

การสร้างโปรแกรมลักษณะรายการ (Menu) จาก
โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC หรือ SPSS/PC⁺

A Development of Menu Program from Statistical Package
SPSS/PC or SPSS/PC⁺



เลขหมู่	04 76.73 14 56 9032 0.1
เลขทะเบียน	029701
	26 ค.ย. 2534 /

นทีกานต์ สุขเมฆสิขิกุล M.Sc. (Biostatistics) Mahidol
 ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
 มหาวิทยาลัยสงขลานครินทร์
 วิทยาเขตหาดใหญ่
 มิถุนายน 2534

บทคัดย่อ

งานวิจัยครั้งนี้ ได้นำคอมพิวเตอร์เข้ามาช่วยในการสร้างโปรแกรมลักษณะ
รายการ (Menu) เป็นภาษาไทย โดยใช้ภาษา Turbo Pascal และ IRC Thai Card
บนเครื่องไมโครคอมพิวเตอร์ IBM PC/XT พร้อมฮาร์ดดิสก์ เพื่ออำนวยความสะดวกแก่ผู้ใช้
ในการจะใช้โปรแกรมสำเร็จรูป SPSS/PC และ SPSS/PC⁺ โดยลักษณะของงานที่ทำจะ
ครอบคลุมตั้งแต่การเรียกใช้แฟ้มข้อมูล การกำหนดชื่อตัวแปร การใช้คำสั่งทางสถิติ การเก็บ
แฟ้มข้อมูลที่แสดงผลลัพธ์ รวมทั้งการออกผลลัพธ์ต่างๆ ที่ได้ ทางเครื่องพิมพ์ โปรแกรมนี้จะทำ
การสร้างชุดคำสั่งต่างๆ ของโปรแกรมสำเร็จรูป SPSS/PC และ SPSS/PC⁺ หลังจากนั้น
ผู้ใช้จะต้องนำไปประมวลผลบนเครื่องที่มีโปรแกรมสำเร็จรูป SPSS/PC และ SPSS/PC⁺
ติดตั้ง

ABSTRACT

Computer research work, in general, deals with using statistical packages for analyzing data. For those who are already familiar with statistical computing, statistical packages present SPSS/PC or SPSS/PC⁺. Therefore it is necessary to know how to write the commands available for data using the Thai menu and how to make the commands efficient.

The following research paper, a computer program consisting of Thai menus, has been developed from SPSS/PC or SPSS/PC⁺. Both of these statistical package include tutorial study commands, and commands available for defining, modifying, tabulating and analyzing data using appropriate statistical procedures.

In developing this computer program, the system has been implemented on a 16-bit microcomputer with a Thai card by using the International Research Corporation company (IRC) code and by using Turbo Pascal to create the programs.

In addition, the computer program consists of four main parts, namely, data definition, statistical procedures, managing files and utilities. It is menu-driven, and very easy to use. Choices on the menu are displayed in Thai, so that anybody can use it. After that the results of using are run on that computer, which has already installed SPSS/PC or SPSS/PC⁺.

In conclusion, the aim of this project has been to find out a user-friendly computer system which can be used with the Thai language menu and whose code is determined by IRC.

สารบัญ

หน้า

ชื่องานวิจัย

คำนำ

บทคัดย่อ

กิตติกรรมประกาศ

สารบัญ

บทที่ 1	บทนำ	1
1.1	ที่มาและความสำคัญของปัญหา	1
1.2	วัตถุประสงค์	2
1.3	ขอบเขตของการศึกษา	2
1.4	อุปกรณ์ ภาษาและโปรแกรมที่ใช้	2
1.5	นิยามศัพท์	2
1.6	ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2	โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC ⁺	4
บทที่ 3	วางแผนระบบงาน ผังการทำงานและโปรแกรมที่พัฒนา	6
3.1	วางแผนระบบงาน ผังการทำงาน	6
3.2	โปรแกรมที่พัฒนา	14
บทที่ 4	เอกสารประกอบการใช้	25
บทที่ 5	สรุป อภิปรายและข้อเสนอแนะ	28
5.1	สรุป อภิปราย	28
5.2	ข้อเสนอแนะ	29
บรรณานุกรม		30
ภาคผนวก		
ก	คู่มือการใช้งาน	31
ข	โปรแกรม	141

สารบัญม้งงาน จอภาพ และตาราง

		หน้า
ม้งงาน		
ก	รายการหลัก	7
1	ข้อมูล	7
2	คำสั่งทางสถิติ	8
2.1	ความถี่	9
2.2	สถิติพรรณนา	9
2.3	สถิติเบื้องต้นที่แบ่งกลุ่ม	9
2.4	ตารางการกระจาย	10
2.5	ทดสอบ T-Test	10
2.6	ความแปรปรวนทางเดียว	10
2.7	ความแปรปรวนหลายทาง	11
2.8	สหสัมพันธ์	11
2.9	วิเคราะห์ถดถอย	12
2.10	เขียนกราฟ	12
2.11	ทดสอบนอนพารามิเตอร์	13
3	คำสั่งช่วย	14
4	อรรถประโยชน์	14
จอภาพ		
1	ข้อมูล	33
1.1	กำหนดลักษณะข้อมูล	34
1.1.1	เรียกเพิ่มข้อมูล	35
1.1.2	บอกรายละเอียดเกี่ยวกับตัวแปร	36
1.1.2.1	กำหนดตำแหน่งคอลัมน์แบบคงที่	37
1.1.2.2	กำหนดตำแหน่งคอลัมน์แบบอิสระ	38
1.1.3	บอกค่าไม่สมบูรณ์ให้กับตัวแปร	39
1.1.4	บอกรายละเอียดเกี่ยวกับค่าตัวแปร	40
1.1.5	บอกรายละเอียดเกี่ยวกับชื่อตัวแปร	41
1.2	เปลี่ยนรูปข้อมูล	42
1.2.1	เปลี่ยนค่าของตัวแปร	43
1.2.2	สร้างตัวแปรใหม่โดยการคำนวณ	44
1.2.3	สร้างตัวแปรใหม่โดยการกำหนดเงื่อนไข	45

สารบัญ้งาน จอภาพ และตาราง (ต่อ)

	หน้า	
จอภาพ		
1.2.4	สร้างตัวแปรใหม่โดยการนับ	47
1.2.5	กำหนดน้ำหนักแก่ตัวแปร	48
1.3	คัดเลือกข้อมูล	49
1.3.1	จำนวนชุดข้อมูล	50
1.3.2	โดยวิธีการสุ่มตัวอย่าง	51
1.3.3	โดยกำหนดเงื่อนไขทางตรรก	52
1.3.4	โดยกำหนดเงื่อนไขทางตรรกแบบชั่วคราว	53
2	คำสั่งทางสถิติ	55
2.1	ความถี่	56
2.1.1	ชื่อตัวแปรของความถี่	57
2.1.2	ระบุส่วนขยายในกราฟแท่ง	58
2.1.2.1	ระบุค่าต่ำสุด, สูงสุด และความถี่ในกราฟแท่ง	59
2.1.3	ฮิสโตแกรม	60
2.1.3.1	ระบุค่าต่ำสุด, สูงสุด และความถี่ในฮิสโตแกรม	61
2.1.4	เปอร์เซ็นต์ไทล์	62
2.1.5	เอนไทล์	63
2.1.6	ค่าสถิติในความถี่	64
2.2	สถิติพรรณนา	66
2.2.1	ตัวแปรในสถิติพรรณนา	67
2.2.2	รูปแบบของการคำนวณ	68
2.2.3	ค่าสถิติในสถิติพรรณนา	69
2.3	สถิติเบื้องต้นที่ต้องการแบ่งกลุ่ม	71
2.3.1	ตัวแปรที่คำนวณค่าสถิติเบื้องต้น	72
2.3.2	ตัวแปรแบ่งกลุ่ม	73
2.3.3	รูปแบบในสถิติเบื้องต้น	74
2.3.4	ค่าสถิติในสถิติเบื้องต้นที่แบ่งกลุ่ม	75
2.4	ตารางการแจกแจง	76
2.4.1	ตัวแปรในตารางการแจกแจง	77
2.4.2	รูปแบบในตารางการแจกแจง	78
2.4.3	ค่าสถิติในตารางการแจกแจง	79

สารบัญม้งงาน จอภาพ และตาราง (ต่อ)

	หน้า
จอภาพ	
2.5 ทดสอบ T-Test	80
2.5.1 กลุ่มตัวอย่างเป็นอิสระต่อกัน	81
2.5.1.1 ตัวแบบในการทดสอบ T-Test	82
2.5.1.2 กลุ่มข้อมูลของกลุ่มตัวอย่างเป็นอิสระต่อกัน	83
2.5.2 กลุ่มตัวอย่างไม่เป็นอิสระต่อกันของการทดสอบ	84
2.5.3 รูปผลลัพธ์ของการทดสอบ T-Test	85
2.6 ความแปรปรวนทางเดียว	86
2.6.1 ตัวแปรตามในความแปรปรวนทางเดียว	87
2.6.2 ตัวแปรแบ่งกลุ่มในความแปรปรวนทางเดียว	88
2.6.3 กำหนดการแบ่งกลุ่มของแหล่งผันแปรในความแปรปรวนทางเดียว	89
2.6.4 การเปรียบเทียบ, ทดสอบค่าเฉลี่ยในรูปเวกเตอร์ในความแปรปรวนทางเดียว	90
2.6.5 กำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ย	91
2.6.6 รูปแบบของการวิเคราะห์ความแปรปรวน	92
2.6.7 ค่าสถิติในความแปรปรวน	93
2.7 ความแปรปรวนหลายทาง	94
2.7.1 ตัวแปรที่วิเคราะห์ความแปรปรวนหลายทาง	95
2.7.2 ตัวแปรที่ให้ตัวแปรแบ่งกลุ่มของความแปรปรวนหลายทาง	96
2.7.3 ตัวแปรที่เป็นตัวแปรร่วมของความแปรปรวนหลายทาง	97
2.7.4 ค่าสถิติของความแปรปรวนหลายทาง	98
2.7.5 รูปแบบการวิเคราะห์ความแปรปรวน	99
2.8 รายการสหสัมพันธ์	100
2.8.1 ตัวแปรของสหสัมพันธ์	101
2.8.2 รายละเอียดเพิ่มเติมในสัมพันธ์	102
2.9 รายการวิเคราะห์ถดถอย	104
2.9.1 ตัวแปรของการวิเคราะห์ถดถอย	105
2.9.2 วิธีการคัดเลือกตัวแปรในการวิเคราะห์ถดถอย	106

สารบัญ้งาน จอภาพ และตาราง (ต่อ)

จอภาพ		หน้า
2.10	การเขียนกราฟ	107
2.10.1	การกำหนดข้อความและสเกลในแนวนอน	108
2.10.2	การกำหนดข้อความและสเกลในแนวตั้ง	109
2.10.3	การกำหนดชื่อกราฟ	110
2.11	รายการทดสอบ Nonparametric	111
2.11.1	ชื่อตัวแปรของการทดสอบ Nonparametric	112
2.11.2	รายการทดสอบ Binomial	113
2.11.3	รายการทดสอบ Mcnemar	114
2.11.4	รายการทดสอบ Wilcoxon	115
2.11.5	รายการทดสอบ Cochran Q Test	116
2.11.6	รายการทดสอบ Frideman	117
2.11.7	รายการทดสอบ Kendall	118
2.11.8	รายการทดสอบ Median	119
2.11.9	รายการทดสอบ Mann-Whitney U	120
2.11.10	รายการทดสอบ Kolmogorov-Smirnov	121
2.11.11	รายการทดสอบ Kruskal-Wallis	122
3	รายการคำสั่งสถิติ	123
3.1	แสดงตัวแปรและค่า	124
3.1.1	แสดงตัวแปร	125
3.1.2	ขอบเขตข้อมูล	126
3.2	การเก็บข้อมูลและผลลัพธ์ลงเพิ่มข้อมูล	127
3.2.1	ชื่อเพิ่มข้อมูลใหม่	128
3.2.2	เก็บชื่อตัวแปร	128
3.3	เก็บข้อมูลแบบ ASCII	129
3.3.1	ชื่อเพิ่มข้อมูลใหม่ของการเก็บข้อมูลแบบ ASCII	130
3.3.2	รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่	131
3.3.3	รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่	132
3.3.4	เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่	133
3.4	รายการเกี่ยวกับเพิ่มข้อมูล	134
3.4.1	อ่านเพิ่มข้อมูลจากโปรแกรมอื่น	135

สารบัญม้งงาน จอภาพ และตาราง (ต่อ)

		หน้า
จอภาพ		
3.4.2	รายชื่อตัวแปรที่ไม่เก็บใ้เพิ่มข้อมูลใหม่	136
3.4.3	รายชื่อตัวแปรที่เก็บใ้เพิ่มข้อมูลใหม่	137
3.4.4	เปลี่ยนรายชื่อตัวแปรใ้เพิ่มข้อมูลใหม่	138
3.5	การเรียงลำดับข้อมูล	139
4	รายการอั้ตถประโชยน์	140
ตารางที่		
3.1	ค่าคงตัวกำหนดแบบกรอบหน้าต่าง	18
3.2	ค่าคงตัวกำหนดพารามิเตอร์ของกรอบหน้าต่าง	19
3.3	ค่าคงตัวกำหนดแบบข้อมูล	21
3.4	ค่าคงตัวกำหนดแบบตัวแปร	22
3.5	กระบวนความใ้ยูนิต Win	22
3.6	กระบวนความใ้ยูนิต Windo	23

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ด้วยเหตุที่การวิจัยเป็นเครื่องมือหนึ่ง ที่จะนำไปสู่การค้นคว้าเพื่อเรียนรู้ในสาขาวิชาต่างๆ การวิจัยมีบทบาททั้งในภาครัฐบาล ภาคเอกชน และผู้ที่อยู่ในวงการศึกษ การวิจัยส่วนใหญ่มักต้องการประมวลผลทางสถิติ รวมทั้งการแปลผลให้ได้ความหมายและความถูกต้อง รวดเร็ว การประมวลผลนี้สามารถกระทำได้โดยง่าย เมื่อใช้โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺ ซึ่งเป็นโปรแกรมสำเร็จรูปที่นิยมใช้กันมาก และใช้กับเครื่องไมโครคอมพิวเตอร์ IBM PC/XT 16 Bits ที่มีฮาร์ดดิสก์ (Hard Disk) แต่เนื่องจากการใช้โปรแกรมสำเร็จรูปทางสถิตินี้ซึ่งเป็นโปรแกรมลักษณะคำสั่ง (Command Driven) ผู้ใช้จำเป็นต้องศึกษาถึงวิธีการใช้ การเขียนคำสั่ง ป้อนข้อมูลว่ามีรูปแบบอย่างไรบ้าง เช่น จะต้องบอกตำแหน่งของข้อมูลว่าอยู่ที่ตำแหน่งใดและประเภทข้อมูลเป็นลักษณะใด รวมทั้งต้องมีความรู้สถิติด้วย

