้ดังนี้

```
> plot(Petal.Width ~ Petal.Length, col=Species)
```

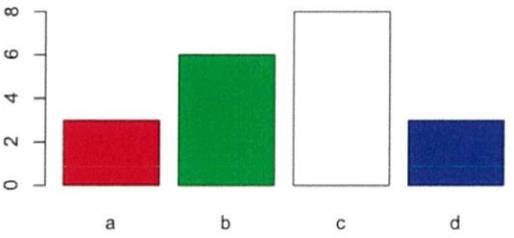


รูปที่ 2.24 ข้อมูลดอกกุหลาบโดยแยกสีตามสปีชีส์

2.3.11.7 การแสดงผลด้วยกราฟแท่ง

สามารถสั่งวาดกราฟแท่งโดยแจกแจงจำนวนข้อมูลที่มีในแต่ละกลุ่มได้ เช่น มีข้อมูลเกรด a,b,c,d ของนักเรียน จำนวน 20 คนเก็บไว้ในตัวแปร x จากนั้นใช้คำสั่ง >barplot(table(x)) หากต้องการแสดงสีที่ต่างกันสามารถเติมคำสั่ง col=c("red", "green", "white","blue") ตัวอย่างกราฟแท่งแสดงการแจกแจงเกรดของนักเรียนโดยใส่สีดังรูปที่ 2.25

>barplot(table(x), col=c("red", "green", "white", "blue"))

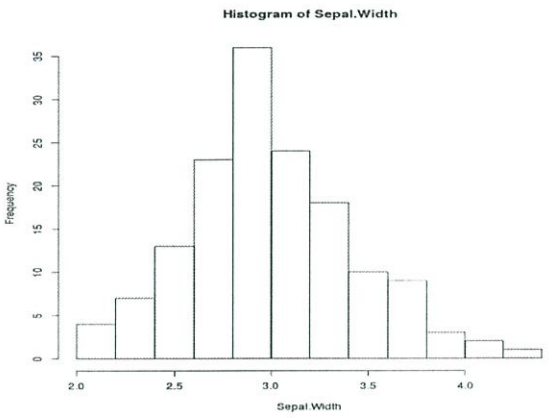


รูปที่ 2.25 กราฟแท่งแสดงการแจกแจงเกรดของนักเรียนโดยใส่สี

2.3.11.8 ฮิสโตแกรม (histogram)

ฮิสโตแกรมเป็นหนึ่งในวิธีพื้นฐานสำหรับแสดงการแจกแจงข้อมูล โดยจะแสดงแท่งสี่เหลี่ยมที่ความสูงต่างกัน แต่ละแท่งบอกว่ามีข้อมูลจำนวนเท่าใดที่เกิดขึ้นในช่วงนั้นๆ โดยใช้คำสั่ง hist() ตัวอย่างการพล็อตฮิสโตแกรมแสดงความกว้างกลีบเลี้ยงดอกกุหลาบ แสดงผลลัพธ์ดังรูปที่ 2.26

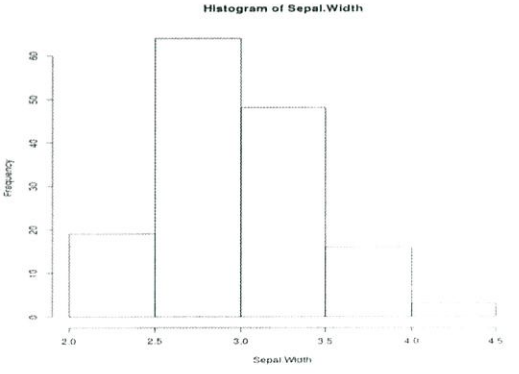
```
> hist(Sepal.Width)
```



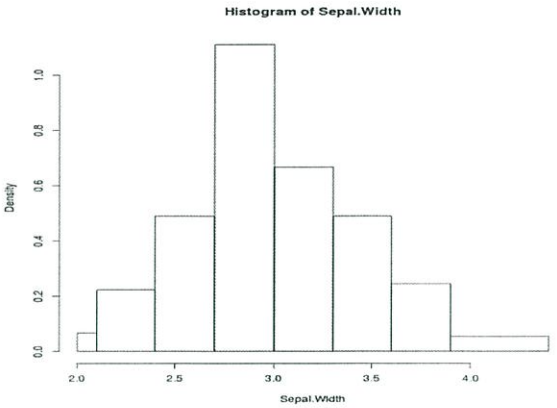
รูปที่ 2.26 ฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบ

นอกจากนี้ยังสามารถเปลี่ยนจำนวนรอยต่อระหว่างช่วงที่ต้องการได้ ตัวอย่างเช่นต้องการฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบโดยขีดเส้นแบ่งช่วง 4 ครั้ง สามารถทำได้ด้วยการระบุอาร์กิวเมนต์ breaks เพิ่มเข้าไป จะแสดงผลลัพธ์ที่ต้องการดังรูปที่ 2.27 หรืออาจระบุตำแหน่งทั้งหมดของรอยต่อระหว่างช่วงที่ต้องการแบ่งก็ได้ โดยมีอิสระที่ไม่จำเป็นต้องเลือกให้แต่ละช่วงมีขนาดเท่ากันได้ผลลัพธ์แสดงดังรูปที่ 2.28

```
> hist(Sepal.Width, breaks=4)
> hist(Sepal.Width, breaks=c(2,2.1,2.4,2.7,3.0,3.3,3.6,3.9,4.4))
```



รูปที่ 2.27 ฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบ
โดยขีดเส้นแบ่งช่วง 4 ครั้ง



รูปที่ 2.28 ฮิสโตแกรมความกว้างกลีบเลี้ยงดอกกุหลาบ
โดยแบ่งขนาดช่วงไม่เท่ากัน

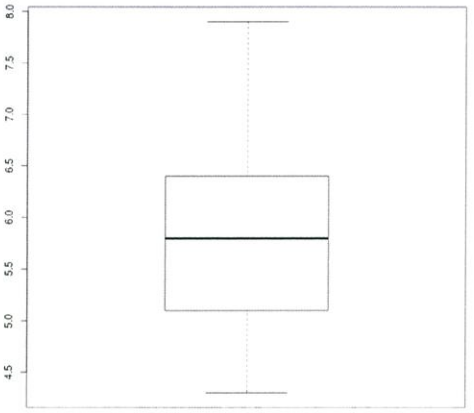
2.3.11.9 พล็อตกล่อง (box plot)

พล็อตกล่องนั้นไม่ได้แสดงจำนวนการเกิดของข้อมูลเป็นช่วงๆ เหมือนอย่างฮิสโตแกรม แต่จะแสดงข้อมูลด้วยค่าสรุปจากควอร์ไทล์แทนผลลัพธ์ที่ได้จากพล็อตกล่องความกว้างกลีบเลี้ยงดอกกุหลาบในรูปที่ 2.29 มีลักษณะเป็นกล่องสี่เหลี่ยมที่ลอยอยู่และมีขายื่นออกมาข้างบนและล่าง เส้นหนาที่ลากผ่านกลางกล่องคือมัธยฐาน ขอบล่างของกล่องคือควอร์ไทล์ที่ 1 และขอบบนของกล่องคือควอร์ไทล์ที่ 3 สำหรับขาแต่ละข้างที่ยื่นออกมาจากตัวกล่องนั้น สุดปลายขาล่างคือค่าต่ำสุด และสุดปลายขาบนคือค่าสูงสุดนั่นเอง แม้ว่าพล็อตกล่องจะแสดงการกระจายได้ไม่ละเอียดเท่าฮิสโตแกรม แต่ข้อได้เปรียบของพล็อตกล่องก็คือสามารถแสดงข้อมูลหลายๆ ชุดเปรียบเทียบกันได้ง่ายกว่า

```
> boxplot(Sepal.Length)
```

จุดเด่นอีกอย่างของพล็อตกล่อง คือสามารถแยกให้เห็นค่าผิดปรกติ (outlier) ได้ง่าย โดยแสดงเป็นจุดที่อยู่ไกลเลยขากล่องไปนั่นเอง หลักเกณฑ์สำหรับคัดแยกค่าผิดปรกติ คือ ค่านั้นมีค่าน้อยกว่าหรือมากกว่าค่าจากควอร์ไทล์ที่ 1หรือควอร์ไทล์ที่ 3 อยู่ 1.5 เท่าของพิสัยระหว่างควอร์ไทล์ตามลำดับ ซึ่ง

อาจตีความหมายได้ว่า ค่านั้นเกิดจากการจดบันทึกข้อมูลผิดเพี้ยน และควรลบทิ้งก่อนนำข้อมูลไปประมวลผลต่อไป

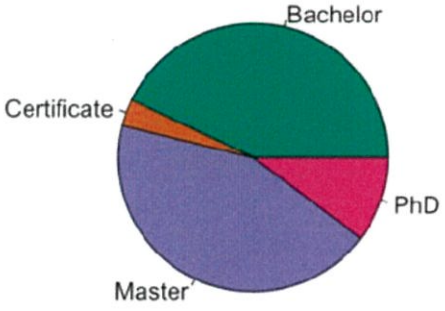


รูปที่ 2.29 พล็อตกล่องความกว้างกลีบเลี้ยงดอกกุหลาบ

2.3.11.10 กราฟวงกลม

การสร้างกราฟวงกลมจะแสดงข้อมูลเป็นสัดส่วนร้อยละของข้อมูลส่วนต่าง ๆ กราฟวงกลมสามารถทำได้โดยใช้คำสั่ง >pie(table(x)) ดังตัวอย่างการพล้อตข้อมูลแจกแจงระดับการศึกษา ดังรูปที่ 2.30

```
>group <- table(data$Edu)
>pie(group, col=brewer.pal(4,"Dark2"), radius=3)
```



รูปที่ 2.30 การแจกแจงระดับการศึกษา

2.3.12 ประกาศฟังก์ชัน

การประกาศฟังก์ชันในภาษาอาร์นั้นทำได้โดยใช้คำสำคัญว่า function ตามด้วยพารามิเตอร์ แล้วจบด้วยเนื้อหาในฟังก์ชันนั้นๆ สิ่งสำคัญที่ต่างออกไปจากภาษาอื่น คือการตั้งชื่อให้ฟังก์ชันจะต้องทำด้วยวิธีเดียวกันกับการประกาศตัวแปรเท่านั้นดังตัวอย่าง

```
>random.int <- function(n) { sample.int(n, 1) }
```

โค้ดข้างต้นสร้างฟังก์ชันชื่อว่า random.int สำหรับสุ่มเลขจำนวนเต็มระหว่าง 1 ถึง n ขึ้นมาหนึ่งตัว ตอนจบของฟังก์ชันไม่จำเป็นต้องสั่ง return เพื่อบอกว่าต้องการคืนค่าใดจากฟังก์ชัน แต่คำสั่งสุดท้ายในฟังก์ชันจะถูกส่งคืนค่าโดยอัตโนมัติ

2.3.13 วนซ้ำจนกว่าจะผิดเงื่อนไข (while loop)

การวนซ้ำจนกว่าจะผิดเงื่อนไข ประกาศโดยคำสำคัญ while และมีโครงสร้างเหมือนกับภาษาทั่วไปดังนี้

```
> while (test_expression){
   statement }
```

2.3.14 วนซ้ำสำหรับของแต่ละชิ้น (for loop)

การวนซ้ำสำหรับของแต่ละชิ้น ประกาศโดยใช้คำสำคัญ for ตามด้วยตัวแปรสำหรับรับสิ่งของในแต่ละรอบ แล้วตามด้วยเวกเตอร์ของสิ่งของทั้งหมดที่ต้องการวนซ้ำ มีโครงสร้างคือ

```
> for (varname in seq) {
statement(s) }
```

เมื่อ varname คือชื่อตัวแปร และ seq คือ vector ส่วน statements จะทำงานสำหรับค่าแต่ละค่าของ seq ดังตัวอย่างง่ายๆในการใช้โปรแกรม for loop คือ โปรแกรมนับถอยหลัง ดังนี้

```
> for ( i in 10:1 ) print (i)
```

2.3.15 ทำหรือไม่ตามแต่เงื่อนไขจะกำหนด (if-else)

การตรวจสอบถ้า-แล้ว สามารถใช้คำสำคัญได้เช่นเดียวกันกับภาษาทั่วไป คือ if else if และ else ดังตัวอย่าง

```
> a = c(5,7,2,9)
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd"  "odd"  "even" "odd"
```

จากโปรแกรมด้านบนจะเห็นว่า ถ้า a หาร 2 ลงตัวจะแสดงค่า even แต่ถ้าหารไม่ลงตัว จะแสดงค่า odd