ดังนั้นเพื่อให้ผู้ใช้ได้ใช้งานได้สะดวกขึ้นโดยไม่จำเป็นต้องทราบถึงวิธีการเขียนคำสั่ง และรูปแบบการป้อนข้อมูลมากนัก ของโปรแกรมสำเร็จรูปนี้ การวิจัยครั้งนี้จึงได้เขียนโปรแกรมเป็นลักษณะรายการ (Menu) ขึ้นซึ่งมีรายการเป็นภาษาไทยบนเครื่องไมโครคอมพิวเตอร์ IBM PC/XT ขนาด 16 Bits พร้อมฮาร์ดดิสก์ ชื่อว่า SPSMENU เพียงแต่ผู้ใช้กดปุ่มบนแป้นพิมพ์ ใส่ข้อมูลตามที่รายการปรากฏบนหน้าจอว่า ให้ใส่อะไรบ้าง แล้วโปรแกรมดังกล่าวจะทำการสร้างคำสั่งต่างๆ ขึ้น ให้สอดคล้องกับโปรแกรมสำเร็จรูปนี้ หลังจากนั้นผู้ใช้จะต้องนำคำสั่งต่างๆ ที่ถูกสร้างขึ้นนี้ ไปประมวลผลบนเครื่องไมโครคอมพิวเตอร์ที่มีโปรแกรมสำเร็จรูป SPSS/PC และ SPSS/PC⁺ ติดตั้งอยู่ด้วย

1.2 วัตถุประสงค์

1. เพื่อสร้างโปรแกรมเป็นลักษณะรายการ (Menu) ภาษาไทยจากโปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺
2. เพื่อนำโปรแกรมสำเร็จรูปที่ได้ไปใช้ในการวิเคราะห์ข้อมูล

1.3 ขอบเขตการศึกษา

สร้างโปรแกรมลักษณะรายการ (Menu) เป็น ภาษาไทย แบบ Pull Down จากโปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺ บนเครื่องไมโครคอมพิวเตอร์ IBM PC/XT 16 Bits พร้อมฮาร์ดดิสก์

1.4 อุปกรณ์ ภาษาและโปรแกรมที่ใช้

1. เครื่องไมโครคอมพิวเตอร์ IBM PC/XT Proturbo 88 ขนาด 16 Bits ประกอบด้วยไมโครโปรเซสเซอร์ 8088
2. หน่วยความจำ (RAM) ขนาด 640 Kbytes
3. ฮาร์ดดิสก์ มีความจุอย่างน้อย 30 Megabytes
4. ดิสก์ไดรฟ์ (Disk Drive) อย่างน้อย 1 ตัว
5. จอโมโนโครม (Monochrome)
6. เครื่องพิมพ์ (Printer) LQ 1050
7. ระบบปฏิบัติการที่ใช้ (Operating System) เป็น MS DOS รุ่น 3.3
8. ภาษาที่ใช้คือ Turbo Pascal รุ่น 5.0
9. Thai Driver เป็น Thaitip
10. โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺

1.5 นิยามศัพท์

Command Driven หมายถึงการดึงคำสั่งต่างๆที่มีอยู่มาเขียนเป็นชุดคำสั่ง

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โปรแกรมลักษณะรายการภาษาไทยจากโปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺
2. เพื่ออำนวยความสะดวก แก่ผู้ใช้ในการวิเคราะห์ข้อมูลโดยใช้โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺

บทที่ 2

โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺

โปรแกรมสำเร็จรูปทางสถิตินี้มี 2 รุ่นคือ SPSS/PC และ SPSS/PC⁺ โดยที่ SPSS/PC⁺ เป็นโปรแกรมที่พัฒนาขึ้นจาก SPSS/PC เพื่อให้ใช้งานได้มากขึ้น สำหรับโปรแกรมนี้ ผลิตโดยบริษัท SPSS INC. ประเทศสหรัฐอเมริกา และโปรแกรมนี้บรรจุอยู่ในแผ่นดิสก์เก็ต จำนวน 11 แผ่น และ 20 แผ่น ตามลำดับ ซึ่งต้องใช้กับเครื่องไมโครคอมพิวเตอร์ที่มีฮาร์ดดิสก์

สำหรับความสามารถในการจัดการกับข้อมูลของโปรแกรมสำเร็จรูปทางสถิตินี้ แบ่งเป็น 2 ประเภทใหญ่ๆ คือ

2.1 ความสามารถทั่วไป

2.2 ความสามารถทางสถิติ

2.1 ความสามารถทั่วไป

1. การเปลี่ยนรูปข้อมูล (Data Transformation) โดยนำข้อมูลเดิมที่มีอยู่มาคำนวณในลักษณะต่าง ๆ กัน

2. การคัดเลือกข้อมูล (Data Selection) เป็นการคัดเลือกข้อมูลที่มีอยู่โดยมีเงื่อนไขตามที่ผู้ใช้ต้องการ

3. การนำผลลัพธ์ที่ได้ ไปทำงานต่อเนื่องกัน

4. การให้น้ำหนักความสำคัญ (Weight) แก่ชุดข้อมูล

5. การเรียงลำดับข้อมูล

6. การบันทึกข้อมูลในลักษณะต่าง ๆ เป็นแบบ Binary หรือ แบบ ASCII

2.2 ความสามารถทางสถิติ

โปรแกรมสำเร็จรูปนี้ สามารถประมวลผลข้อมูล ได้ตั้งแต่การคำนวณค่าสถิติเบื้องต้น (Descriptive Statistics) จนถึงสถิติหลายตัวแปร (Multivariate Statistics) และ สถิติอนพารามิเตอร์ (Nonparametrics Statistics) เช่น

1. การคำนวณค่าสถิติเบื้องต้น (Descriptive Statistics) เช่น ค่าเฉลี่ย (Mean) มัชฌิม (Median) ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) ฯลฯ

2. การแจกแจงความถี่ สามารถแจกแจง ความถี่แบบทางเดียว (Frequency Distribution) แบบหลายทาง (Crosstabulation Table) พร้อมคำนวณค่าสถิติต่างๆ เช่น Chi-Square

3. การหาความสัมพันธ์ของตัวแปร เช่น ค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient)
4. การวิเคราะห์ถดถอย (Regression Analysis)
5. การเปรียบเทียบค่าเฉลี่ย วิเคราะห์ข้อมูลเกี่ยวกับค่าเฉลี่ยของประชากร ในลักษณะต่าง ๆ เช่น T-Test
6. การทดสอบนอนพารามิเตอร์ (Nonparametric Test) เช่น U-Test

บทที่ 3

วางแผนระบบงาน ผังการทำงาน และโปรแกรมที่พัฒนา

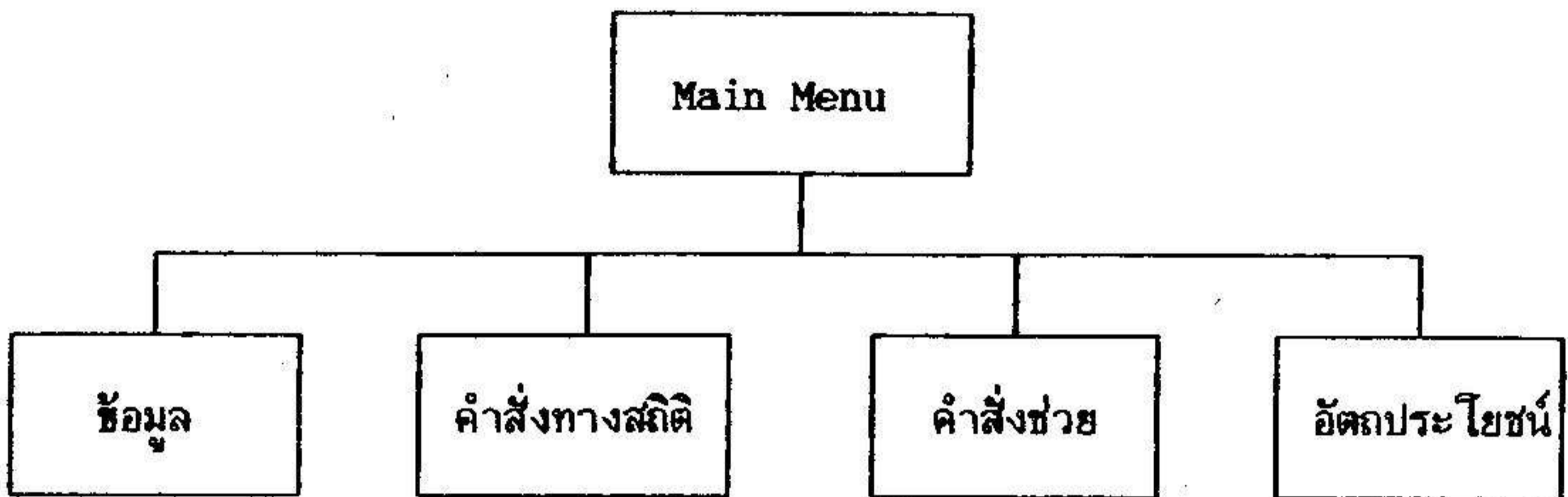
ในการทำวิจัยครั้งนี้ได้วางแผนระบบงาน ผังการทำงาน และ โปรแกรมที่พัฒนา โดยจะกล่าวถึงในหัวข้อที่ 3.1 และ 3.2 ตามลำดับ

3.1 ระบบงาน ได้วางแผนระบบงานเป็นรายการหลัก (Main Menu) (ผังงาน ก) และแบ่งย่อยการทำงาน เป็น 4 ขั้นตอนคือ

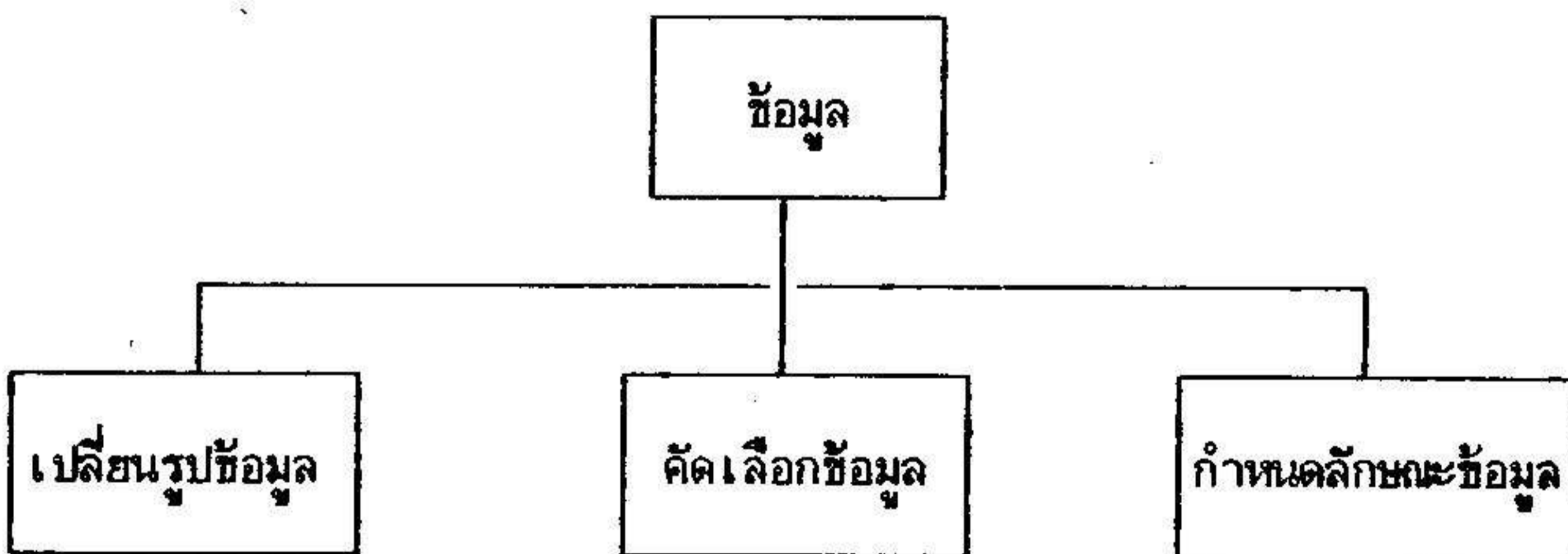
1. ข้อมูล (ผังงาน 1) ส่วนนี้จะแยกย่อยการทำงานเป็น
 - เปลี่ยนรูปข้อมูล
 - คัดเลือกข้อมูล
 - กำหนดลักษณะข้อมูล
2. คำสั่งทางสถิติ (ผังงาน 2) แยกย่อยการทำงานเป็น
 - ความถี่ (ผังงาน 2.1)
 - สถิติพรรณนา (ผังงาน 2.2)
 - สถิติเบื้องต้นที่แบ่งกลุ่ม (ผังงาน 2.3)
 - ตารางการแจกแจง (ผังงาน 2.4)
 - ทดสอบ T-Test (ผังงาน 2.5)
 - ความแปรปรวนทางเดียว (ผังงาน 2.6)
 - ความแปรปรวนหลายทาง (ผังงาน 2.7)
 - สหสัมพันธ์ (ผังงาน 2.8)
 - วิเคราะห์ถดถอย (ผังงาน 2.9)
 - เขียนกราฟ (ผังงาน 2.10)
 - ทดสอบนอนพารามิเตอร์ (ผังงาน 2.11)
3. คำสั่งช่วย (ผังงาน 3) แยกย่อยการทำงานเป็น
 - แสดงตัวแปรและค่า
 - เก็บข้อมูลและผลลัพธ์
 - เก็บข้อมูลแบบ ASCII (Export)
 - อ่านเพิ่มข้อมูลจากโปรแกรมอื่นๆ (Import)
 - เรียงลำดับข้อมูล
 - เครื่องพิมพ์