2.3.16 เรียกใช้โปรแกรมไฟล์อาร์

การเรียกใช้โปรแกรมไฟล์อาร์สามารถทำได้โดยใช้คำสั่ง > source() แล้วใส่ที่อยู่ไฟล์อาร์ที่ต้องการดังนี้

```
> source("C:/Users/company /coupon.R")
```

2.3.17 ดาวน์โหลดแพ็กเกจเสริมความสามารถ

ในภาษาอาร์สามารถติดตั้งแพ็กเกจเพิ่มเติมได้โดยที่แพ็กเกจเป็นชุดคำสั่งหรือฟังก์ชันต่าง ๆ ที่มีผู้เขียนเพิ่มเติมและต้องการแบ่งปันให้ผู้อื่นได้ใช้ด้วย สามารถเข้าไปดูที่สถานที่จัดเก็บแพ็กเกจหลักอย่างเป็นทางการของภาษาอาร์มีชื่อว่า Comprehensive R Archive Network (CRAN) ซึ่งปัจจุบันมีแพ็กเกจเสริมความสามารถกว่าหนึ่งหมื่นแพ็กเกจให้เลือกใช้งาน วิธีการดาวน์โหลดแพ็กเกจเหล่านี้ทำได้โดยเรียกฟังก์ชันต่อไปนี้

```
> install.pacakges("packageName")
```

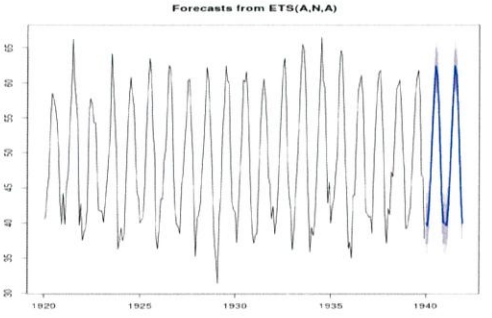
โดยเปลี่ยน "packageName" เป็นชื่อแพ็กเกจที่ต้องการ แล้วรอโปรแกรมดาวน์โหลดและติดตั้งแพ็กเกจเหล่านั้น หลังจากติดตั้งเสร็จสิ้น เมื่อต้องการนำเข้าแพ็คเกจเหล่านั้นมาใช้งาน ก็ทำได้โดยใช้คำสั่ง

```
> library(packageName)
```

2.3.18 ทำนายอนาคต

สำหรับข้อมูลที่มีแกนหนึ่งเป็นแกนเวลา เช่น ข้อมูลดินฟ้าอากาศ จำนวนประชากร หรือกระทั่งราคาหุ้น หากข้อมูลเหล่านั้นมีรูปแบบที่เกิดซ้ำมากพอ ก็อาจทำนายอนาคตของข้อมูลนั้นได้ดังตัวอย่างการพยากรณ์อุณหภูมิของเมืองนอตทิงแฮม รูปที่ 2.31

```
> library(forecast)
> plot(forecast(nottem))
```



รูปที่ 2.31 พยากรณ์อุณหภูมิของเมืองนอตทิงแฮม (เส้นสีน้ำเงิน)

2.3.19 แผนที่โลก

นอกจากการพล็อตข้อมูลเชิงสถิติแล้วอาร์ยังสามารถพล็อตแผนที่โลกได้อีกด้วย ดังตัวอย่างการพล็อต แผนที่โลกพร้อมตำแหน่งเกิดแผ่นดินไหวดังรูปที่ 2.32

```
> library(maps)
> library(mapdata)
> map("world2Hires", xlim=c(110,190), ylim=c(-50,0))
> points(quakes[2:1], pch=".", col="red")
```

รูปที่ 2.32 แผนที่โลกพร้อมตำแหน่งเกิดแผ่นดินไหว (จุดสีแดง)

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

กระบวนการออกแบบโปรแกรมวิเคราะห์และจัดเรียงข้อมูลโดยใช้ภาษาอาร์นั้น จะต้องมีการรวบรวมข้อมูลของผู้ให้บริการของแต่ละเครือข่ายไม่ว่าจะเป็น DTAC AIS และ TRUE MOVE แล้วทำความเข้าใจถึงข้อมูล ความหมายของแต่ละข้อมูลรวมถึงแหล่งที่มาของข้อมูล จากนั้นต้องมีการประชุมกับผู้บริหารและพนักงานในองค์กรถึงข้อมูลสำคัญต่าง ๆ ที่จะถูกนำมาช่วยในการตัดสินใจแก้ปัญหาเกี่ยวกับการทำงานที่เกิดขึ้นเช่น ข้อมูลความคืบหน้าของโครงการ ข้อมูลการส่งงาน และข้อมูลแสดงประสิทธิภาพของพนักงานในองค์กรเป็นต้น นอกจากนี้ต้องรู้ถึงกระบวนในการจัดเรียงข้อมูล วิธีการที่ใช้ในการนำข้อมูลมาวิเคราะห์และรูปแบบของการแสดงผล แล้วจึงนำภาษาอาร์มาช่วยในเขียนโปรแกรมช่วยวิเคราะห์ข้อมูลได้ อีกทั้งเพื่อความสะดวกในการใช้งานยังสามารถใช้ Shiny แพคเกจแอพพลิเคชั่นแบบโต้ตอบกับผู้ใช้งานสำหรับภาษาอาร์มาช่วยในการออกแบบหน้ารูปแบบการใช้งานโปรแกรมวิเคราะห์ข้อมูลได้อีกด้วย

3.1 การออกแบบ

3.1.1 การประชุมกับบริษัทถึงข้อมูลที่ต้องการประมวลผล

การประชุมกับบริษัทถึงข้อมูลที่ต้องการประมวลผลนั้นเพื่อวิเคราะห์ถึงปัญหาภายในบริษัท ปัญหาการทำงานของผู้รับเหมา ตลอดจนปัญหาการทำงานของพนักงานในองค์กร แล้วสรุปหาข้อมูลที่ต้องการประมวลพร้อมแสดงผล เพื่อช่วยในการประกอบการตัดสินใจของผู้บริหาร การออกแบบในขั้นตอนนี้ จำเป็นต้องใช้เวลาในการประชุม พร้อมทั้งปรึกษากับพนักงานภายในองค์กรถึงข้อมูลที่สำคัญต่าง ๆ ตลอดจน ต้องวิเคราะห์ว่าจะต้องการแสดงผลในส่วนของข้อมูลใดบ้างดังนี้

1) Report Submitted Time (เวลาที่ผู้รับเหมาส่งงาน)

ข้อมูลแสดงเวลาที่ผู้รับเหมาส่งงานของแต่ละโครงการ โดยจะแสดงว่ามีการส่งงานเข้ามาภายภายในระบบเพื่อรอรับการตรวจงานเวลาใด และเป็นจำนวนกี่งานการติดตั้งสัญญาณ ข้อมูลนี้จะช่วยแก้ถึงปัญหาการตรวจรับงานไม่ทันของพนักงานในองค์กร

2) Clock in – Clock out (เวลาที่ผู้รับเหมาเข้าและออกไซต์งาน)

ข้อมูลแสดงเวลาที่ผู้รับเหมาเข้าและออกไซต์งานการติดตั้งเสาสัญญาณ จะแสดงถึงเวลาที่ผู้รับเหมาเข้าไปทำงานในสถานที่ทำงาน ตลอดจนเวลาที่ผู้รับเหมาออกจากสถานที่ทำงาน จะแสดงให้เห็นว่าผู้รับเหมาส่วนมากทำงานในเวลาใด ซึ่งข้อมูลนี้เองจะช่วยสรุปปัญหาการส่งงานไม่ทัน หรือการติดตั้งที่ล่าช้าเนื่องจากเวลาในการเข้าไซต์งานการติดตั้งที่น้อยในแต่ละวัน

3) Report Status by Employees (สถานะการตรวจงานผู้รับเหมาภายในแผนก)

ข้อมูลแสดงสถานะการตรวจงานผู้รับเหมาของพนักงานภายในแผนก เมื่อผู้รับเหมาส่งงานเข้ามา พนักงานในแผนกจะทำหน้าที่ในการตรวจงานโดยจะสรุปสถานะของแต่ละไซต์งานการติดตั้งเอาไว้ ว่าในงานติดตั้งเสาสัญญาณนั้น ๆผ่านหรือไม่ผ่านการตรวจรับของบริษัท ข้อมูลนี้จะแสดงถึงประสิทธิภาพของพนักงานภายในองกรณ์ตลอดจนแสดงถึงปัญหางานที่ไม่ผ่านการตรวจรับ เพื่อนำไปวิเคราะห์ถึงสาเหตุต่อไป

4) Report Status by ISDP System (สถานะงานของผู้รับเหมาในระบบ)

ข้อมูลแสดงสถานะงานของผู้รับเหมาในระบบ เพื่อสรุปงานถึงสถานะงานต่าง ๆ เช่น งานที่รอติดตั้ง งานที่รอตรวจรับ งานที่ผ่านการตรวจรับ และงานที่ไม่ผ่านการตรวจรับ เป็นต้น

3.1.2 การวิเคราะห์ข้อมูล

การวิเคราะห์ถึงข้อมูลที่ต้องการแสดงผลว่า ต้องใช้ข้อมูลส่วนใด และจะจัดการกับข้อมูลที่มีอยู่อย่างไร เพื่อให้ได้มาซึ่งผลสรุปของข้อมูล ในส่วนขั้นตอนนี้จำเป็นต้องรู้ถึงความหมายของแต่ละข้อมูลภายในองค์กร รู้ถึงแหล่งที่มาของข้อมูล และทำความเข้าใจกับข้อมูล

3.1.3 การออกแบบการเขียนโปรแกรมวิเคราะห์ข้อมูลโดยใช้ภาษาอาร์

การออกแบบในส่วนของการเขียนโปรแกรมนั้นจะใช้ภาษาอาร์ในการเขียนโปรแกรมจัดเรียงข้อมูล ตลอดจนประมวลผล และแสดงผลของข้อมูลทั้งหมด ในขั้นตอนนี้ต้องเรียนรู้วิธีการเขียนโปรแกรม รูปแบบตัวแปรในภาษาอาร์ คำสั่งและแพคเกจการทำงานต่างๆ ของโปรแกรมอาร์ แล้วจึงนำมาใช้ในการออกแบบโปรแกรมวิเคราะห์และจัดเรียงข้อมูลและแสดงผลออกมาเป็นรูปกราฟ

3.1.4 การสร้างแอพสำหรับแสดงผลข้อมูล

การสร้างแอพโดยใช้ Shiny ซึ่งเป็นแพกเกจหนึ่งซึ่งเปิดให้ติดตั้งได้ฟรี เป็นแพจเกจในการสร้างแอพพลิเคชั่นสำหรับภาษาอาร์ โดย Shiny จะประกอบไปด้วยสององค์ประกอบคือ ในส่วนของ UI (user-interface) กับส่วนของ Server (server script) โดยหน้าตาของแอพพลิเคชั่น Shiny ที่ทำการออกแบบสำหรับแสดงผลข้อมูลทั้งหมดแสดงดังรูปที่ 3.1 โดยจะมีส่วนที่สามารถโต้ตอบกับผู้ใช้งาน เมื่อผู้ใช้งานทำการเปลี่ยนค่าข้อมูลที่ต้องการ ข้อมูลที่แสดงผลก็จะเปลี่ยนไปด้วยเช่นกัน

รูปที่ 3.1 แอพพลิเคชั่น Shiny ที่ทำการออกแบบ

3.2 เครื่องมือที่ใช้ในการทดลอง

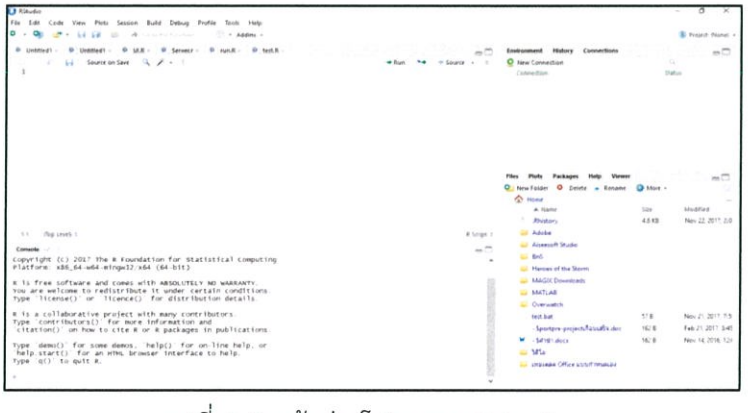
1) คอมพิวเตอร์ที่มีการติดตั้งภาษาอาร์

คอมพิวเตอร์ที่นำมาใช้ต้องมีการติดตั้งโปรแกรมอาร์โดยในขั้นตอนการติดตั้งโปรแกรมเริ่มต้นจากการดาวน์โหลดโปรแกรมจากเว็บไซต์ https://cran.r-project.org/ และเลือกระบบปฏิบัติการที่ต้องการติดตั้ง โปรแกรมภาษาอาร์จะมีลักษณะไอคอนดังรูปที่ 3.2