4. อັตถประโยชนั (ผังงาน 4) แยกย่อยการทำงานเป็น

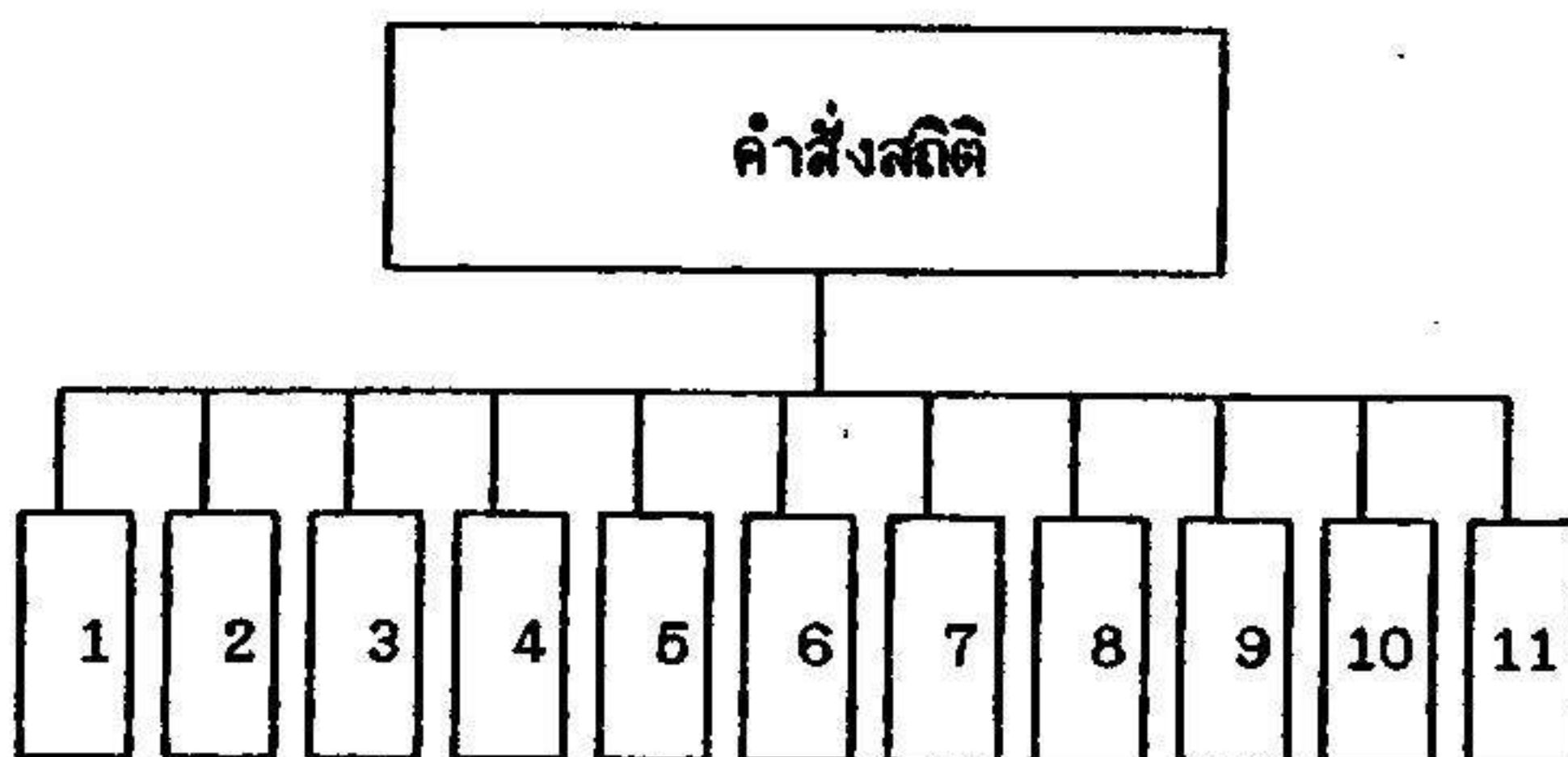
- แสดงชื่อแฟ้มข้อมูล (Dir)
- ออกจากการทำงาน (Exit)



ผังงาน ก รายการหลัก

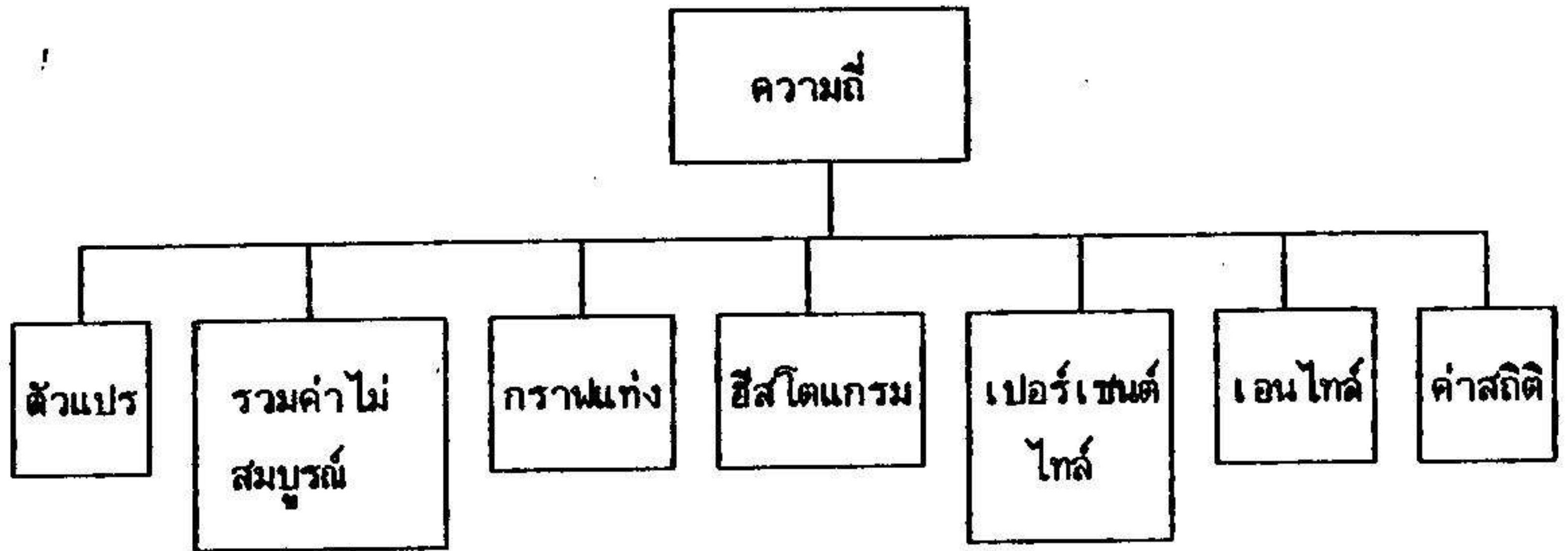


ผังงาน 1 ข้อมูล

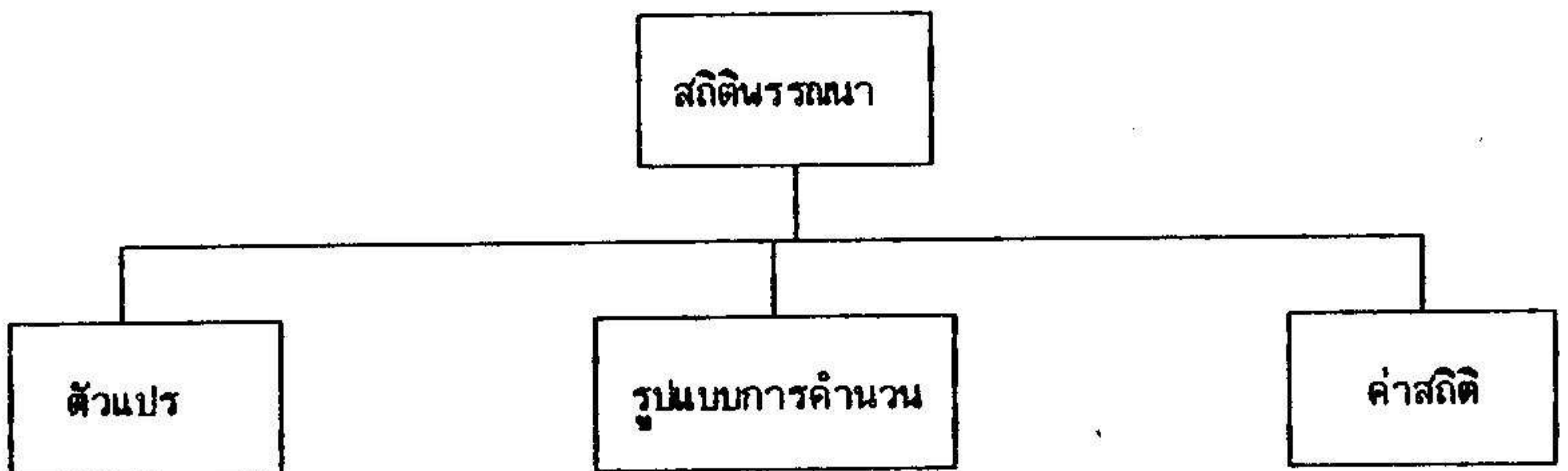


- | | | | | | |
|---|---|----------------------------|----|---|---------------------|
| 1 | : | ความถี่ | 6 | : | ความแปรปรวนทางเดียว |
| 2 | : | สถิติพรรณนา | 7 | : | ความแปรปรวนหลายทาง |
| 3 | : | สถิติเบื้องต้นที่แบ่งกลุ่ม | 8 | : | สหสัมพันธ์ |
| 4 | : | ตารางการแจกแจง | 9 | : | วิเคราะห์ถดถอย |
| 5 | : | ทดสอบ T-Test | 10 | : | เขียนกราฟ |
| | | | 11 | : | ทดสอบ NONPARAMETRIC |

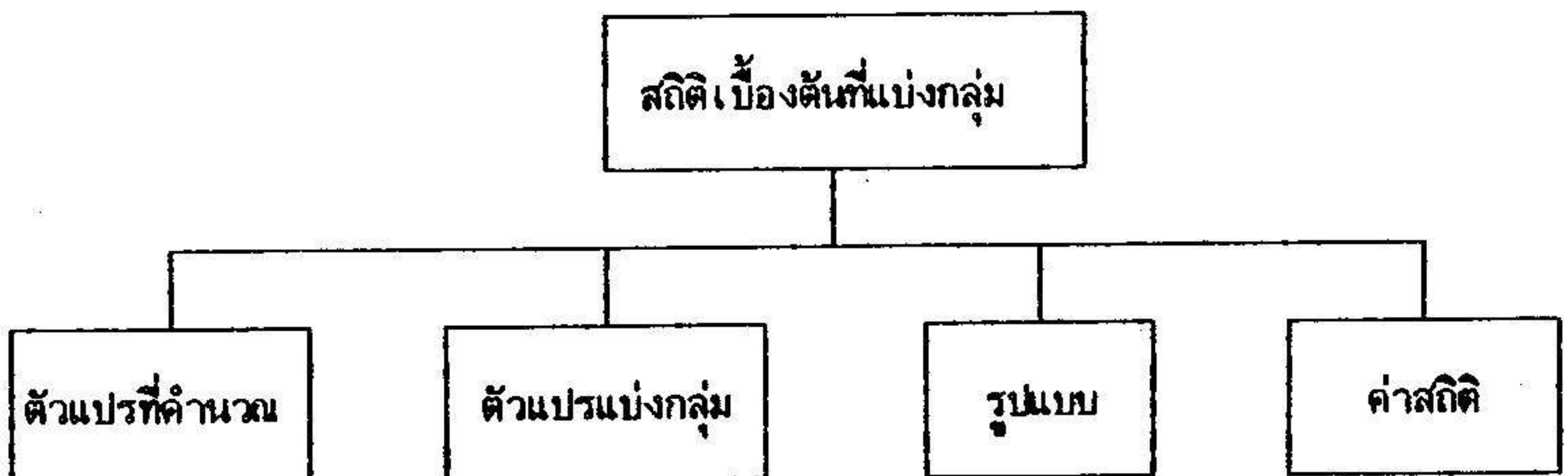
สำหรับผังคำสั่งทางสถิติ ในแต่ละคำสั่งทางสถิติสามารถแยกย่อยได้อีก ดังนี้



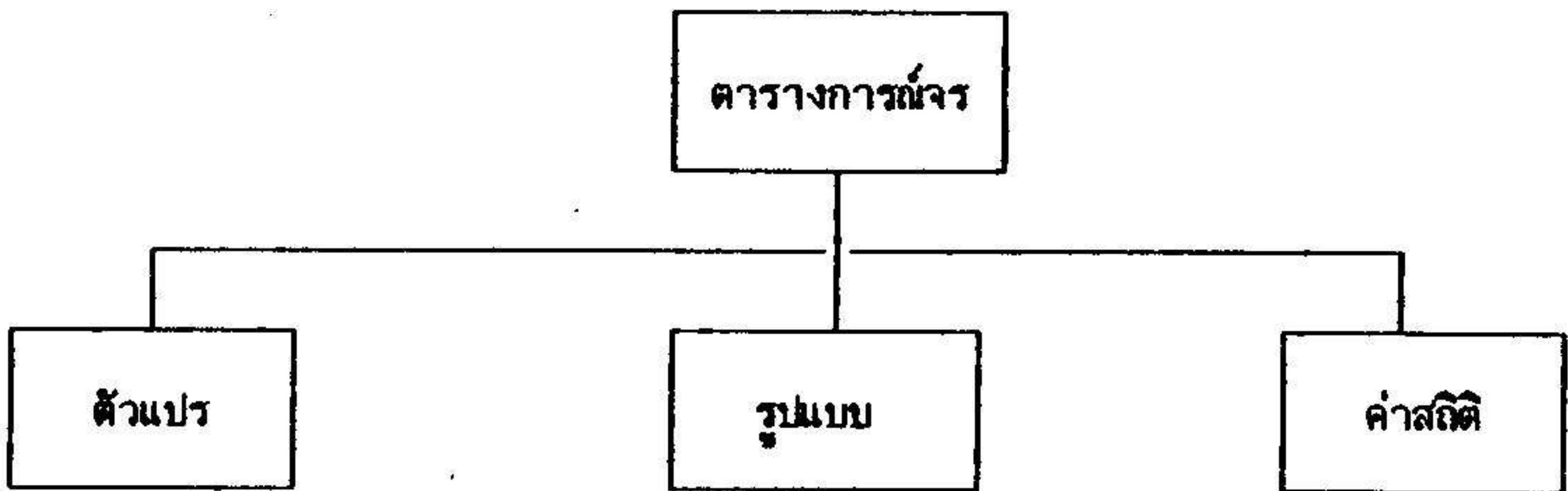
ผังงาน 2.1 ความถี่



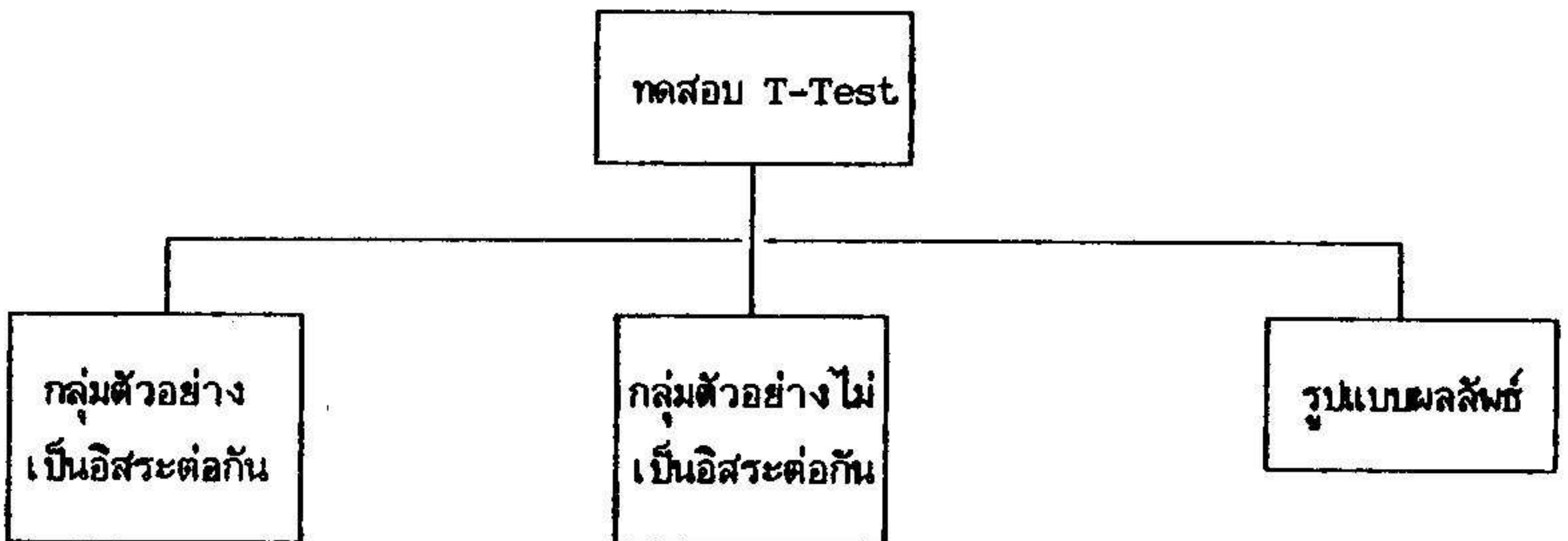
ผังงาน 2.2 สถิติพรรณนา



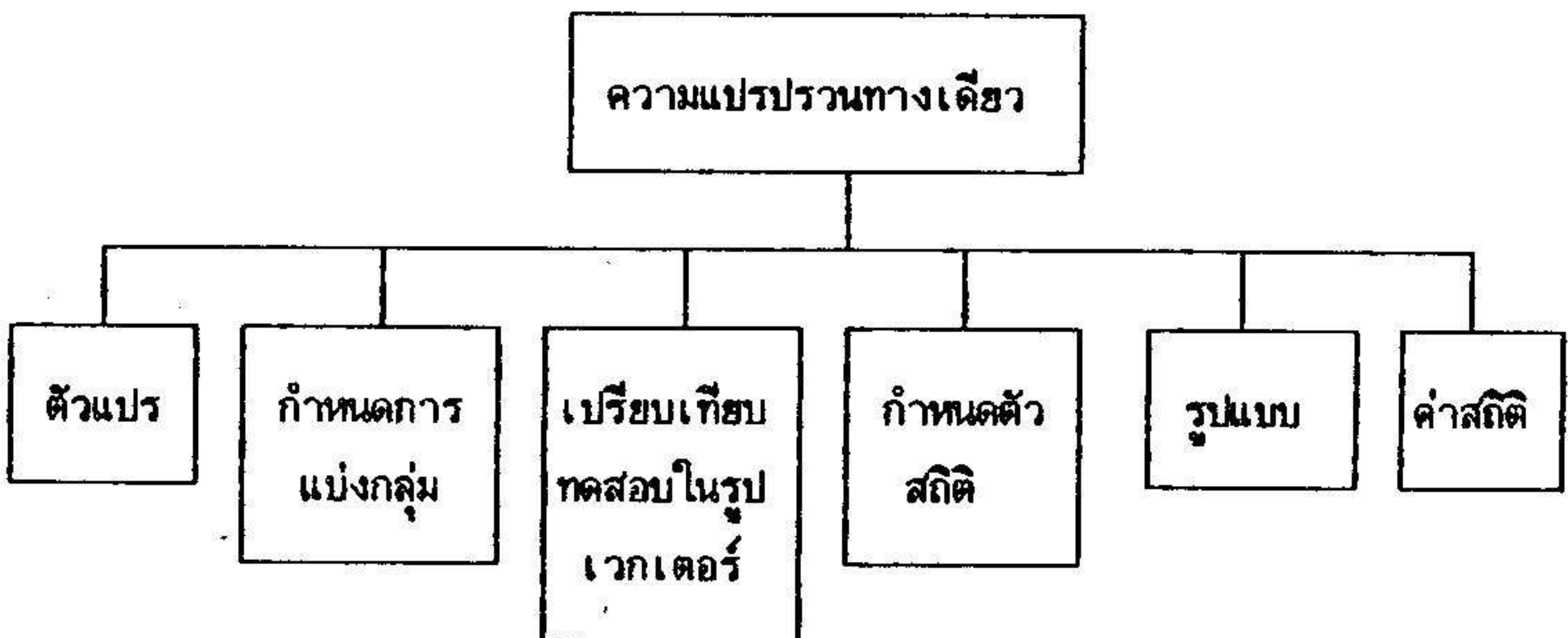
ผังงาน 2.3 สถิติเบื้องต้นที่แบ่งกลุ่ม



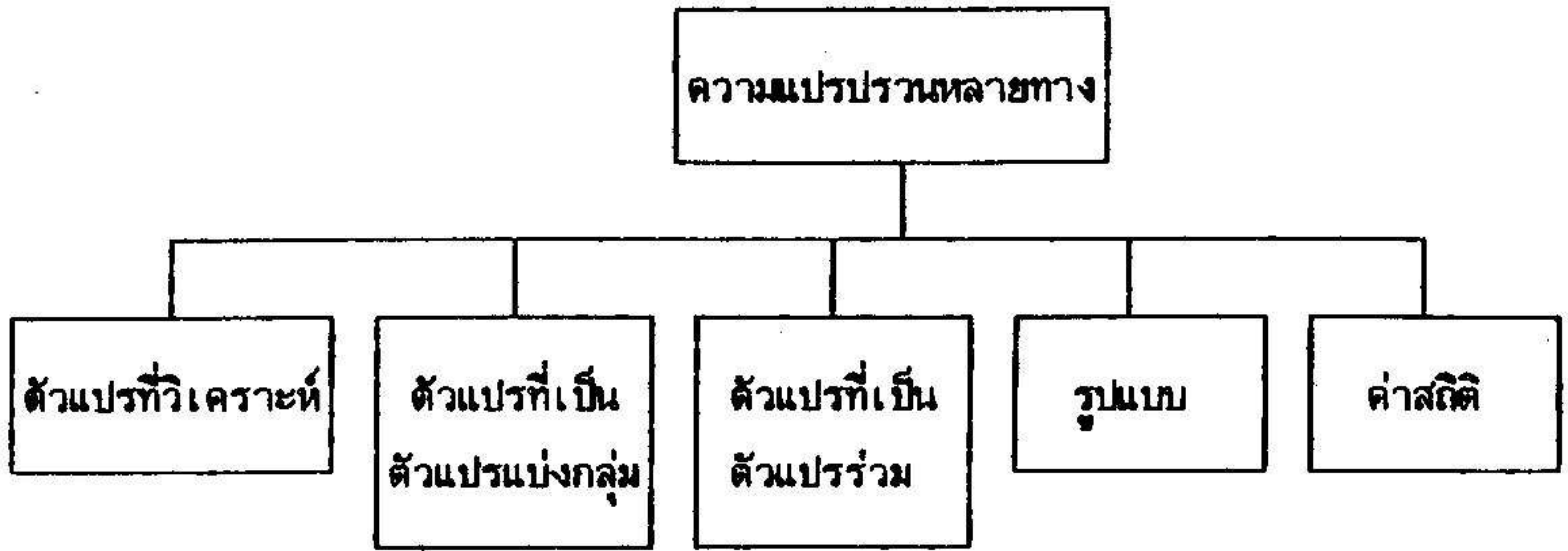
ผังงาน 2.4 ตารางการสำรวจ



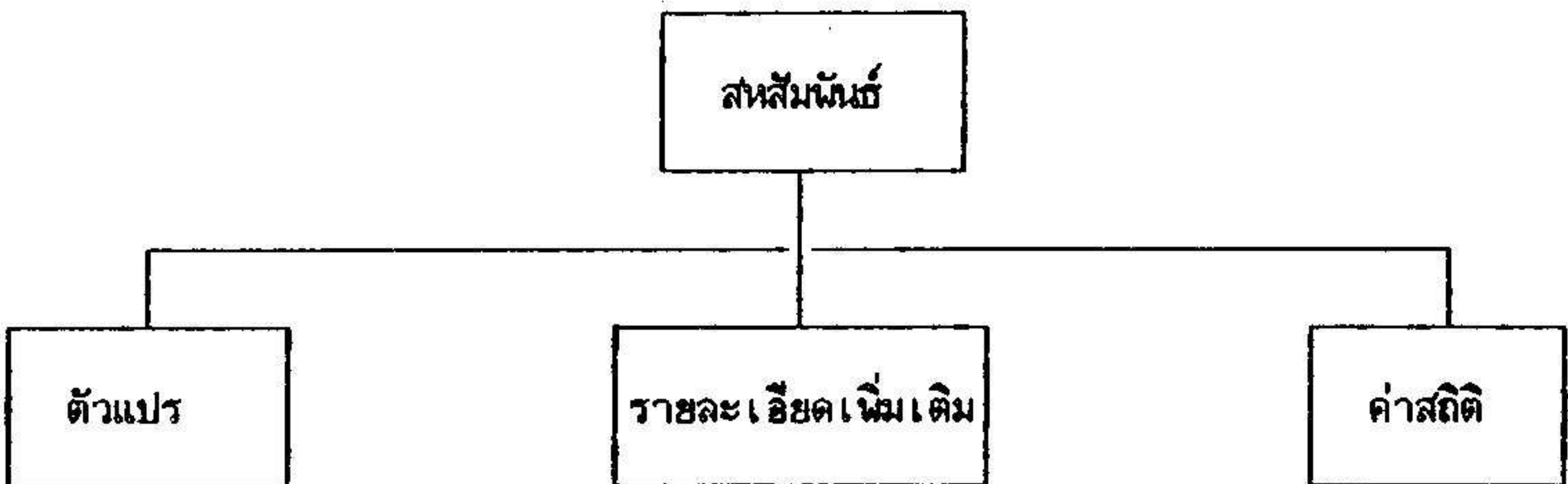
ผังงาน 2.5 ทดสอบ T-Test



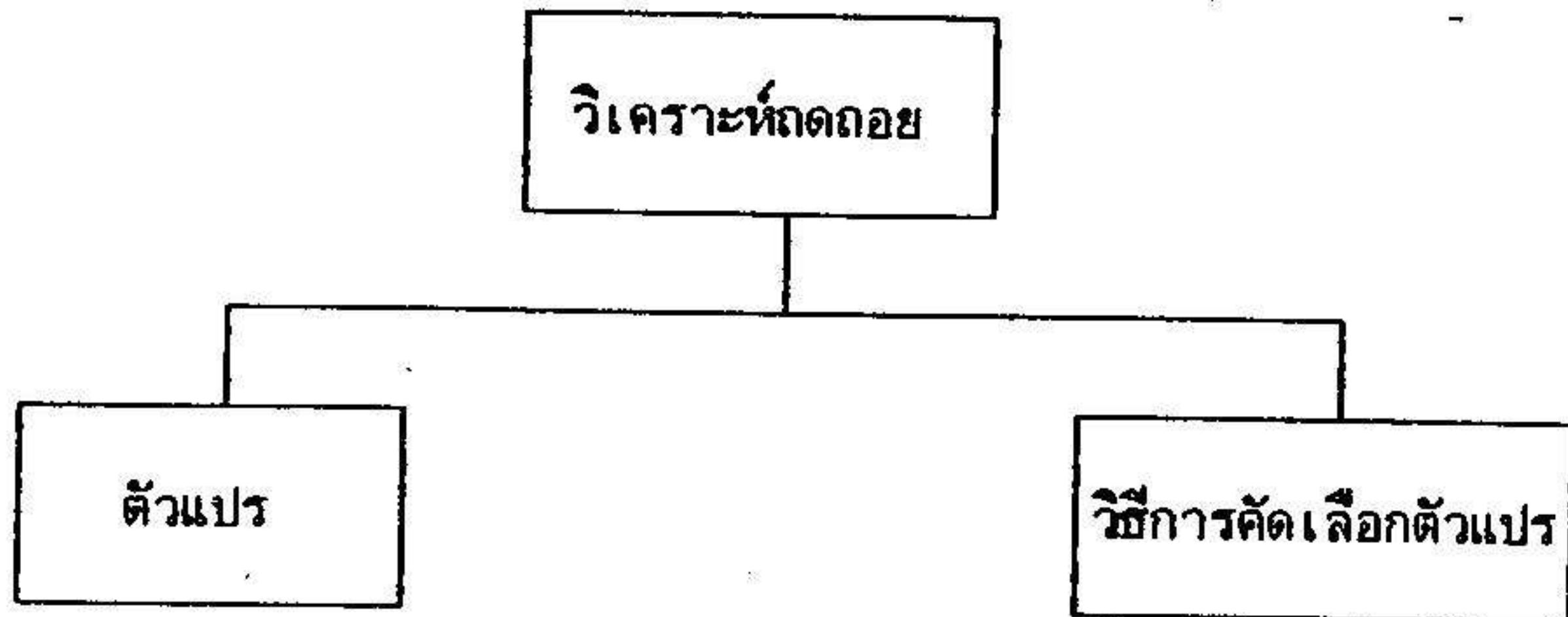
ผังงาน 2.8 ความแปรปรวนทางเดียว



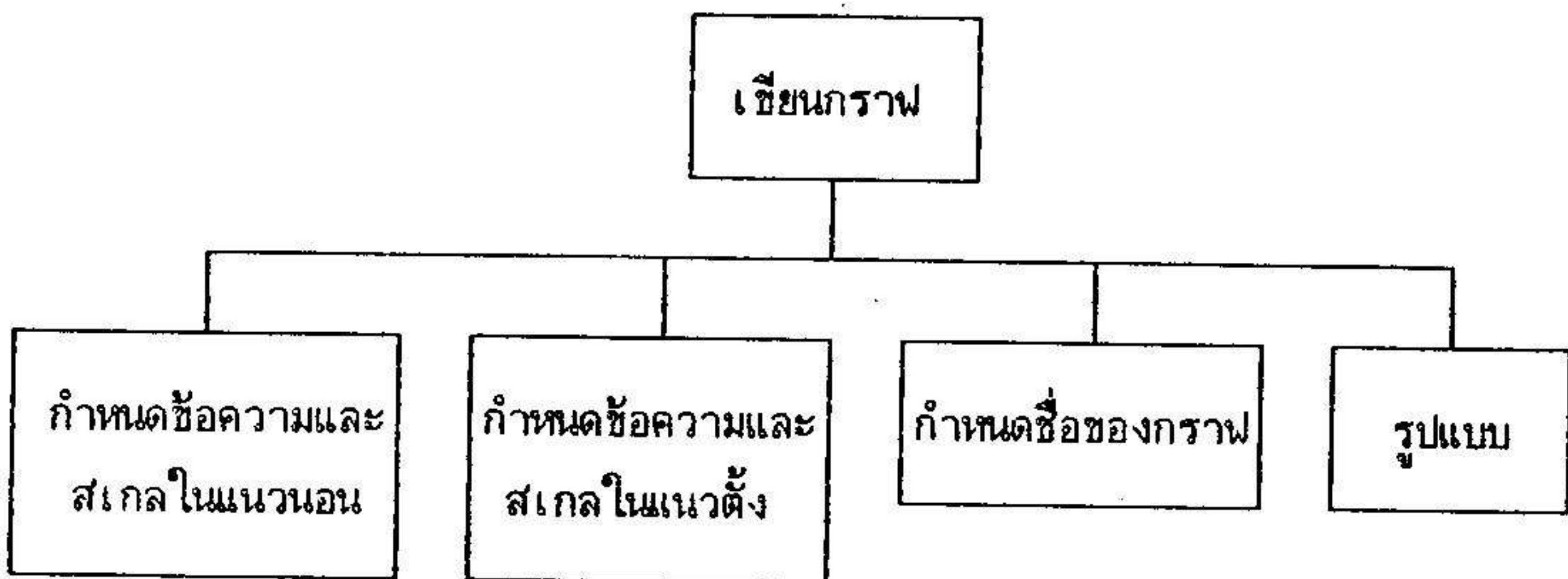
ผังงาน 2.6 ความแปรปรวนหลายทาง



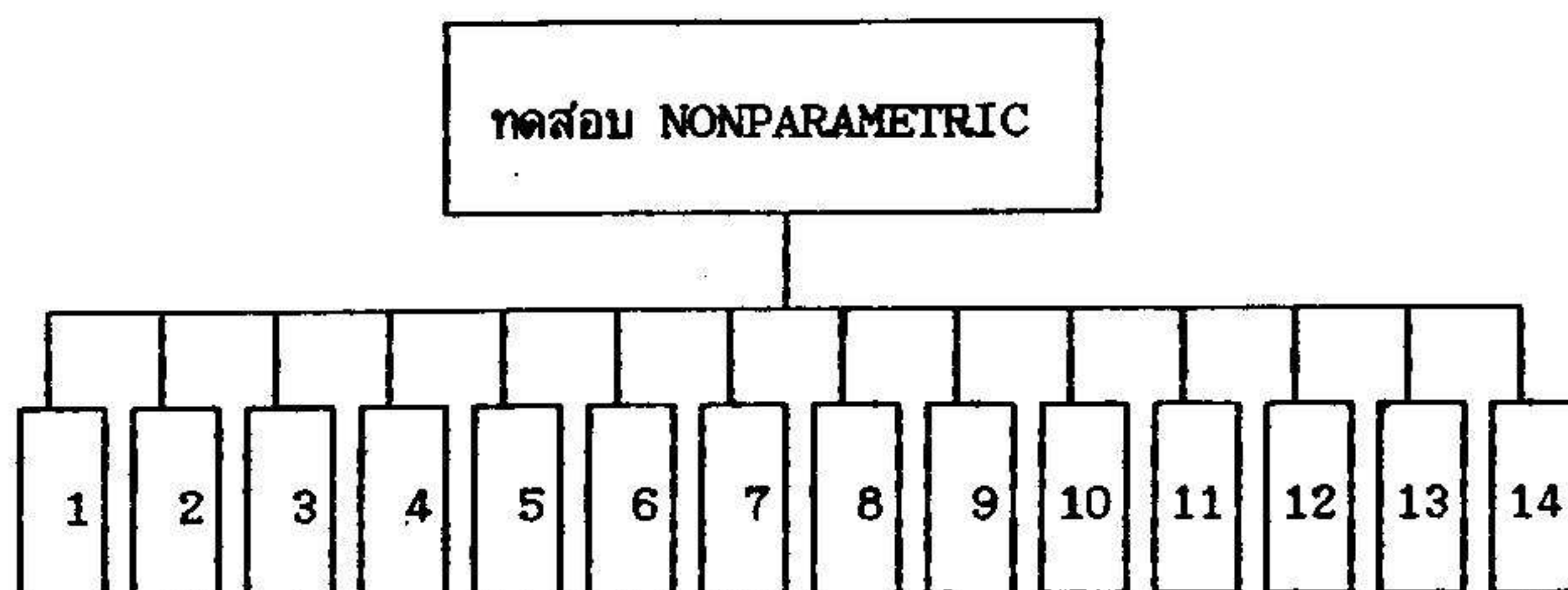
ผังงาน 2.8 สหสัมพันธ์



ผังงาน 2.9 วิเคราะห์ถดถอย

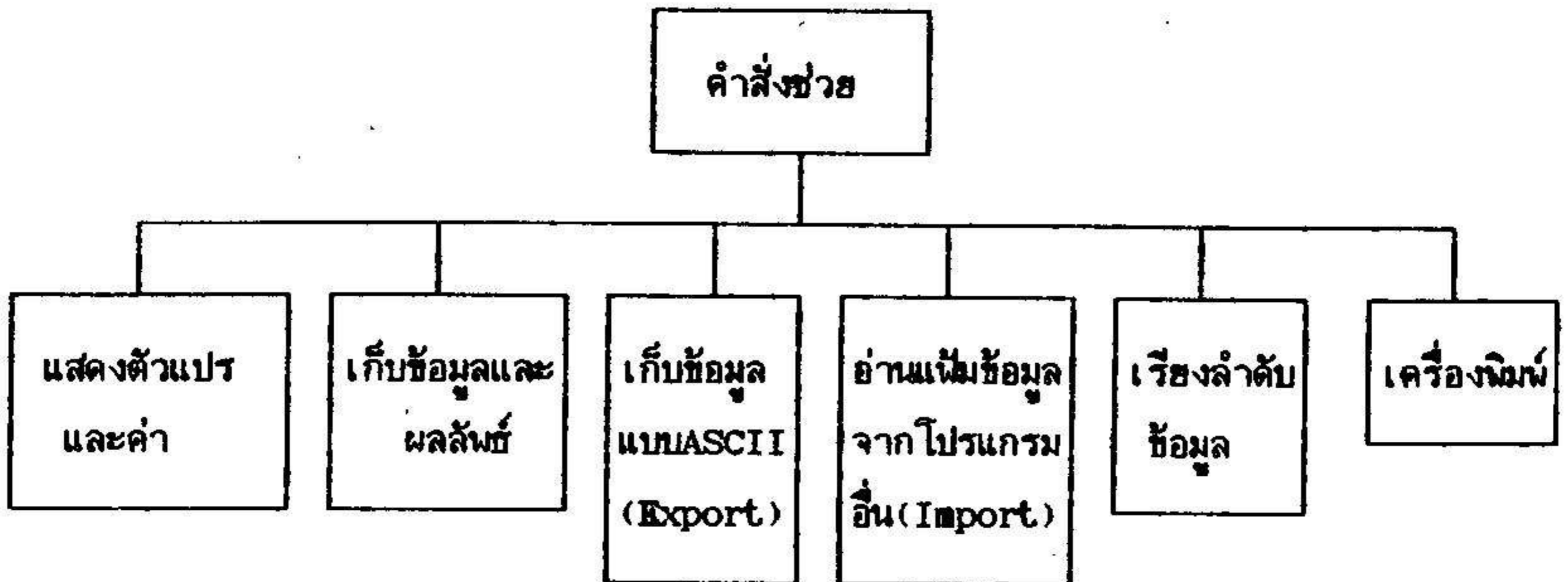


ผังงาน 2.10 เขียนกราฟ

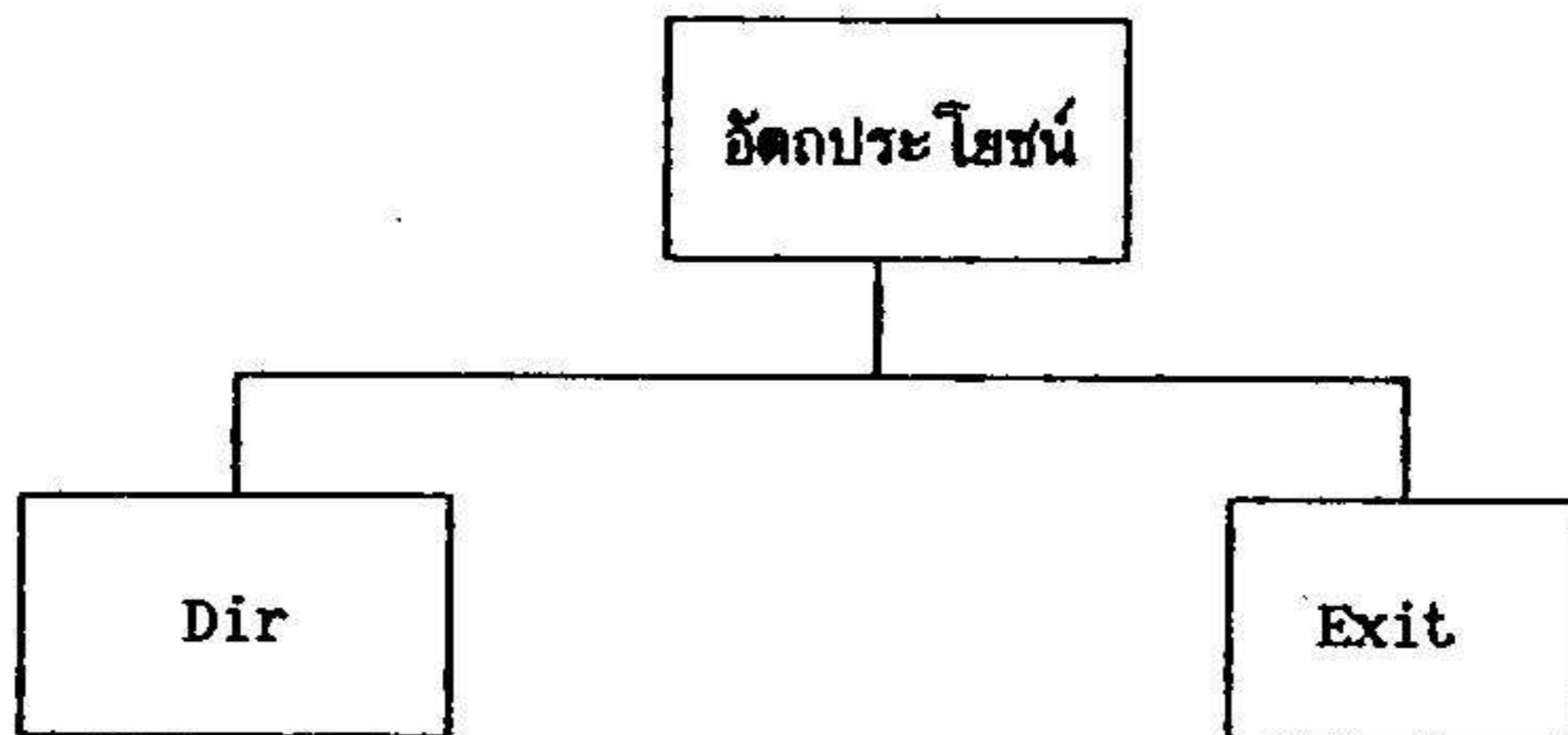


1	:	ตัวแปรที่ทดสอบ	8	:	MEDIAN
2	:	BINOMIAL	9	:	MANN-WHITNEY U
3	:	MCNEMAR	10	:	KOMOGOROV SMIROV
4	:	WILCOXON	11	:	WALD-WOIFWITZ
5	:	COCHRAN	12	:	KRUSKAL-WALLIS
6	:	FRIEDMAN	13	:	รูปแบบ
7	:	KENDALL	14	:	ค่าสถิติ

ผังงาน 2.11 ทดสอบนอนพารามิเตอร์



ผังงาน 3 คำสั่งช่วย



ผังงาน 4 อັดถประโชษน์

3.2 โปรแกรมที่พัฒนา แบ่งเป็นส่วนต่างๆ ดังนี้ คือ

3.2.1 โปรแกรมหลัก (Main Program) ชื่อเพิ่มข้อมูลคือ spsmenu.pas เป็นโปรแกรมที่เรียกใช้โปรแกรมย่อยทั้งหมด และ สร้างรายการหลักทั้ง 4 คือ ข้อมูล , คำสั่งทางสถิติ, คำสั่งช่วย อັดถประโชษน์ และ เรียกใช้คำสั่ง Help จากเพิ่มข้อมูล ที่มีนามสกุล Txt

3.2.2 ยูนิตต่างๆ ที่สัมพันธ์กับโปรแกรมหลัก ได้แก่

- 1 ยูนิต screen
- 2 ยูนิต keyboard
- 3 ยูนิต win

- 4 ยูนิต windo
- 5 ยูนิต ldir
- 6 ยูนิต swin0 ,ssub
- 7 ยูนิต display
- 8 ยูนิต fi
- 9 ยูนิต trans ,proc และ uti

1. ยูนิต Screen

Screen เป็นยูนิตทำหน้าที่ควบคุมการแสดงผล ได้แก่ การรับแอตทริบิวต์ (Attribute) ของ ตัวอักษร และการปรับเคอร์เซอร์