รูปที่ 3.2 ไอคอนภาษาอาร์

2) Rstudio

การติดตั้ง Rstudio เพิ่มเติมจะช่วยอำนวยความสะดวกในเวลาที่ต้องใช้ชุดคำสั่งเดิมหลายๆครั้ง โดยในการทดลองจะทำการเขียนโปรแกรมผ่าน Rstudio ซึ่งมีหน้าต่างโปรแกรมดังรูปที่ 3.3 จะเห็นว่า มีทั้งในส่วน R console ที่รันโปรแกรมทีละคำสั่ง และ R script ที่สามารถรันคำสั่งๆหลายๆคำสั่งพร้อมกันได้



รูปที่ 3.3 หน้าต่างโปรแกรม Rstudio

3) ข้อมูลที่ต้องการประมวลผล (Big data)

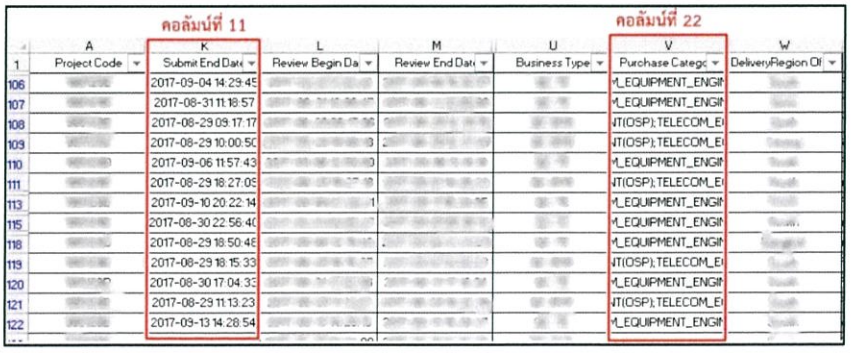
ข้อมูลที่ต้องการประมวลผล (Big data) อาจจะอยู่ในรูปแบบของของไฟล์ Excel หรือไฟล์นามสกุล .csv (Comma-Separated Value) ข้อมูลนี้จะถูกนำมาจัดเรียงโดยใช้การเขียนโปรแกรมในภาษาอาร์ แล้วถูกนำมาแสดงผลเป็นรูปกราฟ

3.3 การจัดเก็บผลการทดลอง

3.3.1 การนำไฟล์เข้าโดยใช้ XLConnect

ในการนำไฟล์รูปแบบของเอกเซลเข้ามาในโปรแกรมภาษาอาร์ จำเป็นต้องมีการติดตั้งแพกเกจ XLConnect เพื่อทำการดึงไฟล์ของข้อมูลก่อน โดยในการใช้งานต้องมีการเรียกใช้งานไลบรารี่ แล้วจึงนำไฟล์เข้ามาได้ นอกจากนี้ยังสามารถเลือกชีทของงานในหน้าเอกเซล ตลอดจนเลือกตำแหน่งคอลัมน์ที่ต้องการดึงมาใช้งานได้อีกด้วย ดังตัวอย่างการเขียนโปรแกรมอาร์เรียกใช้งานไลบรารี่ XLConnect ดังนี้ เมื่อต้องการดึงไฟล์นามสกุลเอกเซลที่มีชื่อว่า “Project01” และมีหน้าชีทภายในไฟล์เอกเซลว่า “Task” โดยในทีนี้จะดึงข้อมูลเฉพาะคอลัมน์ที่ 11 ถึง คอลัมน์ที่ 22 เท่านั้น โดยรูปแบบไฟล์เอกเซลที่ทำการดึงข้อมูลแสดงดังรูป 3.4 เป็นข้อมูล Big data ขนาดใหญ่ที่ประกอบด้วยข้อมูลจำนวนมาก

```
> library(XLConnect)
> Project <- "C:/Users/Desktop/Rstudio/Project01.xlsx"
> T <- loadWorkbook(Project)
> TMBB <-readWorksheet(T, sheet = "Task",startCol = 11,endCol = 22 )
```



รูปที่ 3.4 ข้อมูล Big Data ที่ทำการดึงเข้าโปรแกรมอาร์

3.3.2 การใช้คำสั่ง subset และ substr

คำสั่ง subset และ substr ใช้สำหรับการกรองข้อมูลที่สนใจ โดย Subset จะทำการกรองคอลัมน์ข้อมูลที่ตรงกับเงื่อนไข และ Substr จะทำการดึงตัวอักษรที่สนใจในคอลัมน์ออกมา เช่นเมื่อมีข้อมูลแสดงดังรูปที่ 3.5 โดยในข้อมูลนี้สนใจเพียงแค่คอลัมน์ “Submit End Date” เท่านั้น เมื่อต้องการข้อมูลเฉพาะตัวอักษรที่ 11-13 สามารถทำได้โดยใช้คำสั่ง Substr และถ้าหากต้องการกรองเฉพาะแถวที่มีข้อมูลเป็น “14” ก็สามารถทำได้โดยใช้คำสั่ง subset ดังการเขียนโปรแกรมด้วยภาษาอาร์ดังนี้

```
> TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
> data <- subset(TMBB, Submit == "14" )
```

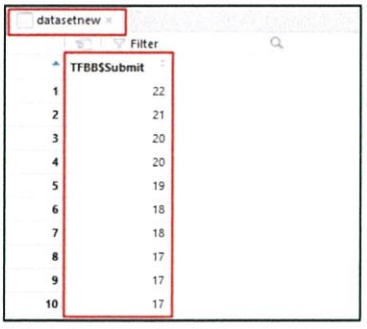
Submit.End.Date	Review.Begin.Date	Review.End.Date	Actual.Reviewer
2017-04-11 12:13:57	[redacted]	[redacted]	[redacted]
NA	[redacted]	[redacted]	[redacted]
2017-04-11 15:31:28	[redacted]	[redacted]	[redacted]
NA	[redacted]	[redacted]	[redacted]
2017-06-23 14:49:10	[redacted]	[redacted]	[redacted]
2017-04-19 15:29:58	[redacted]	[redacted]	[redacted]
2017-04-18 16:51:54	NA	NA	NA
NA	[redacted]	[redacted]	[redacted]
2017-05-08 14:20:55	[redacted]	[redacted]	[redacted]

รูปที่ 3.5 ข้อมูลประกอบการใช้คำสั่ง Subset และ Substr

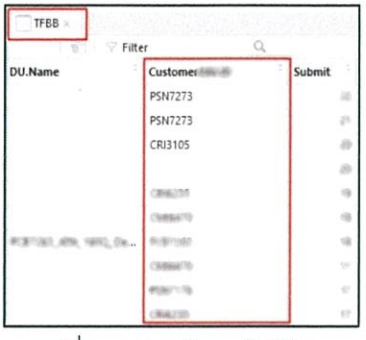
3.3.3 คำสั่ง cbind

คำสั่ง cbind เป็นคำสั่งที่ใช้ในการนำคอลัมน์ของแต่ละไฟล์ข้อมูลมารวมกัน ดังคำสั่งที่ใช้ดังนี้ เมื่อมีข้อมูลไฟล์ A และข้อมูลไฟล์ B ดังรูปที่ 3.6 (ก) และ (ข) ตามลำดับ เมื่อต้องการรวมคอลัมน์ของทั้งสองไฟล์เข้าด้วยกันแล้วนำไปเก็บไว้ในไฟล์ใหม่ สามารถใช้คำสั่ง cbind ในการรวมไฟล์ดังนี้

```
> dataset <- cbind(datasetnew, ID = TFBB$Customer)
```



รูปที่ 3.6 (ก) ข้อมูลไฟล์ A

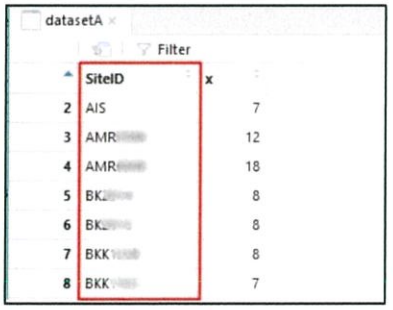


รูปที่ 3.6 (ข) ข้อมูลไฟล์ B

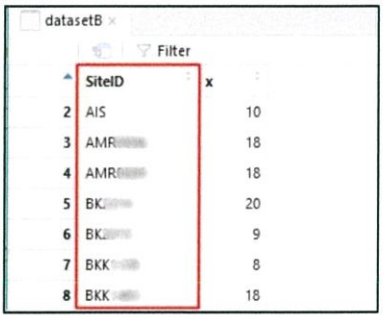
3.3.4 คำสั่ง merge

คำสั่ง merge มีลักษณะการทำงานที่คล้ายกับ VLOOKUP ในโปรแกรมเอกเซลคือการรวมข้อมูลสองชุดโดยใช้คอลัมน์หรือชื่อแถวร่วมกัน โดยในที่นี้เมื่อมีข้อมูล 2 ชุดคือ ข้อมูล datasetA และข้อมูล datasetB ดังรูปที่ 3.7 (ก) และ (ข) ตามลำดับ แล้วต้องการรวมไฟล์โดยใช้คอลัมน์ที่มีชื่อว่า “SiteID” ร่วมกัน ดังคำสั่งดังนี้

```
> dataset = merge(x = datasetA, y = datasetB, by = "SiteID", all = TRUE)
```



รูปที่ 3.7 (ก) ข้อมูลไฟล์ datasetA



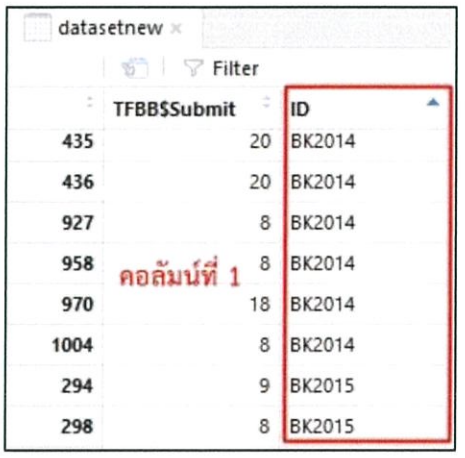
รูปที่ 3.7 (ข) ข้อมูลไฟล์ datasetB

3.3.5 คำสั่ง Aggregate

คำสั่ง Aggregateเป็นฟังก์ชั่นชุดหนึ่งใช้สำหรับคำนวณค่าผลรวมของข้อมูลทั้งตารางหรือเพียงบางกลุ่มของแถวข้อมูล โดยในที่นี้เมื่อมีข้อมูลชื่อว่า “datasetnew” แสดงดังรูปที่ 3.8 ต้องการคำนวณค่าผลรวมของข้อมูลที่มี “ID”ซ้ำกัน โดยคำนวณโดยใช้คำสั่ง min และ max เพื่อการหาค่าต่ำสุดและสูงสุดของข้อมูลคอลัมน์ที่ 1 สามารถเขียนโปรแกรมอาร์ดังนี้

> `datasetA` <- aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), min)

> `datasetB` <- aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), max)



รูปที่ 3.8 ข้อมูลไฟล์ “datasetnew” ประกอบการใช้คำสั่ง Aggregate

3.3.6 การพล็อตกราฟแท่ง Barplot

คำสั่ง barplot เป็นคำสั่งที่ใช้ในการพล็อตกราฟแบบง่ายๆดังตัวอย่างการพล็อตกราฟเมื่อมีข้อมูลแสดงการแจกแจงดังรูปที่ 3.9 สามารถนำมาพล็อตกราฟแท่งแสดงการแจกแจงข้อมูลโดยใช้คำสั่ง barplot () นอกจากนี้ยังสามารถกำหนดชื่อแกน X และแกน Y กำหนดชื่อกราฟ กำหนดสีให้กับแท่งกราฟ กำหนดสีขอบเส้นกราฟ และกำหนดขนาดของแกน Y ได้อีกด้วย ดังตัวอย่างโปรแกรมอาร์ต่อไปนี้