สำหรับยูนิตนี้ได้กำหนดค่าคงตัว ตัวแปร ระยะเวลา ซึ่งจะกล่าวถึงในหัวข้อที่ 1.1 , 1.2 และ 1.3 ตามลำดับ

1.1 ค่าคงตัว ค่าคงตัวในยูนิตนี้ประกอบด้วย ค่าคงตัวกำหนดแอตทริบิวต์ และ ค่าคงตัวกำหนดเซกเมนต์ของวีดีโอแรม ค่าคงตัวกำหนดแอตทริบิวต์มีดังนี้

const	NoDisplay	=	\$00;	{ ไม่แสดงตัวอักษร }
	LowDisplay	=	\$07;	{ ตัวอักษรจาง }
	HighDisplay	=	\$0F;	{ ตัวอักษรเข้มก }
	UnderLineLow	=	\$01;	{ ตัวจางขีดเส้นใต้ }
	UnderLineHigh	=	\$09;	{ ตัวเข้มขีดเส้นใต้ }
	ReverseLow	=	\$70;	{ ตัวจางรีเวอร์ส }
	ReverseHigh	=	\$70;	{ ตัวเข้มรีเวอร์ส }
	BlinkLow	=	\$87;	{ ตัวจางกระพริบ }
	BlinkHigh	=	\$8F;	{ ตัวเข้มกระพริบ }
	UndBlinkLow	=	\$81;	{ ตัวจางขีดเส้นใต้กระพริบ }
	UndBlinkhigh	=	\$89;	{ ตัวเข้มขีดเส้นใต้กระพริบ }
	RevBlinkLow	=	\$F0;	{ ตัวจางรีเวอร์สกระพริบ }
	RevBlinkHigh	=	\$F8;	{ ตัวเข้มรีเวอร์สกระพริบ }

ค่าคงตัวเหล่านี้ ใช้เป็นพารามิเตอร์สำหรับปรับแอตทริบิวต์ตัวอักษร ด้วยระยะเวลา SetAttr (HighDisplay)

ค่าคงตัวกำหนดเซกเมนต์มีดังนี้

const	ColorSeg	=	\$B800;
	MonoSeg	=	\$B000;

เมื่อ ColorSeg และ MonoSeg เป็นค่าคงตัวซึ่งเก็บค่าวิดีโอแรมเซกเมนต์ของจอสีและจอโมโนโครม ตามลำดับ

1.2 ตัวแปร ในหน่วยนี้มี 4 ตัวแปร คือ

```
var VideoSeg      : word;
    CrtType       : byte Absolute $0040:$0049;
    CursorMode    : word Absolute $0040:$0060;
    Vport         : word Absolute $0040:$0063;
```

VideoSeg เป็นตัวแปรที่เก็บค่าวิดีโอเซกเมนต์ของจอภาพที่ใช้งานในขณะนั้น

CrtType เป็นตัวแปรแอปโซลูตที่บ่งบอกชนิดจอภาพตามรหัสที่ได้จากไบออสดังนี้

0	จอภาพแบบ	40x25	ขาวดำ
1	จอภาพแบบ	40x25	สี
2	จอภาพแบบ	80x25	ขาวดำ
3	จอภาพแบบ	80x25	สี
4	จอภาพแบบ	80x25	โมโนโครม

CursorMode จะเก็บลักษณะรูปร่างของเคอร์เซอร์ และตัวแปร Vport เก็บค่าวิดีโอพอร์ตของจอภาพที่ใช้งานขณะนั้น