> mybar = barplot(data , ylab = "Total Report",xlab ="Time",names.arg = m4$Time,main = "Project01 Submitted Time",col = ifelse(y== max(y), "cadetblue4","lightseagreen"), border="white", ylim = c(0,max(y)+100))

รูปที่ 3.9 ข้อมูลแสดงการแจกแจงจำนวนงานที่เวลาต่างๆ

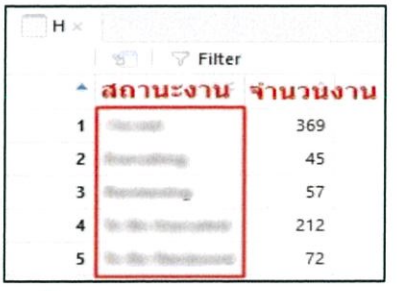
ทั้งนี้ยังสามารถเพิ่มข้อความแสดงจำนวนบนแท่งกราฟได้ โดยใช้คำสั่งโปรแกรมอาร์ดังนี้

> text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey")

3.3.7 การพล็อตกราฟวงกลม Pie

คำสั่ง pie มีการทำงานที่เหมือนกับคำสั่ง barplot ต่างแค่รูปกราฟจะแสดงอยู่ในรูปของวงกลมแสดงเป็นสัดส่วนร้อยละของข้อมูลทั้งหมด เมื่อมีข้อมูลแสดงการแจกแจงสถานะของงานและจำนวนงานดังรูปที่ 3.10 เมื่อต้องการพล็อตกราฟข้อมูลร้อยละของงานทั้งหมด สามารถเรียกคำสั่ง pie() ดังตัวอย่างการพล็อตกราฟวงกลมดังนี้

>pie(data,labels = PER , main = "Report Status",col=c("#4682B4","#B0C4DE","#ADD8E6","#87CEFA","#4169E1","Pink") ,border = "WHITE")



รูปที่ 3.10 ข้อมูลแสดงการแจกแจงสถานะของงาน

นอกจากนี้ยังสามารถกำหนดชื่อของกราฟ สีของกราฟ เส้นขอบกราฟ และพื้นหลังได้เหมือนกับการพล็อตกราฟแท่ง

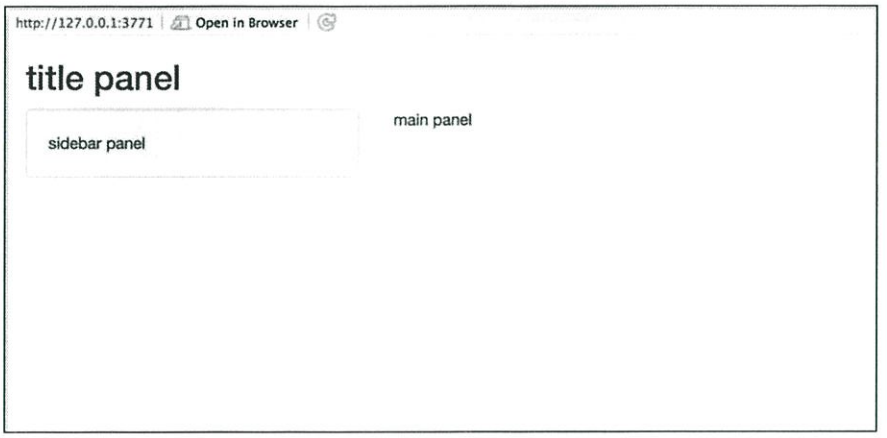
3.3.8 การสร้างแอพพลิเคชั่นเว็บจาก Shiny

Shiny เป็นแพคเกจของภาษาอาร์สำหรับการสร้างแอพพลิเคชั่นเว็บแบบโต้ตอบ (interactive web) โดยต้องทำการติดตั้งแพคเกจ Shiny ก่อนจึงจะสามารถใช้งานได้ ซึ่งการเขียนโปรแกรมจะประกอบด้วยสององค์ประกอบที่สำคัญคือ user-interface และ server script ส่วนของ user-interface จะควบคุมรูปแบบและลักษณะที่ปรากฏบนแอพพลิเคชั่น ส่วน server script จะเป็นตัวกำหนดและประมวลโปรแกรมที่ต้องการให้แสดง ลักษณะการเขียนโปรแกรมอย่างง่ายแสดงดังนี้

```
> library(shiny)
> ui <- ...                                    # ส่วนของ UI
> server <- ...                                # ส่วนของ Server
> shinyApp(ui = ui, server = server)
```

1) ส่วนของ user-interface

Shiny ใช้ฟังก์ชัน fluidPage เพื่อสร้างหน้าจอที่ปรับให้เข้ากับขนาดของหน้าต่างบราวเซอร์ของผู้ใช้ในทีนี้จะสร้างหน้าต่างของ User interface ให้โต้ตอบกับผู้ใช้ได้ โดยจะทำการเก็บค่าจากผู้ใช้ เมื่อผู้ใช้เปลี่ยนรูปแบบการส่งค่า ข้อมูลที่แสดงผลก็จะเปลี่ยนไปเช่นกัน รูปแบบของการหน้าต่างโปรแกรมแสดงดังรูปที่ 3.11 โดยจะสร้างส่วนโปรแกรมอาร์นี้ที่ผู้ใช้สามารถโต้ตอบได้ใน Sidebar panel 3 รูปแบบ นอกจากนี้ยังสามารถปรับรูปแบบตัวอักษร สี ขนาดตัวอักษรที่แสดงบนหน้าต่างแอพพลิเคชั่นเว็บได้อีกด้วย



รูปที่ 3.11 รูปแบบของแอพพลิเคชั่นเว็บจากแพคเกจ Shiny

รูปแบบที่ 1 กล่องที่มีตัวเลือกให้เลือก ดังรูปที่ 3.12 (ก) มีรูปแบบการเขียนโปรแกรมอาร์ดังนี้

```
> library(shiny)                                    # เรียกใช้งานไลบรารี่ Shiny
> shinyui <- fluidPage(                             # ส่วนของ UI
      titlePanel("Summary Report"),
      sidebarLayout(
      sidebarPanel(
      selectInput("var",  label = h3("Select Project"),
       choices =list("Project01","Project02","Project03",
                     " Project04","Project05"),
       selected = "Project01"))
```

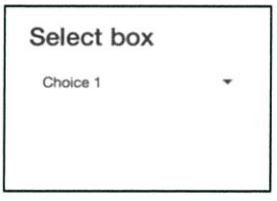
รูปแบบที่ 2 ชุดปุ่มวิทยุ ดังรูปที่ 3.12 (ข) มีรูปแบบการเขียนโปรแกรมอาร์ดังนี้

```
> library(shiny)                                    # เรียกใช้งานไลบรารี่ Shiny
> shinyui <- fluidPage(                             # ส่วนของ UI
      titlePanel("Summary Report"),
      sidebarLayout(
```

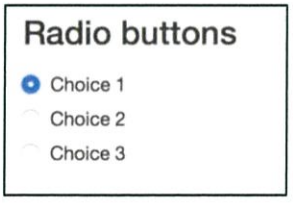
```
        sidebarPanel(
        radioButtons("radio", h3("Business Type"),
        choices = list("Installation" = 1, "Safety" = 2),
        selected = 1))
```

รูปแบบที่ 3 การอัพโหลดไฟล์นามสกุล .CSV ดังรูปที่ 3.12 (ค) มีรูปแบบการเขียนโปรแกรมอาร์ดังนี้

```
> library(shiny)                                  # เรียกใช้งานไลบรารี่ Shiny
> shinyui <- fluidPage(                           # ส่วนของ UI
        titlePanel("Summary Report"),
        sidebarLayout(
        sidebarPanel(
         helpText("Status Report"),
         fileInput("file1", label = h3("File input .csv file"))
```



รูปที่ 3.12 (ก)
กล่องที่มีตัวเลือกให้เลือก



รูปที่ 3.12 (ข)
ชุดปุ่มวิทยุ

รูปที่ 3.12 (ค)
อัพโหลดไฟล์

นอกจากนี้ยังสามารถเพิ่มรูปภาพบนหน้าต่างแอพพลิเคชั่นเว็บได้อีกด้วย โดยใช้คำสั่ง img() แล้วตามด้วยชื่อไฟล์ ซึ่งสามารถกำหนดขนาดของรูปภาพได้ตามความหมาะสม แต่การใส่ไฟล์รูปภาพนั้นจะมีข้อกำหนดคือ ไฟล์รูปภาพที่นำมาใช้แสดง

ในเว็บนั้นจะต้องถูกจัดเก็บไว้ในโฟล์เดอร์ชื่อว่า “WWW” ของโฟล์เดอร์ที่จัดเก็บไฟล์แอพพลิเคชั่นโปรแกรมอาร์ แสดงการจัดเก็บไฟล์รูปดังรูปที่ 3.13

```
> img(src = "my_image.png", height = 72, width = 72)
```

รูปที่ 3.13 การจัดเก็บไฟล์รูปที่ใช้แสดงบนแอพพลิเคชั่นเว็บ

2) ส่วนของ server script

ส่วนของ server script จะแสดงค่าผลลัพธ์ โดยจะมีค่าสอดคล้องกับส่วนของ User Interface ข้อมูลผลลัพธ์ที่นำมาแสดงเป็นไปตามโปรแกรมอาร์ที่ทำการออกแบบไว้ โดยในที่นี้ทำการออกแบบโปรแกรมวิเคราะห์และจัดเรียงข้อมูล ดังนั้นจะนำค่ากราฟที่ผ่านการจัดเรียงข้อมูลแล้ว มาแสดงผลในส่วนนี้ ซึ่งโปรแกรมอาร์ที่ทำการออกแบบจะประกอบไปด้วยการแสดงค่าผลลัพธ์ 3 รูปแบบ

รูปแบบที่ 1 ข้อมูลแสดงเวลา ข้อมูลนี้จะแสดงผลลัพธ์เป็นเวลาและวันที่ ออกทางหน้าต่างแอพพลิเคชั่นเว็บ

```
> Shinyserver <- function(input, output, session ) {
        output$currentTime <- renderText({  #output เป็นตัวอักษรแสดงเวลา
        invalidateLater(1000, session)
        paste("DATE and TIME :", Sys.time())
 })
```

รูปแบบที่ 2 ข้อมูลแสดงค่าที่จัดเก็บเข้ามา เมื่อผู้ใช้งานทำการเลือกข้อมูลในส่วนของ user-interface แล้วนั้น ข้อมูลนี้จะถูกจัดเก็บแล้วนำมาแสดงค่าที่หน้าต่างแอพพลิเคชั่นเว็บเช่นเมื่อผู้ใช้งานทำการเลือกโปรเจค01 โปรแกรมจะรับค่าว่าผู้ใช้งานทำการเลือกโปรเจค01 แล้วนำมาแสดงเป็นตัวอักษรบนหน้าจอว่าผู้ใช้งานทำการเลือกข้อมูลแบบใด

```
> Shinyserver <- function(input, output, session ) {
      output$selected_var <- renderText({         # output เป็นตัวอักษร
          paste("You have selected", input$var)   #นำค่า input กลับมาแสดงผล
})
```

ข้อมูลที่ 3 ข้อมูลแสดงผลลัพธ์กราฟ เมื่อผู้ใช้งานทำการเลือกข้อมูลที่ต้องการดูแล้วนั้น โปรแกรมจะจัดเรียงข้อมูลและนำข้อมูลมมาแสดงผลลัพธ์ในรูปแบบของกราฟ ทั้งกราฟแท่ง และกราฟวงกลม

```
> Shinyserver <- function(input, output, session ) {
          output$myPlot <- renderPlot({            #output แสดงการพล็อตกราฟ
          if(input$var == "Project01"   && input$radio == 1  ){
                  ....                 # โปรแกรมในส่วนของวิเคราะห์และแสดงผลข้อมูล
}
```