1.3 กระทบความ ในหน่วย Screen มี 4 กระทบความคือ

SetAttr	ปรับแอตทริบิวต์การแสดงผลตัวอักษร
SetCursor	ปรับขนาดเคอร์เซอร์
Cursoron	แสดงตัวเคอร์เซอร์
Cursoroff	ไม่แสดงตัวเคอร์เซอร์

2. หน่วย Keyboard

Keyboard เป็นหน่วยที่ประกอบด้วยค่าคงตัวประจำปุ่มฟังก์ชัน ตัวแปรเก็บรหัสการกดปุ่ม และกระทบความการอ่านเป็นนิมฟ์

สำหรับหน่วยนี้ได้กำหนดค่าตัวแปร ค่าคงตัวและกระทบความ ซึ่งจะกล่าวถึงในหัวข้อที่ 2.1 และ 2.2 ตามลำดับ

2.1 ตัวแปรและค่าคงตัว ตัวแปรในหน่วยนี้มีการกำหนดไว้ดังนี้

```
var Key          : char;
    FuncKey      : boolean;
```

โดยที่

ตัวแปร Key ใช้ในการเก็บรหัสจากการกดปุ่มบนแป้นพิมพ์ ส่วนตัวแปร FuncKey เป็นแฟล็กที่บ่งบอกว่ารหัสในตัวแปร Key เป็นรหัสของตัวอักษรหรือว่ารหัสของปุ่มฟังก์ชันคือ ถ้า FuncKey มีค่าเป็น false แสดงว่าผู้ใช้กดปุ่มตัวอักษร และ ค่าในตัวแปร Key จะเก็บรหัสแอสกี (ASCII) ของตัวอักษรนั้น แต่ถ้า FuncKey มีค่า true แสดงว่าผู้ใช้กดปุ่มฟังก์ชัน ค่าในตัวแปร Key ก็จะเก็บรหัสสแกนโค้ดของปุ่มฟังก์ชันนั้น ส่วนค่าคงตัวภายในหน่วยประกอบด้วยรหัสสแกนโค้ดดังนี้

```
const ReturnJKey = #13;   EscJKey = #27;
      ShiftJTab = #15;
      F1JKey = #59;      F10JKey = #68;
      HomeJKey = #71;   UpJKey = #72;
      PgUpJKey = #73;  LtJKey = #75;
      RtJKey = #77;    EndJKey = #79;
      DnJKey = #80;    PgDnJKey = #81;
      InsJKey = #82;   DelJKey = #83;
```

2.2 กระบวนการภายในหน่วย มีกระบวนการอ่านแป้นพิมพ์รวม

6 กระบวนการ ดังนี้

ReadFuncKey	อ่านการกดปุ่มบนแป้นพิมพ์หนึ่งตัวอักษร
ReadChar	อ่านสตริงโดยรับทีละตัวอักษร
ReadChar1	อ่านสตริงและตัวเลขจำนวนเต็ม
ReadInt	อ่านตัวเลขจำนวนเต็ม
ReadReal	อ่านตัวเลขจำนวนจริง
Checkword	นับคำเพื่อจะขึ้นต้นบรรทัดใหม่

3. หน่วย win

หน่วย win เป็นหน่วยของการใช้ระบบช่องหน้าต่าง ประกอบด้วยค่าคงที่แบบข้อมูล ตัวแปร กระบวนการในการสร้างช่องหน้าต่าง และพารามิเตอร์กำหนดลักษณะช่องหน้าต่าง นอกจากนี้ยังทำหน้าที่ควบคุมการแสดงผล ได้แก่ การปรับแอดตริบิวต์ของตัวอักษร

สำหรับหน่วยนี้ได้กำหนด ค่าคงตัวแบบกรอบหน้าต่าง ข้อมูล ตัวแปร และกระบวนการ ซึ่งจะกล่าวถึงใน หัวข้อ 3.1 , 3.2 , 3.3 และ 3.4 ตามลำดับ

3.1 ค่าคงตัวกำหนดแบบกรอบหน้าต่าง ค่าคงตัวนี้เป็นชื่อของแบบกรอบ
หน้าต่างซึ่งมี 2 แบบดังนี้

```
const Single = 1;
      Double = 2;
```

ตารางที่ 3.1 ค่าคงตัวกำหนดแบบกรอบหน้าต่าง

ชื่อค่าคงตัว	การใช้งาน
Single	เส้นกรอบหน้าต่างแบบเส้นเดี่ยว
Double	เส้นกรอบหน้าต่างแบบเส้นคู่

และค่าคงตัวนี้แบ่งเป็น

ก. ค่าคงตัวกำหนดพารามิเตอร์ของกรอบหน้าต่าง

ข. ค่าคงตัวกำหนดแอตทริบิวต์ และ ค่าคงตัวกำหนดเชกเมนต์ของวีดีโอแรม

ก. ค่าคงตัวกำหนดพารามิเตอร์ของกรอบหน้าต่าง ค่าคงตัวในกลุ่มนี้เป็นค่าคงตัวแบบ
ตัวแปร ซึ่งเก็บลักษณะของหน้าต่างไว้ดังนี้

```
const BoxStyle      : byte = Single;
      AttrofBox     : byte = HighDisplay;
      AttrofWindow  : byte = HighDisplay;
      AttrofHeader  : byte = highDisplay;
      AttrofChar    : byte = HighDisplay;
      HeaderofWinDow : string = "";
      TypeofBox     : array[1..2,1..8] of char =
      ((#196,#179,#179,#196,#218,#191,#192,#217),
      (#205,#186,#186,#205,#201,#187,#200,#188));
```

ตารางที่ 3.2 ค่าคงตัวกำหนดพารามิเตอร์ของกรอบหน้าต่าง

ชื่อค่าคงตัว	การใช้งาน
Boxstyle	
AttrOfWindow	แอตทริบิวต์ของพื้นที่หน้าต่าง ค่าปกติเป็น HighDisplay
AttrOfBox	แอตทริบิวต์ของเส้นกรอบหน้าต่าง ค่าปกติเป็น HighDisplay
AttrOfHeader	แอตทริบิวต์ของเฮดเดอร์ ค่าปกติเป็น HighDisplay
AttrOfChar	แอตทริบิวต์ของตัวอักษรในจอหน้าต่าง ค่าปกติ เป็น HighDisplay
HeaderOfWindow	สตริงของเฮดเดอร์ ค่าปกติเป็นสตริงมีความยาวศูนย์
TypeOfBox	อะเรย์ของเส้นกรอบหน้าต่าง

ข. ค่าคงตัวกำหนดแอตทริบิวต์ และ ค่าคงตัวกำหนดเชกเมนต์ของวิดีโอแรม
ค่าคงตัวกำหนดแอตทริบิวต์มีดังนี้

const	NoDisplay	=	\$00;	{ ไม่แสดงตัวอักษร }
	LowDisplay	=	\$07;	{ ตัวอักษรจาง }
	HighDisplay	=	\$0F;	{ ตัวอักษรเข้ม }
	UnderLineLow	=	\$01;	{ ตัวจางขีดเส้นใต้ }
	UnderLineHigh	=	\$09;	{ ตัวเข้มขีดเส้นใต้ }
	ReverseLow	=	\$70;	{ ตัวจางรีเวอร์ส }
	ReverseHigh	=	\$70;	{ ตัวเข้มรีเวอร์ส }
	BlinkLow	=	\$87;	{ ตัวจางกระพริบ }
	BlinkHigh	=	\$8F;	{ ตัวเข้มกระพริบ }
	UndBlinkLow	=	\$81;	{ ตัวจางขีดเส้นใต้กระพริบ }
	UndBlinkhigh	=	\$89;	{ ตัวเข้มขีดเส้นใต้กระพริบ }
	RevBlinkLow	=	\$F0;	{ ตัวจางรีเวอร์สกระพริบ }
	RevBlinkHigh	=	\$F8;	{ ตัวเข้มรีเวอร์สกระพริบ }

ค่าคงตัวเหล่านี้ ใช้เป็นพารามิเตอร์สำหรับปรับแอสตริบิวต์ตัวอักษร ด้วยกระบวนการความ SetAttr (HighDisplay)

ค่าคงตัวกำหนดเซกเมนต์มีดังนี้

```
const ColorSeg      = $B800;
      MonoSeg       = $B000;
```

เมื่อ ColorSeg และ MonoSeg เป็นค่าคงตัวซึ่งเก็บค่าวิดีโอแรมเซกเมนต์ของจอสีและจอโมโนโครม ตามลำดับ

3.2 ค่าคงตัวกำหนดแบบข้อมูล แบบข้อมูลในยูนิตเป็นแบบข้อมูลสำหรับกำหนดเรคอร์ดของช่องหน้าต่างดังนี้

```
Type      ScreenLine = array [1..80]    of integer;
          ScreenArray = array [1..25]    of Screenline;
          ScreenBlock = array [1..2000]  of integer;
          WindowLink  = ^WindowControlblock;
          WindowControlBlock = record
              x1,y1,x2,y2    : integer; {Window boundaries}
              x,y            : integer  {Cursor location}
              ID             : byte     {ID Number}
              Backlink       : Windowlink;
              ScreenContents : ScreenBlock;
          end;
```

ตารางที่ 3.3 ค่าคงตัวกำหนดแบบข้อมูล

ชื่อแบบข้อมูล	การใช้งาน
ScreenLine	อะเรย์ของจำนวนเต็มขนาด 80 หน่วย เป็นแบบของข้อมูลของจอภาพ 1 บรรทัด ซึ่งประกอบด้วยตัวอักษรและแอดดรีบิต
ScreenArray	อะเรย์ขนาด 25 หน่วยของ ScreenLine ใช้เป็นแบบข้อมูลของจอภาพ หนึ่งหน้าจอ
ScreenBlock	อะเรย์ของตัวเลขจำนวนเต็ม 2000 ตัว ใช้เป็นแบบข้อมูลของจอภาพเช่นเดียวกับ ScreenArray แต่มองจอภาพเป็นบล็อกขนาดใหญ่ต่อเนื่องกันไป
WindowLink	พอยน์เตอร์ของ WindowControlBlock
WindowControlBlock	<p>เรคอร์ดของหน้าต่างประกอบด้วย</p> <p>x_1, y_1, x_2, y_2 เป็นขอบมุมบนซ้ายและมุมล่างขวาของหน้าต่าง</p> <p>x, y เป็นตำแหน่ง เคอร์เซอร์ปัจจุบัน</p> <p>ID เป็นหมายเลขหน้าต่าง</p> <p>BackLink เป็นพอยน์เตอร์ที่ไปยังหน้าต่างก่อนหน้า</p> <p>ScreenContent เป็นที่เก็บข้อมูลของจอภาพ</p>
HeadString	สตริงขนาด 80 ตัวอักษรสำหรับเป็นเฮดเดอร์ของจอหน้าต่าง

3.3 ค่าคงตัวกำหนดแบบตัวแปร

ตารางที่ 3.4 ค่าคงตัวกำหนดแบบตัวแปร

	การใช้งาน
WindowCount	เก็บจำนวนหน้าต่างที่เปิดใช้
ErrorWindow	เป็นรหัสผิดพลาดจากการเปิดช่องหน้าต่าง
VideoSeg	เป็นตัวแปรที่เก็บค่าวิดีโอเซกเมนต์ของจอภาพที่ใช้งานในขณะนั้น
CrtType	เป็นตัวแปรแบบไบนารีที่บ่งบอกชนิดจอภาพตามรหัสที่ได้จากไบออสดังนี้
	0 จอภาพแบบ 40x25 ขาวดำ
	1 จอภาพแบบ 40x25 สี
	2 จอภาพแบบ 80x25 ขาวดำ
	3 จอภาพแบบ 80x25 สี
	4 จอภาพแบบ 80x25 โมโนโครม

3.4 กระบวนการ ในหน่วย win ประกอบด้วยกระบวนการสำหรับการปิดเปิดช่องหน้าต่างและกระบวนการกำหนดพารามิเตอร์ของช่องหน้าต่าง

ตารางที่ 3.5 กระบวนการในหน่วย win

ชื่อกระบวนการ	การใช้งาน
WindowBox	วาดกรอบหน้าต่าง
WindowOpen	เปิดใช้ช่องหน้าต่าง
WindowClose	ยกเลิกช่องหน้าต่างปัจจุบัน
SetBoxStyle	กำหนดแบบกรอบหน้าต่าง

ตารางที่ 3.5 (ต่อ)

ชื่อกระบวนการ	การใช้งาน
SetAttr	ปรับแอตทริบิวต์การแสดงผลตัวอักษร
SetBoxAttr	กำหนดแอตทริบิวต์กรอบหน้าต่าง
SetWinAttr	กำหนดแอตทริบิวต์ของพื้นที่หน้าต่าง
SetCharAttr	กำหนดแอตทริบิวต์ของตัวอักษรในหน้าต่าง
SetHeadAttr	กำหนดแอตทริบิวต์ข้อความซึ่งจะพิมพ์ที่หัวหน้าต่าง
SetWinHeader	กำหนดข้อความซึ่งจะพิมพ์ที่หัวหน้าต่าง
SaveWinStatus	เก็บรักษาสถานภาพของจอหน้าต่าง
RestoreWinStatus	เรียกคืนสถานะภาพของจอหน้าต่าง

4. ยูนิต windo

ในยูนิต windo ประกอบด้วยกระบวนการสร้างกรอบหน้าต่างและปิดหน้าต่างที่ติดต่อกันแบบ interactive อยู่นั้น ดังนี้

ตารางที่ 3.6 กระบวนการในยูนิต windo

ชื่อกระบวนการ	การใช้งาน
Windlclose	ปิดกรอบหน้าต่างและตัวอักษรทุกตัวที่อยู่ภายในกรอบหน้าต่างนั้น
Open1	สร้างกรอบหน้าต่างโดยสามารถสร้างกรอบหน้าต่างซ้อนหลาย ๆ ชั้นได้

5. ยูนิต ldir

ในยูนิต ldir ประกอบด้วยกระบวนการ Displayfilename ซึ่งทำหน้าที่ในการสร้างไดเรกทอรีลิสต์และแสดงรายชื่อไดเรกทอรี

6. ยูนิต Swin0, Ssub

ยูนิตทั้ง 2 นี้มีหน้าที่ในการสร้าง Menu ย่อยจาก Menu หลักทั้ง 4 และเรียกใช้ ยูนิต Trans, Proc, Uti รวมทั้งเรียกใช้คำสั่ง Help จากแฟ้มข้อมูลที่มีนามสกุล Txt

7. ยูนิต Display

ยูนิต Display ประกอบด้วยกระบวนการความ ReadJTxt ซึ่งทำหน้าที่ในการอ่านแฟ้มข้อมูลที่ได้ทำการสร้างเก็บไว้เป็นที่เรียบร้อยแล้วแสดงผลออกทางจอภาพ ซึ่งในที่นี้จะใช้ในกรณี การกดฟังก์ชันคีย์ F1 ซึ่งคือ Help ในจอภาพ เมื่อกด F1 ครั้งใดก็จะมีการเรียกกระบวนการความ ReadJtxt ขึ้นทุกครั้ง

8. ยูนิต Fi

ยูนิต Fi ประกอบด้วยกระบวนการความ Makefi ซึ่งทำหน้าที่ในการสร้างแฟ้มข้อมูล, เพิ่มเติมข้อมูลลงไปในแฟ้มข้อมูลที่มีอยู่แล้วโดยวิธีการ append ต่อจาก บรรทัดสุดท้ายของแฟ้มข้อมูลเก่าที่มีข้อมูล

9. ยูนิต Trans, Proc, Uti

ยูนิตทั้ง 3 นี้เป็นยูนิตที่ถูกเรียกใช้โดยโปรแกรมหลัก SPSMENU มีหน้าที่ในการรับข้อมูลจากผู้ใช้เพื่อไปเก็บลงแฟ้มข้อมูล Ex.sps

เอกสารประกอบการใช้

โปรแกรม SPSMENU ที่พัฒนาขึ้นมา มีขั้นตอนการเตรียมและการเรียกใช้งานดังนี้

1. เมื่อเริ่มต้นจะใช้งาน ผู้ใช้ต้องเตรียมเครื่องไมโครคอมพิวเตอร์ ที่มีฮาร์ดดิสก์ และมีแผ่นดิสก์เก็ตที่มีแฟ้มข้อมูล INSTALL.BAT
2. เรียกใช้โปรแกรมโดยใช้คำสั่ง INSTALL.BAT ในไดรฟ์ A โดยที่แผ่นดิสก์เก็ตเกิดในไดรฟ์ A ประกอบด้วยแฟ้มข้อมูลต่างๆ ดังนี้ คือ

- SPSMENU.EXE
- EX.SPS
- EXO.SPS
- THAITIP.COM
- INSTALL.BAT
- START.BAT
- แฟ้มข้อมูลที่มีนามสกุล .TXT จำนวน 21 แฟ้มข้อมูล

หลังจากเรียกใช้คำสั่ง INSTALL.BAT แล้ว จะทำให้มีการ COPY แฟ้มข้อมูลต่างๆ แฟ้มข้อมูลที่อยู่ในไดรฟ์ A ยกเว้น แฟ้มข้อมูล INSTALL.BAT ลงไปที่ไดรฟ์ C โดยจะมีการสร้าง Subdirectory ในไดรฟ์ดังนี้

1. สร้าง Subdirectory SPTOOL เพื่อเก็บแฟ้มข้อมูล SPMENU.EXE, EX.SPS, EXO.SPS, THAITIP.COM และ START.BAT
2. สร้าง Subdirectory ออกจาก SPTOOL คือ HELP เพื่อเก็บแฟ้มข้อมูลที่มีนามสกุล .TXT จำนวน 21 แฟ้มข้อมูล
3. หลังจาก INSTALL แฟ้มข้อมูลข้อมูลเป็นที่เรียบร้อยแล้ว ขณะนี้ PROMPT จะอยู่ที่ C:\SPTOOL> เรียกใช้คำสั่ง START.BAT และจะเข้าสู่โปรแกรม จะปรากฏจอภาพ 1 (ดูภาคผนวก ก พร้อมวิธีการใช้)
4. ในกรณีที่ผู้ใช้ Exit ออกจากโปรแกรมนี้แล้ว จะนำแฟ้มข้อมูลที่ได้สร้างขึ้นไปวิเคราะห์ ผู้ใช้จะต้องมีโปรแกรมสำเร็จรูปทางสถิติ SPSS/PC หรือ SPSS/PC+ ติดตั้งเสียก่อนโดยมีขั้นตอน ดังนี้

4.1 ขณะอยู่ที่ Prompt ของ

C:\SPSS>

เรียกแฟ้มข้อมูล ZERODISK ก่อน

C:\SPSS>ZERODISK แล้วกด Enter

4.2 C:\SPSS>SPSSPC เรียกเพิ่มข้อมูล SPSSPC
แล้วกด Enter

4.3 เข้าสู่ตัวโปรแกรม SPSS/PC หรือ SPSS/PC⁺ ใช้คำสั่ง include
เรียกเพิ่มข้อมูลที่สร้างขึ้น โดยเพิ่มข้อมูลที่สร้างนี้ชื่อ EX.SPS
ดังตัวอย่าง

SPSS : include 'ex.sps'.

4.4 เมื่อได้ผลลัพธ์ที่ต้องการแล้ว ออกจาก SPSS/PC หรือ SPSS/PC⁺
โดยใช้คำสั่ง finish ดังตัวอย่าง

SPSS : finish.

สำหรับการเลือกใช้รายการ (Menu) ของแต่ละรายการจะใช้เป็นนิมฟ์ต่างๆ
ดังนี้

1. กลุ่มฟังก์ชัน (Function Keys) ที่ใช้ได้แก่ปุ่ม F1 , F10

F1 เมื่อต้องการดูรายละเอียดของรายการที่เลือก (Help)

F10 เมื่อต้องการออกจากระบบการใช้งานทั้งหมดของโปรแกรม SPSMENU

ตัวอย่าง เมื่อผู้ใช้อยู่ในหน้าต่าง สถิติเบื้องต้นที่แบ่งกลุ่ม เมื่อกดปุ่ม F1 จะปรากฏจอภาพ
ดังนี้

สถิติเบื้องต้นที่แบ่งกลุ่ม

เป็นคำสั่งที่ใช้ในการแบ่งข้อมูลเป็นกลุ่มย่อยแล้วคำนวณค่าสถิติ
เบื้องต้นภายในกลุ่มย่อยแต่ละกลุ่ม เช่น ค่าเฉลี่ย ค่าส่วนเบี่ยงเบนมาตรฐาน
จะแสดงค่าดังนี้

1. ตัวแปรที่ใช้พร้อมส่วนขยาย
2. ค่าสถิติเบื้องต้นของข้อมูลทุกชุด
3. ค่าสถิติเบื้องต้นของแต่ละกลุ่มย่อย
4. จำนวนชุดข้อมูลที่ใช้
5. ข้อมูลชุดที่มีค่า ไม่สมบูรณ์จะถูกตัดจากการคำนวณ

2. ลูกศร (Arrow) ที่ใช้คือ <- , -> ,
เพื่อทำการเลือกรายการของแต่ละรายการเลื่อนไปทางซ้าย ทางขวา
บน และล่าง ตามลำดับ
3. ปุ่ม Esc เมื่อต้องการออกจากแต่ละรายการ

5.1 สรุป อภิปราย

ในการทำวิจัยครั้งนี้ ได้พัฒนาโปรแกรมลักษณะรายการ (Menu) ที่ชื่อว่า SPSMENU เป็นภาษาไทยจากโปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺ และเขียนโปรแกรม โดยใช้ภาษาปาสคาล รุ่น 5.0 บนเครื่องไมโครคอมพิวเตอร์ IBM PC/XT Proturbo 88 ขนาด 16 Bits พร้อมฮาร์ดดิสก์

สำหรับโปรแกรมนี้สามารถคำนวณค่าสถิติเบื้องต้น (Descriptive Statistics) จนถึงสถิติหลายตัวแปร (Multivariate Statistics) และ สถิตินอนพารามิเตอร์ (Nonparametric Statistics) ซึ่งได้แก่

1. ความถี่ สำหรับข้อมูลเชิงคุณภาพ ซึ่งให้ความถี่ ค่าต่ำสุด สูงสุด ร้อยละ เปอร์เซนต์ไทล์ เอนไทล์ มัชยฐาน ฐานนิยม ตัวกลางเลขคณิต ส่วนเบี่ยงเบนมาตรฐาน ส่วนเบี่ยงเบนค่าเฉลี่ย ความโด่ง ความเบ้ ฮิสโตแกรม กราฟแท่ง ผลรวม
2. สถิติพรรณนา สำหรับข้อมูลเชิงปริมาณ ซึ่งให้ค่าสถิติทำนองเดียวกับค่าสังเกต ความถี่
3. ตารางการันเจอร์ ให้ตารางความถี่ ตั้งแต่ 2 ทางขึ้นไป รวมทั้งให้ค่าสถิติ Chi-square , Phi , Contingency Coefficient , Lamda Uncertainty Coefficient , Kendall's Tau-b , Kendall's Tau-c , Gamma , Somer'd , Eta , Pearson's r
4. ทดสอบ T-Test ใช้ทดสอบความแตกต่างค่าเฉลี่ยของกลุ่มตัวอย่าง 2 กลุ่ม ที่เป็นอิสระ (Independent Sample) และ ไม่เป็นอิสระ (Dependent Sample)
5. ความแปรปรวนทางเดียวและหลายทาง ใช้ทดสอบความแตกต่างค่าเฉลี่ยของกลุ่มตัวอย่างตั้งแต่ 2 กลุ่มขึ้นไป
6. สหสัมพันธ์ คำนวณค่าสหสัมพันธ์ (Correlation) รวมทั้งให้ค่าสถิติเบื้องต้น
7. วิเคราะห์ถดถอย ซึ่งใช้วิธีการ Stepwise , Forward , Backward Enter , Remove , Test ในการนำตัวแปรเข้าในสมการถดถอย
8. สถิตินอนพารามิเตอร์ ได้แก่ Binomial , Mcnemar , Wilcoxon , Cochran Q test , Frideman , Kendall , Median , Mann-Whitney U , Kolmogorov-Sminov , Wald-Wolfowitz ,

และ Kruskal-Wallis

โปรแกรมนี้จะสร้างแฟ้มข้อมูลที่ประกอบไปด้วยส่วนของคำสั่งโปรแกรมสำเร็จรูป SPSS/PC หรือ SPSS/PC⁺ เท่านั้นหลังจากนั้นต้องนำแฟ้มข้อมูลที่ได้ นำไปประมวลผล บนเครื่อง ไมโครคอมพิวเตอร์ที่มีโปรแกรมสำเร็จรูป SPSS/PC หรือ SPSS/PC⁺ ติดตั้งอยู่ ทั้งนี้ เพราะโปรแกรมสำเร็จรูปสถิตินี้เป็นโปรแกรมลักษณะคำสั่ง (Command Driven) ซึ่งผู้ใช้ จำเป็นต้องศึกษาถึงวิธีการใช้ก่อน และ วิธีการป้อนข้อมูลว่ามีรูปแบบอย่างไรบ้าง สำหรับวิธีการใช้โปรแกรมนี้ ปรากฏอยู่ในรายงานการวิจัยนี้แล้ว

5.2 ข้อเสนอแนะ

1. ควรมีการพัฒนาโปรแกรมนี้ต่อไป ให้สามารถใช้ติดต่อระหว่าง คอมพิวเตอร์ด้วยกันได้ (Data Communication) ทั้งนี้เพื่อประโยชน์ในการเรียนการสอนที่เกี่ยวข้องกับการใช้โปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺
2. ให้เป็นตัวอย่างกรณีศึกษา (Case Study) ของการพัฒนาโปรแกรม ทางสถิติ ซึ่งในประเทศไทย การพัฒนาโปรแกรมทางด้านนี้ยังมีน้อย

บรรณานุกรม

1. กำพล เต็มประยูร , ศิริชัย พงษ์วิชัย. โปรแกรมสำเร็จรูปทางสถิติสำหรับเครื่องไมโครคอมพิวเตอร์ โรงเรียนจุฬาลงกรณ์มหาวิทยาลัย.
2. วิชัย ปุณวัตร์. คู่มือการใช้ PC-DOS Version 3.20 & 3.30 โรงเรียนเอช-เอเนกาการพิมพ์.
3. สุรศักดิ์ สงวนพงษ์. แอดวานซ์เทอร์โบปาสคาล เวอร์ชัน 4 โรงเรียนเอช-เอเนกาการพิมพ์.
4. Borland International Inc. Turbo Pascal Version 5.0.
5. Marija J. Norusis. SPSS/PC⁺ SPSS Inc. Chicago , Illinois.

ภาคผนวก ก
คู่มือการใช้งาน

การสร้างโปรแกรมลักษณะรายการ (Menu)
จากโปรแกรมสำเร็จรูปทางสถิติ SPSS/PC และ SPSS/PC⁺

1. ทุนอุดหนุนการวิจัยจากคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์
วิทยาเขตหาดใหญ่
2. หัวหน้าโครงการวิจัย
 - อ. นทีกานต์ สุขเมฆสิทธิกุล
3. ผู้ร่วมโครงการ
 - น.ส. นัยนา เครืออูบล
 - น.ส. ลัดดา ปรีชาวีรกุล

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประ โยชน์
--------	----------------	------------	----------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

F1 Help F10 Exit

จอภาพ 1 ข้อมูล

จากจอภาพที่ 1 ใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- กำหนดลักษณะข้อมูล : จะปรากฏจอภาพ 1.1
- เปลี่ยนรูปข้อมูล : จะปรากฏจอภาพ 1.2
- คัดเลือกข้อมูล : จะปรากฏจอภาพ 1.3

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล เปลี่ยนรูปข้อมูล คัดเลือกข้อมูล	เรียกเพิ่มข้อมูล บอกรายละเอียดเกี่ยวกับตัวแปร บอกค่า ไม่สมบูรณ์ให้กับตัวแปร บอกรายละเอียดเกี่ยวกับค่าตัวแปร บอกรายละเอียดเกี่ยวกับชื่อตัวแปร
---	--

F1 Help Esc Menu

จอภาพ 1.1 กำหนดลักษณะข้อมูล

จากจอภาพ 1.1 เมื่อเลือกรายการการกำหนดลักษณะข้อมูลแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- เรียกเพิ่มข้อมูล : จะปรากฏเป็นจอภาพ 1.1.1
- บอกรายละเอียดเกี่ยวกับตัวแปร : จะปรากฏเป็นจอภาพ 1.1.2
- บอกค่า ไม่สมบูรณ์ให้กับตัวแปร : จะปรากฏเป็นจอภาพ 1.1.3
- บอกรายละเอียดเกี่ยวกับค่าตัวแปร : จะปรากฏเป็นจอภาพ 1.1.4
- บอกรายละเอียดเกี่ยวกับชื่อตัวแปร : จะปรากฏเป็นจอภาพ 1.1.5

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	ตัดประ โยชน์
--------	----------------	------------	--------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

เรียกเพิ่มข้อมูล

เพิ่มข้อมูลเก็บแบบ ASCII หรือ System file [A/S] []

ชื่อเพิ่มข้อมูล

Esc Menu

จอภาพ 1.1.1 เรียกเพิ่มข้อมูล

จากจอภาพ 1.1.1 เมื่อเลือกรายการเรียกเพิ่มข้อมูลให้ใส่ข้อมูลว่าเพิ่มข้อมูลที่เก็บเป็นเพิ่มข้อมูลเก็บแบบ ASCII หรือ System file [A/S] หลังจากนั้นให้ใส่ชื่อเพิ่มข้อมูล

นั่นคือ

การกำหนดลักษณะของเพิ่มข้อมูลที่เก็บไว้ เป็น ASCII หรือ System file

- ใส่ A หมายถึง ASCII ถ้าเก็บจาก Editor ต่างๆ เช่น Sidekick , Turbo Editor
- ใส่ S หมายถึง System file

ชื่อเพิ่มข้อมูล

ใส่ชื่อเพิ่มข้อมูลที่นำมาใช้วิเคราะห์ข้อมูล

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล เปลี่ยนรูปข้อมูล คัดเลือกข้อมูล	1. เรียงเพิ่มข้อมูล 2. บอกรายละเอียดเกี่ยวกับตัวแปร 3. บอกค่าไม่สมบูรณ์ให้กับตัวแปร 4. บอกรายละเอียด 5. บอกรายละเอียด	1. กำหนดตำแหน่งคอลัมน์แบบคงที่ 2. กำหนดตำแหน่งคอลัมน์แบบอิสระ
---	---	--