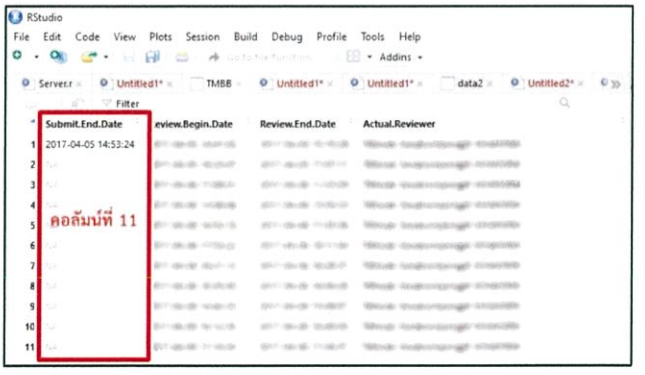
บทที่ 4

ผลการทดลอง

ผลการทดสอบการสร้างโปรแกรมจัดเรียงข้อมูลและแสดงผลผ่านแอพพลิเคชั่นเว็บโดยใช้ภาษาอาร์นั้น ต้องดำเนินการเก็บผลทดลอง ศึกษาพร้อมทั้งความเข้าใจกับคำสั่งต่าง ๆในภาษาอาร์ รวมถึงเครื่องมือตลอดและแพคเกจต่างๆ ที่จะนำมาช่วยในจัดเรียงแสดงผลข้อมูลและพร้อมต่อการเรียกใช้งานของผู้ใช้

4.1 ผลการทดสอบการนำไฟล์เข้าโดยใช้ XLConnect

ผลการทดสอบการนำไฟล์เข้าโดยใช้ XLConnect จะเป็นการทดสอบการนำไฟล์รูปแบบของเอกเซลเข้ามาในโปรแกรมภาษาอาร์ ซึ่งหลังจากมีการติดตั้งแพกเกจ XLConnect แล้วนั้น ทำการเขียนโปรแกรมอาร์เรียกใช้งานไลบรารี่ XLConnect แล้วทดสอบการรันโปรแกรม ผลการทดสอบการนำไฟล์เข้าโดยใช้ XLConnect ข้อมูลที่ทำการดึงเข้าโปรแกรม จะไปแสดงผลบนหน้าต่างโปรแกรมภาษาอาร์ดังรูปที่ 4.1



รูปที่ 4.1 ผลการทดสอบการนำไฟล์เข้าโดยใช้ XLConnect

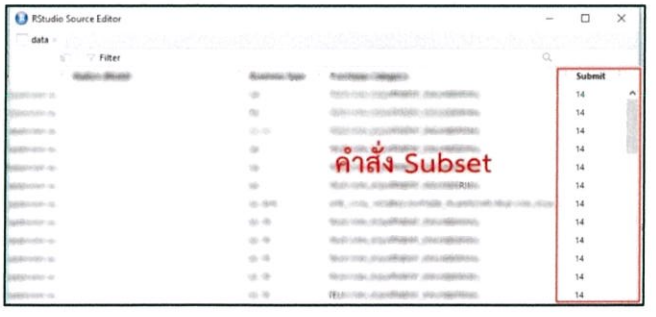
ผลการดังตัวอย่างการเขียนโปรแกรมอาร์เรียกใช้งานไลบรารี่ XLConnect เมื่อต้องการดึงไฟล์นามสกุลเอกเซลที่มีชื่อว่า “Project01” และมีหน้าชีทภายในไฟล์เอกเซลว่า “Task” ข้อมูลแสดงดังรูปที่ 3.4 โดยในทีนี้จะดึงข้อมูลเฉพาะคอลัมน์ที่ 11 ถึง คอลัมน์ที่ 22 เท่านั้น

4.2 ผลการทดสอบการใช้คำสั่ง subset และ substr

ผลการทดสอบการใช้คำสั่ง subset และ substr โดย Subset จะทำการกรองคอลัมน์ข้อมูลที่ตรงกับเงื่อนไข และ Substr จะทำการดึงตัวอักษรที่สนใจในคอลัมน์ออกมา โปรแกรมนี้ใช้คำสั่ง substr ดึงตัวอักษรตัวที่ 11 ถึง 13 ของคอลัมน์ที่มีชื่อว่า “Submit End Date” ของข้อมูลในรูปที่ 3.5 ไปเก็บไว้ในคอลัมน์ที่ชื่อว่า “Submit” ผลการทดสอบการใช้คำสั่ง Substr แสดงดังรูปที่ 4.2

รูปที่ 4.2 การทดสอบคำสั่ง Substr

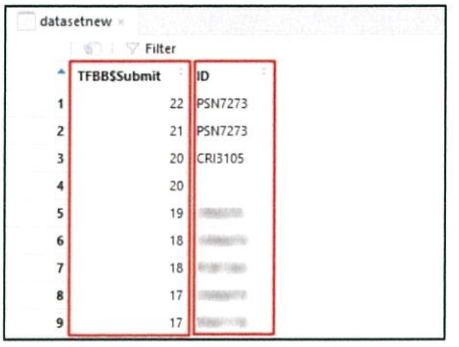
จากนั้นนำข้อมูลข้างต้นมาทำการกรองข้อมูลที่สนใจโดยใช้คำสั่ง subset แล้วนำข้อมูลที่ผ่านการกรองนี้ไปเก็บเป็นดาต้าเฟรมใหม่ที่ชื่อว่า “data” โดยต้องการกรองเฉพาะแถวที่มีข้อมูลคอลัมน์ “Submit” เป็น “14” เท่านั้น ผลการทดสอบการใช้คำสั่ง Subset กรองข้อมูลแสดงดังรูปที่ 4.3



รูปที่ 4.3 การทดสอบคำสั่ง Subset

4.3 ผลการทดสอบการใช้คำสั่ง cbind

ผลการทดสอบการใช้คำสั่ง cbind ซึ่งเป็นคำสั่งที่ใช้ในการนำคอลัมน์ของแต่ละไฟล์ข้อมูลมารวมกัน เมื่อมีข้อมูลไฟล์ A และข้อมูลไฟล์ B ดังรูปที่ 3.6 (ก) และ (ข) โดยให้ข้อมูลไฟล์ A เป็นไฟล์ “datasetnew” มารวมกับข้อมูลไฟล์ B คือคอลัมน์ที่มีชื่อว่า “Customer Site ID” ของไฟล์ “TFBB” แล้วตั้งชื่อคอลัมน์ใหม่ว่า “ID” ผลการทดสอบการรันโปรแกรมอาร์ทดสอบคำสั่ง cbind แสดงดังรูปที่ 4.4


datasetnew

Filter

	TFBBSSubmit	ID
1	22	PSN7273
2	21	PSN7273
3	20	CRI3105
4	20	
5	19	
6	18	
7	18	
8	17	
9	17	

รูปที่ 4.4 การทดสอบคำสั่ง cbind

4.4 ผลการทดสอบการใช้คำสั่ง merge

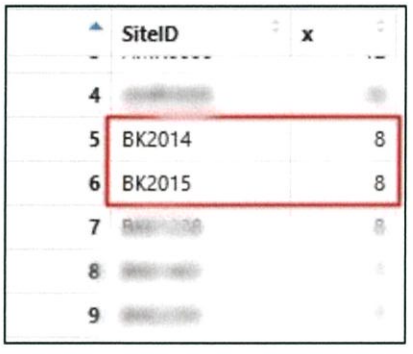
คำสั่ง merge มีลักษณะการทำงานที่คล้ายกับ VLOOKUP ในโปรแกรมเอกเซลคือเป็นการรวมข้อมูลสองชุดโดยใช้คอลัมน์หรือชื่อแถวร่วมกัน โดยในที่นี้เมื่อต้องการรวมข้อมูล 2 ชุดคือข้อมูล datasetA และข้อมูล datasetB ดังรูปที่ 3.7 (ก) และ (ข) แล้วต้องการรวมไฟล์โดยใช้คอลัมน์ที่มีชื่อว่า “SiteID” ร่วมกัน ผลการทดสอบการใช้คำสั่ง merge แสดงดังรูปที่ 4.5

SiteID	datasetA	datasetB
AIS	7	10
AMR	12	18
AMR	18	18
BK	8	20
BK	8	9
BKK	8	8
BKK	7	18

รูปที่ 4.5 การทดสอบคำสั่ง merge

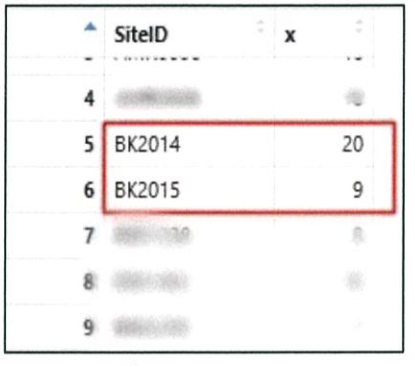
4.5 ผลการทดสอบการใช้คำสั่ง Aggregate

คำสั่ง Aggregate เป็นคำสั่งที่ใช้สำหรับคำนวณค่าผลรวมของข้อมูลทั้งตารางหรือเพียงบางกลุ่มของแถวข้อมูล โดยในที่นี้เมื่อมีข้อมูลชื่อว่า “datasetnew” คำสั่ง จะทำการจัดกลุ่มข้อมูลที่มี “ID”ซ้ำกัน จากนั้นจะคำนวณโดยใช้คำสั่ง min และ max เพื่อการหาค่าต่ำสุดและสูงสุดของข้อมูลในคอลัมน์แถวที่ 1 ของข้อมูลในรูปที่ 3.8 ผลการทดสอบการใช้คำสั่ง Aggregate เมื่อคำนวณโดยใช้คำสั่ง min และ max แสดงดังรูปที่ 4.6 (ก) และ (ข) ตามลำดับ สังเกตว่าในข้อมูลที่มี “ID” เป็น BK2014 นั้น จะมีค่าต่ำสุดเป็น และค่าสูงสุดเป็น 20



	SiteID	x
4	[illegible]	[illegible]
5	BK2014	8
6	BK2015	8
7	[illegible]	[illegible]
8	[illegible]	[illegible]
9	[illegible]	[illegible]

รูปที่ 4.6 (ก) การทดสอบ
การใช้คำสั่ง Aggregate คำนวณค่าต่ำสุด

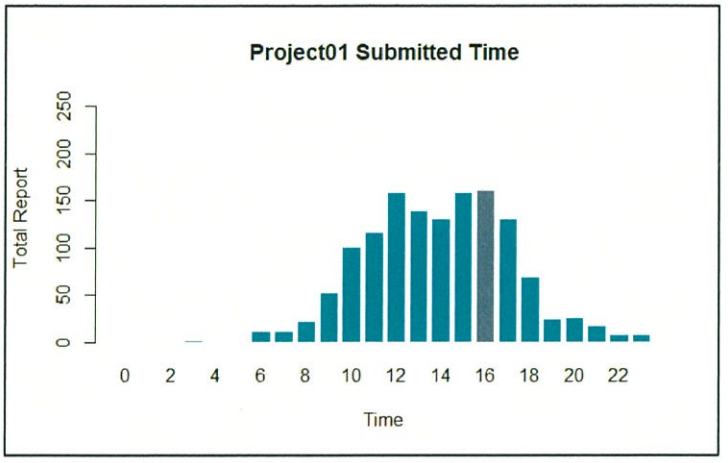


	SiteID	x
4	[illegible]	[illegible]
5	BK2014	20
6	BK2015	9
7	[illegible]	[illegible]
8	[illegible]	[illegible]
9	[illegible]	[illegible]

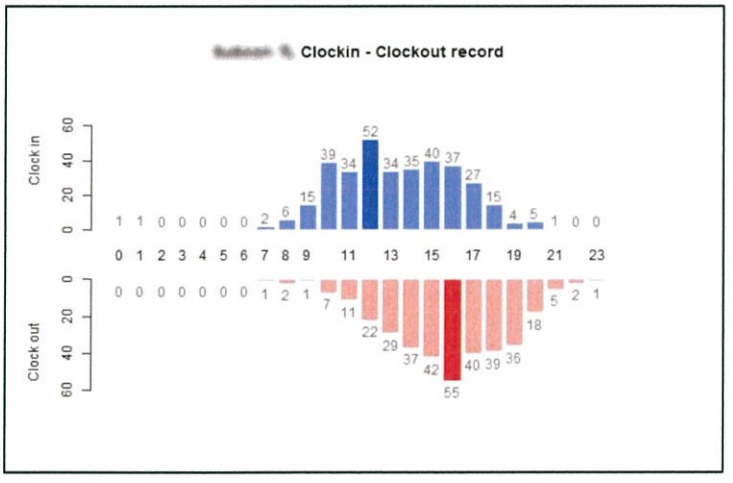
รูปที่ 4.6 (ข) การทดสอบ
การใช้คำสั่ง Aggregateคำนวณค่าสูงสุด

4.6 ผลการทดสอบการพล็อตกราฟแท่ง

ผลการทดสอบการพล็อตกราฟแท่งโดยใช้คำสั่ง barplot เมื่อนำข้อมูลที่มีการแจกแจงมาพล็อตกราฟแท่งแสดงผลการทดสอบการพล็อตกราฟดังรูปที่ 4.7 โดยมีการกำหนดชื่อแกน X และแกน Y กำหนดชื่อกราฟ กำหนดสีให้กับแท่งกราฟ กำหนดสีขอบของเส้นกราฟ และกำหนดขนาดของแกน Y นอกจากนี้เมื่อทดสอบการพล็อตกราฟแบบ 2 กราฟอยู่ในหน้าเดียวก็สามารถทำได้โดยแสดงผลการทดสอบการพล็อตแบบ 2 กราฟดังรูปที่ 4.8



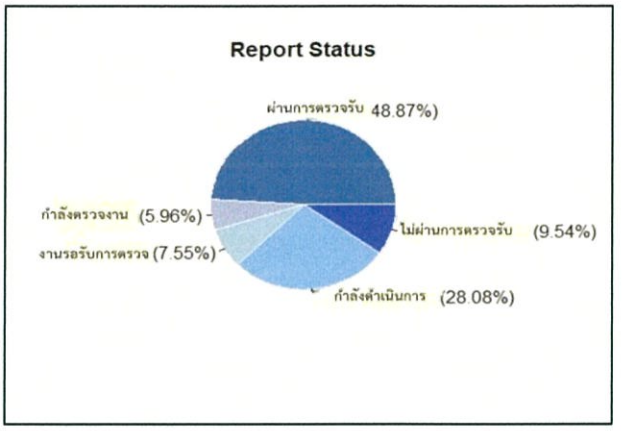
รูปที่ 4.7 การทดสอบการพล็อตกราฟแท่ง



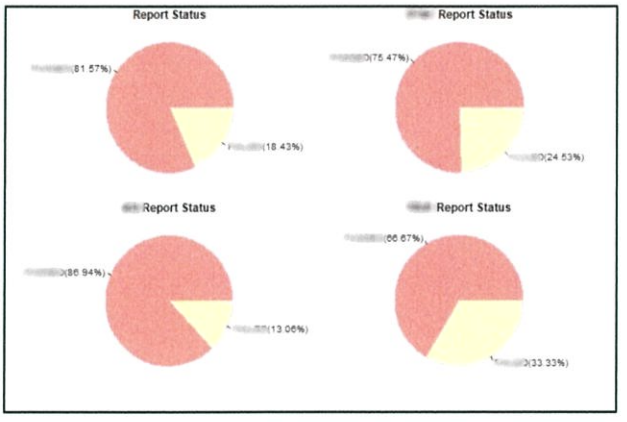
รูปที่ 4.8 การทดสอบการพล็อตกราฟแท่งแบบ 2 กราฟ

4.7 ผลการทดสอบการพล็อตกราฟวงกลม

การทดสอบการพล็อตกราฟวงกลมโดยใช้คำสั่ง pie เมื่อมีข้อมูลแสดงการแจกแจงสถานะของงาน และจำนวนงาน สามารถเรียกคำสั่ง pie() มาใช้ในการพล็อตกราฟวงกลม สามารถกำหนดชื่อกราฟและกำหนดสีของกราฟได้เช่นเดียวกับการพล็อตกราฟแท่ง ผลการพล็อตกราฟแสดงดังรูปที่ 4.9 โดยข้อมูลจะแสดงเป็นสัดส่วนร้อยละของข้อมูลทั้งหมด และการทดสอบการพล็อตกราฟวงกลม 4 กราฟอยู่บนหน้าเดียวกัน แสดงดังรูปที่ 4.10



รูปที่ 4.9 การทดสอบการพล็อตกราฟวงกลม



รูปที่ 4.10 การทดสอบการพล็อตกราฟวงกลมแบบ 4 กราฟ

4.8 ผลการทดสอบการการสร้างแอพพลิเคชั่นเว็บจาก Shiny

Shiny เป็นแพคเกจของภาษาอาร์สำหรับการสร้างแอพพลิเคชั่นเว็บแบบโต้ตอบ (interactive web) โดยจะทำการทดสอบการเขียนโปรแกรมสองส่วนคือส่วน user-interface และส่วน server script ส่วนของ user-interface จะควบคุมรูปแบบและลักษณะที่ปรากฏบนแอพพลิเคชั่น ส่วน server script จะเป็นตัวกำหนดและประมวลโปรแกรมที่ต้องการให้แสดง

1)) ทดสอบส่วนของ user-interface

Shiny ใช้ฟังก์ชัน fluidPage เพื่อสร้างหน้าจอที่ปรับให้เข้ากับขนาดของหน้าต่างบราวเซอร์ของผู้ใช้ในทีนี้จะสร้างหน้าต่างของ User interface ให้โต้ตอบกับผู้ใช้ได้ โดยจะทำการเก็บค่าจากผู้ใช้ เมื่อผู้ใช้เปลี่ยนรูปแบบการส่งค่า ข้อมูลที่แสดงผลก็จะเปลี่ยนไปเช่นกัน ทดสอบการสร้างส่วนโปรแกรมอาร์นี้ที่ผู้ใช้สามารถโต้ตอบได้ใน Sidebar panel 3 รูปแบบ รูปแบบที่ 1 กล่องที่มีตัวเลือกให้เลือก ทำการทดสอบโดยการรันโปรแกรมภาษาอาร์แสดงผลดังรูปที่ 4.11

รูปที่ 4.11 การทดสอบการสร้างกล่องที่มีตัวเลือกให้เลือก

รูปแบบที่ 2 ชุดปุ่มวิทยุ ทำการทดสอบโดยการรันโปรแกรมภาษาอาร์แสดงผลดังรูปที่ 4.12

รูปที่ 4.12 การทดสอบการสร้างชุดปุ่มวิทยุ

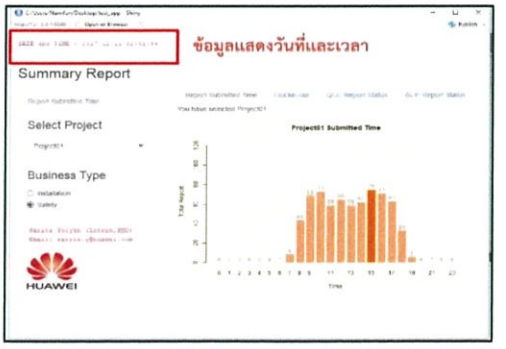
รูปแบบที่ 3 การอัพโหลดไฟล์นามสกุล .CSV ทำการทดสอบโดยการรันโปรแกรมภาษาอาร์ แสดงผลดังรูปที่ 4.13

รูปที่ 4.13 การทดสอบการสร้างการอัพโหลดไฟล์นามสกุล .CSV

2) การทดสอบส่วนของ server script

ส่วนของ server script จะแสดงค่าผลลัพธ์ โดยจะมีค่าสอดคล้องกับส่วนของ User Interface ข้อมูลผลลัพธ์ที่นำมาแสดงเป็นไปตามโปรแกรมอาร์ที่ทำการออกแบบไว้ โดยในที่นี้ทำการออกแบบโปรแกรมวิเคราะห์และจัดเรียงข้อมูล ดังนั้นจะนำค่ากราฟที่ผ่านการจัดเรียงข้อมูลแล้ว มาแสดงผลในส่วนนี้ ทำการทดสอบการแสดงผล 3 รูปแบบ

รูปแบบที่ 1 ทดสอบการแสดงข้อมูลเวลาบนหน้าต่างแอพพลิเคชั่นเว็บ ข้อมูลนี้จะแสดงผลลัพธ์เป็นเวลาและวันที่ ออกทางหน้าต่างแอพพลิเคชั่นเว็บ ผลการทดสอบแสดงดังรูปที่ 4.14

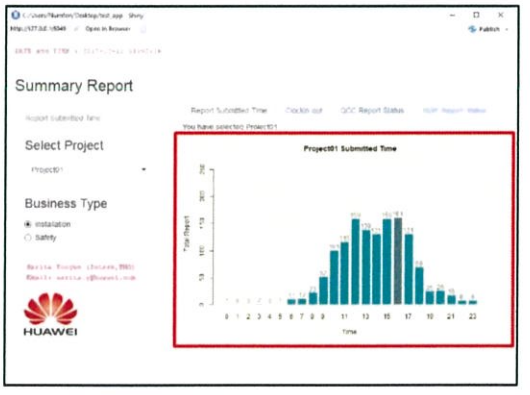


รูปที่ 4.14 ทดสอบการแสดงข้อมูลเวลา

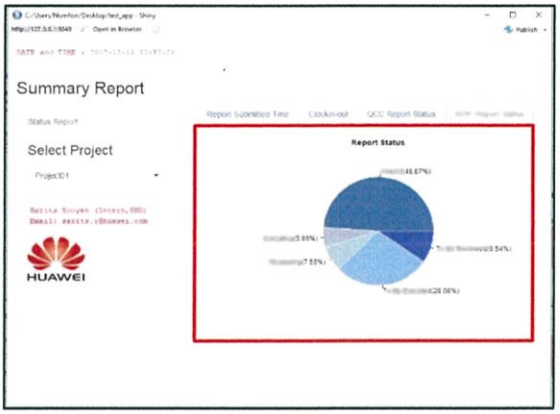
รูปแบบที่ 2 ทดสอบการแสดงข้อมูลที่ผู้ใช้ทำเลือก เมื่อผู้ใช้งานทำการเลือกข้อมูลในส่วนของ user-interface แล้วนั้น ข้อมูลนี้จะถูกจัดเก็บแล้วนำมาแสดงค่าที่หน้าต่างแอพพลิเคชั่นเว็บเช่น เมื่อผู้ใช้งานทำการเลือกโปรเจค03 โปรแกรมจะรับค่าว่าผู้ใช้งานทำการเลือกโปรเจค03 แล้วนำมาแสดงเป็นตัวอักษรบนหน้าจอว่าผู้ใช้งานทำการเลือกข้อมูลแบบใด ผลการทดลองแสดงดังรูปที่ 4.15

รูปที่ 4.15 ทดสอบการแสดงข้อมูลที่ผู้ใช้ทำเลือก

รูปแบบที่ 3 ทดสอบการแสดงผลลัพธ์กราฟบนหน้าต่างแอพพลิเคชั่นเว็บ เมื่อผู้ใช้งานทำการเลือกข้อมูลที่ต้องการดูแล้วนั้น โปรแกรมจะจัดเรียงข้อมูลและนำข้อมูลมมาแสดงผลลัพธ์ในรูปแบบของกราฟ ผลการทดลองแสดงกราฟแท่งบนหน้าต่างแอพพลิเคชั่นเว็บดังรูปที่ 4.16 และผลการทดลองแสดงกราฟวงกลมบนหน้าต่างแอพพลิเคชั่นเว็บแสดงดังรูปที่ 4.17



รูปที่ 4.16 การทดลองแสดงกราฟแท่งบนหน้าต่างแอพพลิเคชั่นเว็บ



รูปที่ 4.17 การทดลองแสดงกราฟวงกลมบนหน้าต่างแอพพลิเคชั่นเว็บ

บทที่ 5

สรุปผลและข้อเสนอแนะ

โครงงานการออกแบบโปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐานโดยใช้ภาษาอาร์ถูกออกแบบมาสำหรับการใช้งานของพนักงานในองค์กรในการจัดทำข้อมูล สรุปและวิเคราะห์ผลข้อมูล เพื่อนำเสนอให้กับผู้บริหารในการวางแผนโครงการต่างๆ โดยโปรแกรมที่ทำการออกแบบนี้สามารถนำมาช่วยบริษัททั้งในด้านของการช่วยจัดเรียงข้อมูลเกี่ยวการติดตั้งเสาสัญญาณ การดูความคืบหน้าของโครงการ ตลอดจนช่วยดูประสิทธิภาพการตรวจรับงานของพนักงานภายในองค์กร นอกจากนี้ยังช่วยประหยัดเวลาให้กับพนักงานภายในองค์กรอีกด้วย

5.1 สรุปผล

โปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐานโดยใช้ภาษาอาร์เกิดขึ้นเนื่องจากการมองเห็นปัญหาของบริษัทในด้านในการจัดทำรายงานสรุปผลข้อมูล โดยจากเดิมการประชุมของผู้บริหารแต่ละครั้งนั้น จะต้องการข้อมูลที่นำไปใช้ในการวางแผนบริหารการจัดการโครงการต่างๆ ตลอดจนหาแนวทางแก้ไขเกี่ยวกับปัญหาการติดตั้งที่เกิดขึ้น ซึ่งข้อมูลที่ถูกนำมาวิเคราะห์และจัดทำเป็นกราฟแสดงผลนั้นจะใช้วิธีการจัดเรียงข้อมูลโดยพนักงานภายในบริษัท แต่เนื่องจากข้อมูลอยู่ในรูปแบบของ Big Data ขนาดใหญ่ที่ประกอบไปด้วยข้อมูลจำนวนมาก ทำให้การสรุปผลข้อมูลในแต่ละครั้งใช้เวลาที่นานและเป็นการเพิ่มงานให้กับพนักงานในบริษัท ด้วยเหตุนี้จึงมีออกแบบโปรแกรมภาษาอาร์ให้สามารถนำมาช่วยในการจัดเรียงและแสดงผลข้อมูลได้ โดยโปรแกรมที่ทำการออกแบบประกอบไปด้วยข้อมูลทั้งสิ้น 4 ข้อมูล แต่ละข้อมูลถูกออกแบบให้เลือกโครงการและรายละเอียดต่างๆได้ ทำการแสดงผลทั้งรูปแบบกราฟวงกลมและกราฟแท่ง โดยทำการออกแบบโปรแกรมเป็นแอพพลิเคชั่นเว็บที่สามารถโต้ตอบกับผู้ใช้งาน ซึ่งข้อมูลจะถูกแสดงผลตามที่ผู้ใช้งานต้องการ สามารถนำมาใช้งานได้ง่าย

5.2 ข้อเสนอแนะ

ภาษาอาร์ที่นำมาใช้ในการออกแบบโปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐานนั้นเป็นภาษาที่ต้องอาศัยเวลาในการทดสอบคำสั่งต่างๆ ผู้ที่จะทำการเขียนโปรแกรมต้องเกิดความเข้าใจทั้งรายละเอียดของข้อมูลและคำสั่งที่ใช้ในการจัดเรียงข้อมูล อีกทั้งภาษาอาร์ข้อเสียคือข้อมูลต่างๆ จะถูกบันทึกไว้ในหน่วยความจำของคอมพิวเตอร์ ซึ่งทำให้กินพื้นที่ทรัพยากรหน่วยความจำและทำให้โปรแกรมต่างๆ ทำงานได้ช้าลง ส่งผลให้คอมพิวเตอร์ทำงานช้าลงด้วย สามารถแก้ไขได้โดยการเพิ่มแรมให้กับเครื่องคอมพิวเตอร์ นอกจากนี้เครื่องคอมพิวเตอร์ที่ต้องการใช้งานโปรแกรมประมวลผลข้อมูลสรุปงานติดตั้งสถานีฐานโดยใช้ภาษาอาร์นั้น ต้องมีการติดตั้งภาษาอาร์และแพคเกจที่ใช้ในโปรแกรมก่อน จึงจะสามารถใช้งานโปรแกรมได้

บรรณานุกรม

[1] ผศ.ดร.โชคชัย แสงดาว. “วิวัฒนาการโทรศัพท์เคลื่อนที่”
http://www.eng.mut.ac.th/article_detail.php?id=62

[1] พงศ์พันธุ์ ปริยวงศ์. “การวิเคราะห์สถานีฐานระบบโทรศัพท์เคลื่อนที่”
http://www.thailandindustry.com/onlinemag/view2.php?id=133&extend=semi&section=83&issues=7

[2] “Telecommunication Towers : เสาสัญญาณโทรคมนาคม”
http://telecom10.blogspot.com/2015/04/

[3] “ภาษาอาร์”
https://th.wikipedia.org/wiki/ภาษาอาร์

[4] วิโรจน์ อรุณมานะกุล. “สถิติและการใช้โปรแกรมอาร์”
http://pioneer.chula.ac.th/~awirote/courses/res-tech-ling/statistics-and-r.pdf

[5] “แนะนำภาษาอาร์สำหรับความหมายที่ซ่อนอยู่ใต้ข้อมูล”
https://www.blognone.com/node/94313

[6] วนิดา พงษ์ศักดิ์ชาติ “Statistical Computing”
https://www.scribd.com/doc/34869064/การใช-โปรแกรมR

ภาคผนวก

โปรแกรมภาษาอาร์

```
library(shiny)
shinyui <- fluidPage(
        code(textOutput("currentTime")),
        titlePanel("Summary Report"),
        sidebarLayout(
        sidebarPanel(
        conditionalPanel(condition="input.conditionedPanels==1",
                helpText("Report Submitted Time"),
                selectInput("var", = h3("Select Project"),
                        choices = list("Project01",
                                "Project02",
                                "Project03",
                                "Project04",
                                "Project05"),
                        selected = "Project01"),
                radioButtons("radio", h3("Business Type"),
                         choices = list("Installation" = 1, "Safety" = 2),
                         selected = 1)),
        conditionalPanel(condition="input.conditionedPanels==2",
                helpText("Clockin-Clockout"),
                selectInput("var1",
                        label = h3("Select Project"),
                        choices = list("Project01",
                                "Project02",
                                "Project03",
                                "Project04",
                                "Project05"),
                        selected = "Project01")) ,
        conditionalPanel(condition="input.conditionedPanels==3",
```

```
                helpText("Status Report"),
                fileInput("file1", label = h3("File input .csv file"))),
        conditionalPanel(condition="input.conditionedPanels==4",
                helpText("Status Report"),
                selectInput("var2",
                        label = h3("Select Project"),
                        choices = list("Project01",
                                "Project02",
                                "Project03",
                                "Project04",
                                "Project05"),
                        selected = "Project01")) ,
    br(),
    code('Sarita Yooyen (Intern,PMO)'),
    br(),
    code('Email: sarita.y@huawei.com'),
    br(),
    br(),
    img(src = "123.png", height = 80, width = 100,Position = 5),
    br(),
    br(),width = 4),
mainPanel(
tabsetPanel(
    tabPanel("Report Submitted Time",
    value=1,textOutput("selected_var"),plotOutput("myPlot")),
    tabPanel("Clockin-out", value=2 ,textOutput("selected_var1"),plotOutput("myPlot1")),
    tabPanel("QCC Report Status", value=3,plotOutput("myPlot2")),
    tabPanel("ISDP Report Status", value=4,plotOutput("myPlot3"))
    , id = "conditionedPanels"))))
```

```
Shinyserver <- function(input, output, session ) {
output$currentTime <- renderText({
        invalidateLater(1000, session)
        paste("DATE and TIME :", Sys.time())})
output$selected_var <- renderText({
         paste("You have selected", input$var) })
output$selected_var1 <- renderText({
        paste("You have selected", input$var1) })
output$myPlot <- renderPlot({
        if(input$var == "Project01"   && input$radio == 1  ){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project01.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Installation")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project01 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }
```

```
else if(input$var == "Project01"  && input$radio == 2){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project01.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Safety")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project01 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project02" && input$radio == 1){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project02.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Installation")
        y <- table(data3$Submit)
```

```
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project02 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project02" && input$radio == 2){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project02.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Safety")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project02 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
```

```
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project03" && input$radio == 1){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project03.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Installation")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project03 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project03" && input$radio == 2){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project03.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Safety")
```

```
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project03 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project04"&& input$radio == 1){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project04.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Installation")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
```

```
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project04 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project04"&& input$radio == 2){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project04.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Safety")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project04 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else if(input$var == "Project05" && input$radio == 1){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project05.xlsx"
        T <- loadWorkbook(Truemove)
```

```
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Installation")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project05 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }

else {
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/ Project05.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T, sheet = " Task " ,startCol = 11, endCol = 22 )
        TMBB$Submit<- as.numeric(substr(TMBB$Submit.End.Date,11,13))
        data3 <- subset(TMBB, Submit != "" && Business.Type == "Safety")
        y <- table(data3$Submit)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , y, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m4$Freq[which(is.na(m4$Freq))] <- "0"
        m4$Freq <- as.numeric(as.character(m4$Freq))
        y <- m4$Freq
```

```
        par(bg= "#FFFFF0")
mybar = barplot(y,ylab = "Total Report",xlab = "Time",names.arg = m4$Time,main =
"Project05 Submitted Time",col = ifelse(y == max(y),"cadetblue4","lightseagreen"),
border="white", ylim = c(0,max(y)+100))
text(mybar, y , paste(y),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey") }})

output$myPlot1 <- renderPlot({
if(input$var1 == "Project01"){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project01.xlsx"
        T <- loadWorkbook(Truemove)
        TFBB <-readWorksheet(T , sheet = " Detail " ,startCol = 4, endCol = 12)
        TFBB$Submit<- as.numeric(substr(TFBB$ Time,11,13))
        datasetnew <- as.data.frame(TFBB$Submit)
        datasetnew <- cbind(datasetnew, ID = TFBB$Customer)
        `datasetA` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), min)
        datasetA <- subset(datasetA, SiteID != "")
        `datasetB` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), max)
        dataset = merge(x = datasetA, y = datasetB, by = "SiteID", all = TRUE)
        colnames(dataset) <- c("DU ID", "ClockIn", "ClockOut")
        dataset <- dataset[ which(dataset$ClockIn<=dataset$ClockOut),]
        WW <- table(dataset$ClockIn)
        EE <- table(dataset$ClockOut)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , WW, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m5 = merge(m4 ,EE , by.x  = "Time" , by.y = "Var1" , all.x = TRUE)
        m5$Freq.x[which(is.na(m5$Freq.x))] <- "0"
        m5$Freq.y[which(is.na(m5$Freq.y))] <- "0"
```

```
        m5$Freq.x <- as.numeric(as.character(m5$Freq.x))
        m5$Freq.y <- as.numeric(as.character(m5$Freq.y))
        b = as.numeric(max(m5$Freq.x)+max(m5$Freq.y)+max(m5$Freq.x))/2
        par(mfrow=c(2,1))
        par(bg= "Snow")
        par(mar=c(0.5,5,5,5))
mybar = barplot(m5$Freq.x,ylab = "Clock in", names.arg = m5$Time,main = "Clockin -
Clockout record",col = ifelse(m5$Freq.x == max(m5$Freq.x),"#3366FF","#6495ED"),
border="white", ylim = c(0,b) )
text(mybar, m5$Freq.x  , paste(m5$Freq.x),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey")
par(mar=c(2.5,5,2.5,5))
mybar1 = barplot(m5$Freq.y,ylab = "Clock out",main = "",col = ifelse(m5$Freq.y ==
max(m5$Freq.y),"#FF3333","#FF9999"), border="white", ylim = c(b,0),names.arg = m5$Time
, xaxt="n" )
text(mybar1, m5$Freq.y , paste(m5$Freq.y),pos = 3 , offset = -1 ,cex=1 ,col = "DimGrey") }

else if(input$var1 == "Project02"){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project02.xlsx"
        T <- loadWorkbook(Truemove)
        TFBB <-readWorksheet(T , sheet = " Detail " ,startCol = 4, endCol = 12)
        TFBB$Submit<- as.numeric(substr(TFBB$ Time,11,13))
        datasetnew <- as.data.frame(TFBB$Submit)
        datasetnew <- cbind(datasetnew, ID = TFBB$Customer)
        `datasetA` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), min)
        datasetA <- subset(datasetA, SiteID != "")
        `datasetB` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), max)
        dataset = merge(x = datasetA, y = datasetB, by = "SiteID", all = TRUE)
```

```
        colnames(dataset) <- c("DU ID", "ClockIn", "ClockOut")
        dataset <- dataset[ which(dataset$ClockIn<=dataset$ClockOut),]
        WW <- table(dataset$ClockIn)
        EE <- table(dataset$ClockOut)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , WW, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m5 = merge(m4 ,EE , by.x  = "Time" , by.y = "Var1" , all.x = TRUE)
        m5$Freq.x[which(is.na(m5$Freq.x))] <- "0"
        m5$Freq.y[which(is.na(m5$Freq.y))] <- "0"
        m5$Freq.x <- as.numeric(as.character(m5$Freq.x))
        m5$Freq.y <- as.numeric(as.character(m5$Freq.y))
        b = as.numeric(max(m5$Freq.x)+max(m5$Freq.y)+max(m5$Freq.x))/2
        par(mfrow=c(2,1))
        par(bg= "Snow")
        par(mar=c(0.5,5,5,5))
mybar = barplot(m5$Freq.x,ylab = "Clock in", names.arg = m5$Time,main = "Clockin -
Clockout record",col = ifelse(m5$Freq.x == max(m5$Freq.x),"#3366FF","#6495ED"),
border="white", ylim = c(0,b) )
text(mybar, m5$Freq.x  , paste(m5$Freq.x),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey")
par(mar=c(2.5,5,2.5,5))
mybar1 = barplot(m5$Freq.y,ylab = "Clock out",main = "",col = ifelse(m5$Freq.y ==
max(m5$Freq.y),"#FF3333","#FF9999"), border="white", ylim = c(b,0),names.arg = m5$Time
, xaxt="n" )
text(mybar1, m5$Freq.y , paste(m5$Freq.y),pos = 3 , offset = -1 ,cex=1 ,col = "DimGrey") }

else if(input$var1 == "Project03"){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project03.xlsx"
```

```
T <- loadWorkbook(Truemove)
TFBB <-readWorksheet(T , sheet = " Detail " ,startCol = 4, endCol = 12)
TFBB$Submit<- as.numeric(substr(TFBB$ Time,11,13))
datasetnew <- as.data.frame(TFBB$Submit)
datasetnew <- cbind(datasetnew, ID = TFBB$Customer)
`datasetA` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), min)
datasetA <- subset(datasetA, SiteID != "")
`datasetB` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), max)
dataset = merge(x = datasetA, y = datasetB, by = "SiteID", all = TRUE)
colnames(dataset) <- c("DU ID", "ClockIn", "ClockOut")
dataset <- dataset[ which(dataset$ClockIn<=dataset$ClockOut),]
WW <- table(dataset$ClockIn)
EE <- table(dataset$ClockOut)
name = data.frame(Time = c(0:23))
m4 = merge(name , WW, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
m5 = merge(m4 ,EE , by.x  = "Time" , by.y = "Var1" , all.x = TRUE)
m5$Freq.x[which(is.na(m5$Freq.x))] <- "0"
m5$Freq.y[which(is.na(m5$Freq.y))] <- "0"
m5$Freq.x <- as.numeric(as.character(m5$Freq.x))
m5$Freq.y <- as.numeric(as.character(m5$Freq.y))
b = as.numeric(max(m5$Freq.x)+max(m5$Freq.y)+max(m5$Freq.x))/2
par(mfrow=c(2,1))
par(bg= "Snow")
par(mar=c(0.5,5,5,5))
mybar = barplot(m5$Freq.x,ylab = "Clock in", names.arg = m5$Time,main = "Clockin -
Clockout record",col = ifelse(m5$Freq.x == max(m5$Freq.x),"#3366FF","#6495ED"),
border="white", ylim = c(0,b) )
text(mybar, m5$Freq.x  , paste(m5$Freq.x),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey")
par(mar=c(2.5,5,2.5,5))
```

```
mybar1 = barplot(m5$Freq.y,ylab = "Clock out",main = "",col = ifelse(m5$Freq.y ==
max(m5$Freq.y),"#FF3333","#FF9999"), border="white", ylim = c(b,0),names.arg = m5$Time
, xaxt="n" )
text(mybar1, m5$Freq.y , paste(m5$Freq.y),pos = 3 , offset = -1 ,cex=1 ,col = "DimGrey") }

else if(input$var1 == "Project04"){
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project04.xlsx"
        T <- loadWorkbook(Truemove)
        TFBB <-readWorksheet(T , sheet = " Detail " ,startCol = 4, endCol = 12)
        TFBB$Submit<- as.numeric(substr(TFBB$ Time,11,13))
        datasetnew <- as.data.frame(TFBB$Submit)
        datasetnew <- cbind(datasetnew, ID = TFBB$Customer)
        `datasetA` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), min)
        datasetA <- subset(datasetA, SiteID != "")
        `datasetB` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), max)
        dataset = merge(x = datasetA, y = datasetB, by = "SiteID", all = TRUE)
        colnames(dataset) <- c("DU ID", "ClockIn", "ClockOut")
        dataset <- dataset[ which(dataset$ClockIn<=dataset$ClockOut),]
        WW <- table(dataset$ClockIn)
        EE <- table(dataset$ClockOut)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , WW, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m5 = merge(m4 ,EE , by.x  = "Time" , by.y = "Var1" , all.x = TRUE)
        m5$Freq.x[which(is.na(m5$Freq.x))] <- "0"
        m5$Freq.y[which(is.na(m5$Freq.y))] <- "0"
        m5$Freq.x <- as.numeric(as.character(m5$Freq.x))
        m5$Freq.y <- as.numeric(as.character(m5$Freq.y))
```

```
        b = as.numeric(max(m5$Freq.x)+max(m5$Freq.y)+max(m5$Freq.x))/2
        par(mfrow=c(2,1))
        par(bg= "Snow")
        par(mar=c(0.5,5,5,5))
mybar = barplot(m5$Freq.x,ylab = "Clock in", names.arg = m5$Time,main = "Clockin -
Clockout record",col = ifelse(m5$Freq.x == max(m5$Freq.x),"#3366FF","#6495ED"),
border="white", ylim = c(0,b) )
text(mybar, m5$Freq.x  , paste(m5$Freq.x),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey")
par(mar=c(2.5,5,2.5,5))
mybar1 = barplot(m5$Freq.y,ylab = "Clock out",main = "",col = ifelse(m5$Freq.y ==
max(m5$Freq.y),"#FF3333","#FF9999"), border="white", ylim = c(b,0),names.arg = m5$Time
, xaxt="n" )
text(mybar1, m5$Freq.y , paste(m5$Freq.y),pos = 3 , offset = -1 ,cex=1 ,col = "DimGrey") }

else{
        options(java.parameters = "-Xmx4g")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project05.xlsx"
        T <- loadWorkbook(Truemove)
        TFBB <-readWorksheet(T , sheet = " Detail " ,startCol = 4, endCol = 12)
        TFBB$Submit<- as.numeric(substr(TFBB$ Time,11,13))
        datasetnew <- as.data.frame(TFBB$Submit)
        datasetnew <- cbind(datasetnew, ID = TFBB$Customer)
        `datasetA` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), min)
        datasetA <- subset(datasetA, SiteID != "")
        `datasetB` <-  aggregate(TFBB$Submit, list(SiteID = datasetnew$ID), max)
        dataset = merge(x = datasetA, y = datasetB, by = "SiteID", all = TRUE)
        colnames(dataset) <- c("DU ID", "ClockIn", "ClockOut")
        dataset <- dataset[ which(dataset$ClockIn<=dataset$ClockOut),]
```

```
        WW <- table(dataset$ClockIn)
        EE <- table(dataset$ClockOut)
        name = data.frame(Time = c(0:23))
        m4 = merge(name , WW, by.x = "Time" , by.y = "Var1" , all.x = TRUE)
        m5 = merge(m4 ,EE , by.x  = "Time" , by.y = "Var1" , all.x = TRUE)
        m5$Freq.x[which(is.na(m5$Freq.x))] <- "0"
        m5$Freq.y[which(is.na(m5$Freq.y))] <- "0"
        m5$Freq.x <- as.numeric(as.character(m5$Freq.x))
        m5$Freq.y <- as.numeric(as.character(m5$Freq.y))
        b = as.numeric(max(m5$Freq.x)+max(m5$Freq.y)+max(m5$Freq.x))/2
        par(mfrow=c(2,1))
        par(bg= "Snow")
        par(mar=c(0.5,5,5,5))
mybar = barplot(m5$Freq.x,ylab = "Clock in", names.arg = m5$Time,main = "Clockin -
Clockout record",col = ifelse(m5$Freq.x == max(m5$Freq.x),"#3366FF","#6495ED"),
border="white", ylim = c(0,b) )
text(mybar, m5$Freq.x  , paste(m5$Freq.x),pos = 3 , offset = 0.2 ,cex=1 ,col = "DimGrey")
par(mar=c(2.5,5,2.5,5))
mybar1 = barplot(m5$Freq.y,ylab = "Clock out",main = "",col = ifelse(m5$Freq.y ==
max(m5$Freq.y),"#FF3333","#FF9999"), border="white", ylim = c(b,0),names.arg = m5$Time
, xaxt="n" )
text(mybar1, m5$Freq.y , paste(m5$Freq.y),pos = 3 , offset = -1 ,cex=1 ,col = "DimGrey") }
})

output$myPlot2 <- renderPlot({
data2 <- input$file1
if (is.null(data2))
return(NULL)
```

```
data2 = read.csv(choose.files(), header =T)
data2$Review.status <- factor(data2$Review.status,levels = c("PASSED","FAILED"))
table(data2$Review.status)
AIS <- subset(data2, Project == "AIS")
DTAC <- subset(data2, Project == "DTAC")
TT <- subset(data2, Project == "TRUE")
DATA <- subset(data2, Project == "TRUE" |Project == "DTAC" | Project == "AIS")
H <- table(DATA$Review.status)
PER <-sprintf("%s(%.2f%%)",rownames(H),H/sum(H)*100)
PER
a <- table(A$Review.status)
PERA <-sprintf("%s(%.2f%%)",rownames(a),a/sum(a)*100)
PERA
d <- table(B$Review.status)
PERD <-sprintf("%s(%.2f%%)",rownames(d),d/sum(d)*100)
PERD
tr <- table(C$Review.status)
PERTR <-sprintf("%s(%.2f%%)",rownames(tr),tr/sum(tr)*100)
PERTR
mul <- par(mfrow=c(2,2))
par(mar=c(0.5,1,1,1))
pie(H ,labels = PER , main = "Report Status",col=c("#FF9999","#FFFACD") ,border =
"PINK")
pie(d ,labels = PERD  ,main = "A Report Status",col=c("#FF9999","#FFFACD"),border =
"PINK")
pie(a , labels = PERA ,main = "B Report Status",col=c("#FF9999","#FFFACD"),border =
"PINK")
pie(tr , labels = PERTR ,main = "C Report Status",col=c("#FF9999","#FFFACD"),border =
"PINK") par(mul) })
```

```
output$myPlot3 <- renderPlot({
if(input$var2 == "Project01"){
        options(java.parameters = "-Xmx1024m")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project01 report.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T , sheet = " Task " ,startCol = 7, endCol = 21 )
        TMBB <- subset(TMBB , Business.Type == "QC-TE")
        H <- table(TMBB$Status)
        PER <-sprintf("%s(%.2f%%)",rownames(H),H/sum(H)*100)
        pie(H ,labels = PER , main = "Report
Status",col=c("#4682B4","#B0C4DE","#ADD8E6","#87CEFA","#4169E1","Pink") ,border =
"WHITE") }}

else if (input$var2 == "Project02"){
        options(java.parameters = "-Xmx1024m")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project02 report.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T , sheet = " Task " ,startCol = 7, endCol = 21 )
        TMBB <- subset(TMBB , Business.Type == "QC-TE")
        H <- table(TMBB$Status)
        PER <-sprintf("%s(%.2f%%)",rownames(H),H/sum(H)*100)
        pie(H ,labels = PER , main = "Report
Status",col=c("#4682B4","#B0C4DE","#ADD8E6","#87CEFA","#4169E1","Pink") ,border =
"WHITE") }}

else if (input$var2 == "Project03"){
        options(java.parameters = "-Xmx1024m")
        library(XLConnect)
```

```
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project03 report.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T , sheet = " Task " ,startCol = 7, endCol = 21 )
        TMBB <- subset(TMBB , Business.Type == "QC-TE")
        H <- table(TMBB$Status)
        PER <-sprintf("%s(%.2f%%)",rownames(H),H/sum(H)*100)
        pie(H ,labels = PER , main = "Report
Status",col=c("#4682B4","#B0C4DE","#ADD8E6","#87CEFA","#4169E1","Pink") ,border =
"WHITE") }}

else if (input$var2 == "Project04"){
        options(java.parameters = "-Xmx1024m")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project04 report.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T , sheet = " Task " ,startCol = 7, endCol = 21 )
        TMBB <- subset(TMBB , Business.Type == "QC-TE")
        H <- table(TMBB$Status)
        PER <-sprintf("%s(%.2f%%)",rownames(H),H/sum(H)*100)
        pie(H ,labels = PER , main = "Report
Status",col=c("#4682B4","#B0C4DE","#ADD8E6","#87CEFA","#4169E1","Pink") ,border =
"WHITE") }}

else  {
        options(java.parameters = "-Xmx1024m")
        library(XLConnect)
        Truemove <- "C:/Users/Numfon/Desktop/Rstudio/Project05 report.xlsx"
        T <- loadWorkbook(Truemove)
        TMBB <-readWorksheet(T , sheet = " Task " ,startCol = 7, endCol = 21 )
        TMBB <- subset(TMBB , Business.Type == "QC-TE")
```

```
        H <- table(TMBB$Status)
        PER <-sprintf("%s(%.2f%%)",rownames(H),H/sum(H)*100)
        pie(H ,labels = PER , main = "Report
Status",col=c("#4682B4","#B0C4DE","#ADD8E6","#87CEFA","#4169E1","Pink") ,border =
"WHITE") }}
})}
```