Esc Menu

จอภาพ 1.1.2 บอกรายละเอียดเกี่ยวกับตัวแปร

จากจอภาพ 1.1.2 เมื่อรายการบอกรายละเอียดเกี่ยวกับตัวแปรแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- กำหนดตำแหน่งคอลัมน์แบบคงที่ : จะปรากฏเป็นจอภาพ 1.1.2.1
- กำหนดตำแหน่งคอลัมน์แบบอิสระ : จะปรากฏเป็นจอภาพ 1.1.2.2

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

กำหนดลักษณะข้อมูล เปลี่ยนรูปข้อมูล คัดเลือกข้อมูล	เรียกเพิ่มข้อมูล บอกรายละเอียดเกี่ยวกับตัวแปร บอกค่า ไม่สมบูรณ์ให้กับตัวแปร บอกรายละเอียดเกี่ยวกับค่าตัวแปร บอกรายละเอียดเกี่ยวกับชื่อตัวแปร
---	--

กำหนดตำแหน่งคงที่

ชื่อตัวแปร

ประเภทข้อมูลของตัวแปร *

ตำแหน่งคอลัมน์ของตัวแปร

* a : ตัวอักษร หรือ
ใส่จำนวนทศนิยม ถ้าจำนวนทศนิยมเป็น 0 : กดผ่าน

Esc Menu

จอภาพ 1.1.2.1 กำหนดตำแหน่งคอลัมน์แบบคงที่

จากจอภาพ 1.1.2.1 เมื่อรายการกำหนดตำแหน่งคงที่ใส่ข้อมูล ได้ดังนี้

- ชื่อตัวแปร : ใส่ชื่อตัวแปร
- ประเภทข้อมูลของตัวแปร : ใส่ประเภทข้อมูลของตัวแปรที่ต้องการ
กด Enter เมื่อมีตัวเลขหลังจุดทศนิยมเป็น ศูนย์
ใส่ ตัวเลข เมื่อมีตัวเลขหลังจุดทศนิยม
ใส่ a เมื่อข้อมูลเป็นแบบข้อความ
- ตำแหน่งคอลัมน์ของตัวแปร : ใส่ตำแหน่งคอลัมน์ของตัวแปรซึ่งตรงกับในเมนูข้อมูลที่ เป็นข้อมูล

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

กำหนดลักษณะข้อมูล เปลี่ยนรูปข้อมูล คัดเลือกข้อมูล	เรียกเพิ่มข้อมูล บอกรายละเอียดเกี่ยวกับตัวแปร บอกค่า ไม่สมบูรณ์ให้กับตัวแปร บอกรายละเอียดเกี่ยวกับค่าตัวแปร บอกรายละเอียดเกี่ยวกับชื่อตัวแปร
---	--

กำหนดตำแหน่งข้อมูลแบบอิสระ

ชื่อตัวแปร

ประเภทตัวแปร *

* A สำหรับตัวอักษร หรือ ระบุความกว้างของตัวแปรที่เป็นข้อความ

Esc Menu

จอภาพ 1.1.2.2 กำหนดตำแหน่งคอลัมน์แบบอิสระ

- จากจอภาพ 1.1.2.2 เมื่อเลือกรายการกำหนดตำแหน่งข้อมูลแบบอิสระใส่ข้อมูล ได้ดังนี้
- ชื่อตัวแปร : ใส่ชื่อตัวแปรเพื่อกำหนดตำแหน่งข้อมูล
- ประเภทตัวแปร : กด Enter ถ้าเป็นตัวเลข
ใส่ A ถ้าเป็นตัวอักษร และ ใส่ตัวเลขบอกความกว้างเป็นคอลัมน์ของตัวแปร
เช่น ใส่ค่า 8 เมื่อตัวแปรที่เป็นข้อความ นั้นมีความกว้างเป็น 8

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

ค่า ไม่สมบูรณ์ ให้กับตัวแปร

ชื่อตัวแปร	<input type="text"/>
รหัสแทนค่าที่ไม่สมบูรณ์	<input type="text"/>

Esc Menu

จอภาพ 1.1.3 บอกค่าที่ไม่สมบูรณ์ให้กับตัวแปร

จากจอภาพ 1.1.3 เมื่อเลือกรายการค่า ไม่สมบูรณ์แล้ว ใส่ข้อมูล ได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการระบุค่าที่ขาดหายไปหรือไม่สมบูรณ์
(Missing values)

รหัสแทนค่าที่ไม่สมบูรณ์ : ใส่ค่าที่เป็นรหัสแทนค่าที่ขาดหายไปของตัวแปรที่ระบุชื่อตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราเปอร์เซ็นต์
--------	----------------	------------	------------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

รายละเอียดเกี่ยวกับค่าตัวแปร

ชื่อตัวแปร	<input type="text"/>
ค่าของตัวแปร	<input type="text"/>
ส่วนขยายของตัวแปร	<input type="text"/>

Esc Menu

จอภาพ 1.1.4 บอกรายละเอียดเกี่ยวกับตัวแปร

จากจอภาพ 1.1.4 เมื่อเลือกรายการรายละเอียดเกี่ยวกับค่าตัวแปรแล้วใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการอธิบายค่าของตัวแปร

ค่าของตัวแปร : ค่าของตัวแปรที่ต้องการอธิบายค่า

ส่วนขยายของตัวแปร : ใส่คำอธิบายหรือส่วนขยายของค่าของตัวแปร
(Value Labels)

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

รายละเอียดเกี่ยวกับชื่อตัวแปร

ชื่อตัวแปร	<input type="text"/>
ส่วนขยายของตัวแปร	<input type="text"/>

Esc Menu

จอภาพ 1.1.5 บอกรายละเอียดเกี่ยวกับชื่อตัวแปร

จากจอภาพ 1.1.5 เมื่อเลือกรายการรายละเอียดเกี่ยวกับชื่อตัวแปรแล้วใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ชื่อตัวแปรที่อธิบายชื่อของตัวแปร

ส่วนขยายของตัวแปร : รายละเอียดหรือส่วนขยายชื่อตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

เปลี่ยนค่าของตัวแปร
สร้างตัวแปรใหม่โดยการคำนวณ
สร้างตัวแปรใหม่โดยการกำหนดเงื่อนไข
สร้างตัวแปรใหม่โดยการนับ
กำหนดน้ำหนักแก่ตัวแปร

F1 Help Esc Menu

จอภาพ 1.2 เปลี่ยนรูปข้อมูล

จากจอภาพ 1.2 เมื่อเลือกรายการเปลี่ยนรูปข้อมูลแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการ
ได้ดังนี้

- เปลี่ยนค่าของตัวแปร : จะปรากฏจอภาพ 1.2.1
- สร้างตัวแปรใหม่โดยการคำนวณ : จะปรากฏจอภาพ 1.2.2
- สร้างตัวแปรใหม่โดยการกำหนดเงื่อนไข : จะปรากฏจอภาพ 1.2.3
- สร้างตัวแปรใหม่โดยการนับ : จะปรากฏจอภาพ 1.2.4
- กำหนดน้ำหนักแก่ตัวแปร : จะปรากฏจอภาพ 1.2.5

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	ตัดประโยชน์
--------	----------------	------------	-------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

เปลี่ยนค่าของตัวแปร

ชื่อตัวแปร	<input type="text"/>
ค่าหรือรหัสเดิม	<input type="text"/>
ค่าหรือรหัสใหม่	<input type="text"/>

Esc Menu

จอภาพ 1.2.1 เปลี่ยนค่าของตัวแปร

- จากจอภาพ 1.2.1 เมื่อเลือกรายการเปลี่ยนค่าของตัวแปรแล้วใส่ข้อมูลดังนี้
- ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการเปลี่ยนค่าใหม่
 - ค่าหรือรหัสเดิม : ใส่ค่าหรือรหัสเดิมของค่าตัวแปร
 - ค่าหรือรหัสใหม่ : ใส่ค่าหรือรหัสใหม่ที่ผู้ใช้ต้องการ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

สร้างตัวแปรใหม่โดยการคำนวณ

ชื่อตัวแปรใหม่	<input type="text"/>
นิพจน์ทางคณิตศาสตร์	<input type="text"/>

Esc Menu

จอภาพ 1.2.2 สร้างตัวแปรใหม่โดยการคำนวณ

- จากจอภาพ 1.2.2 เมื่อเลือกรายการสร้างตัวแปรใหม่โดยการคำนวณแล้ว ใส่ข้อมูลดังนี้
- ชื่อตัวแปรใหม่ : ใส่ชื่อตัวแปรใหม่ซึ่งมีค่าที่ได้จากการคำนวณ
- นิพจน์ทางคณิตศาสตร์ : ประกอบด้วยเครื่องหมายทางคณิตศาสตร์ (+, -, *, /, **)

โดยที่

- + หมายถึง บวก
- ลบ
- * คูณ
- / หาร
- ** ยกกำลัง

หรือใส่ หรือ ฟังก์ชันที่เกี่ยวข้องกับตัวเลข เช่น sqrt
หมายถึง หารากที่ 2

เช่น $incom1 + incom2$ หรือ $sqrt(y)$

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

สร้างตัวแปรใหม่โดยการกำหนดเงื่อนไข

ชื่อตัวแปรใหม่

นิพจน์ทางตรรก

ค่าหรือนิพจน์ที่กำหนดให้

Esc Menu

จอภาพ 1.2.3 สร้างตัวแปรใหม่โดยการกำหนดเงื่อนไข

จากจอภาพ 1.2.3 เมื่อเลือกรายการสร้างตัวแปรใหม่โดยการกำหนดเงื่อนไขแล้ว
ใส่ข้อมูลดังนี้

ชื่อตัวแปรใหม่ : ใส่ชื่อตัวแปรที่ต้องการจะสร้างใหม่โดยการกำหนดเงื่อนไข
นิพจน์ทางตรรก : เงื่อนไขในการสร้างตัวแปร

โดยใช้เครื่องหมายแสดงความสัมพันธ์และการเปรียบเทียบ
ดังนี้

EQ	หรือ	=	หมายถึง	เท่ากับ
LT	หรือ	<	หมายถึง	น้อยกว่า
GT	หรือ	>	หมายถึง	มากกว่า
LE	หรือ	<=	หมายถึง	น้อยกว่าหรือเท่ากับ
GE	หรือ	>=	หมายถึง	มากกว่าหรือเท่ากับ
NE	หรือ	<>	หมายถึง	ไม่เท่ากัน

เช่น age GE 18 หรือ income LT 3000

ค่าหรือนิพจน์ที่กำหนด : ค่าหรือนิพจน์ที่กำหนดให้แก่ตัวแปรใหม่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

สร้างตัวแปรใหม่โดยการนับ

ชื่อตัวแปรใหม่	<input type="text"/>
รายชื่อตัวแปรที่เป็นเงื่อนไข	<input type="text"/>
กำหนดค่ารายชื่อตัวแปร	<input type="text"/>

Esc Menu

จอภาพ 1.2.4 สร้างตัวแปรใหม่โดยการนับ

- จากจอภาพ 1.2.4 เมื่อเลือกรายการสร้างตัวแปรใหม่โดยการนับแล้วใส่ข้อมูลได้ดังนี้
- ชื่อตัวแปรใหม่ : ใส่ชื่อตัวแปรใหม่เพื่อใช้แทนค่าจำนวนตัวแปรที่เป็นไปตามเงื่อนไข
 - รายชื่อตัวแปรที่เป็นเงื่อนไข : ใส่ชื่อกลุ่มตัวแปรที่กำหนดเป็นเงื่อนไขในการนับ
 - กำหนดค่ารายชื่อตัวแปร : ใส่ค่าหรือกลุ่มของค่าของตัวแปรตามที่กำหนดไว้

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	ตัดทประ โยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

กำหนดน้ำหนักของตัวแปร

ชื่อตัวแปร

F1 Help Esc Menu

จอภาพ 1.2.5 กำหนดน้ำหนักแก่ตัวแปร

จากจอภาพ 1.2.5 เมื่อเลือกรายการกำหนดน้ำหนักของตัวแปรแล้วใส่ข้อมูล ได้ดังนี้
ชื่อตัวแปร : ใส่ชื่อตัวแปรเพียง 1 ตัวที่ต้องการให้น้ำหนักความสำคัญ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

จำนวนชุดข้อมูล
โดยวิธีการสุ่มตัวอย่าง
โดยกำหนดเงื่อนไขทางตรรก
โดยกำหนดเงื่อนไขทางตรรก (Procedure)

F1 Help Esc Menu

จอภาพ 1.3 คัดเลือกข้อมูล

จากจอภาพ 1.3 เมื่อเลือกรายการคัดเลือกข้อมูลแล้วให้ลูกศรเลื่อนเพื่อเลือกรายการ
ได้ดังนี้

- จำนวนชุดข้อมูล : จะปรากฏเป็นจอภาพ 1.3.1
- โดยวิธีการสุ่มตัวอย่าง : จะปรากฏเป็นจอภาพ 1.3.2
- โดยกำหนดเงื่อนไขทางตรรก : จะปรากฏเป็นจอภาพ 1.3.3
- โดยกำหนดเงื่อนไขทางตรรก (Procedure) : จะปรากฏเป็นจอภาพ 1.3.4

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
ตัดเลือกข้อมูล

กำหนดจำนวนชุดข้อมูล

จำนวนข้อมูลที่ตัดเลือกมาวิเคราะห์

Esc Menu

จอภาพ 1.3.1 จำนวนชุดข้อมูล

จากจอภาพ 1.3.1 เมื่อเลือกรายการกำหนดชุดข้อมูลแล้วใส่ข้อมูล ได้ดังนี้
จำนวนข้อมูลที่ตัดเลือกมาวิเคราะห์ : ใส่จำนวนชุดข้อมูลเป็นเลขจำนวนเต็ม
เพื่อใช้ตัดเลือกข้อมูล

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัดจอประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล

วิธีการสุ่มตัวอย่าง	
สัดส่วนตัวอย่าง	<input type="text"/>
ขนาดของตัวอย่าง	<input type="text"/>
ขนาดของประชากร	<input type="text"/>

Esc Menu

จอภาพ 1.3.2 โดยวิธีการสุ่มตัวอย่าง

จากจอภาพ 1.3.2 สัดส่วนตัวอย่าง

เมื่อเลือกรายการสุ่มตัวอย่างแล้วส่งข้อมูลได้ดังนี้

: ใส่ตัวเลขเพื่อแสดงสัดส่วนของตัวอย่างที่ต้องการสุ่มมาซึ่งถือว่าข้อมูลทั้งหมด มีสัดส่วนเป็น 1 หรือ 100%

ในกรณี ที่ใส่ค่าสัดส่วนของตัวอย่างที่ต้องการสุ่มแล้ว โปรแกรมจะ ไม่ให้ใส่ค่าของขนาดของตัวอย่าง และขนาดของประชากร แต่ถ้า กด Enter ผ่าน จะต้องใส่ค่าของขนาดของตัวอย่างและขนาดของประชากร

เช่น ใส่ค่า 0.3 ที่ตำแหน่งเคอร์เซอร์รับข้อมูลที่

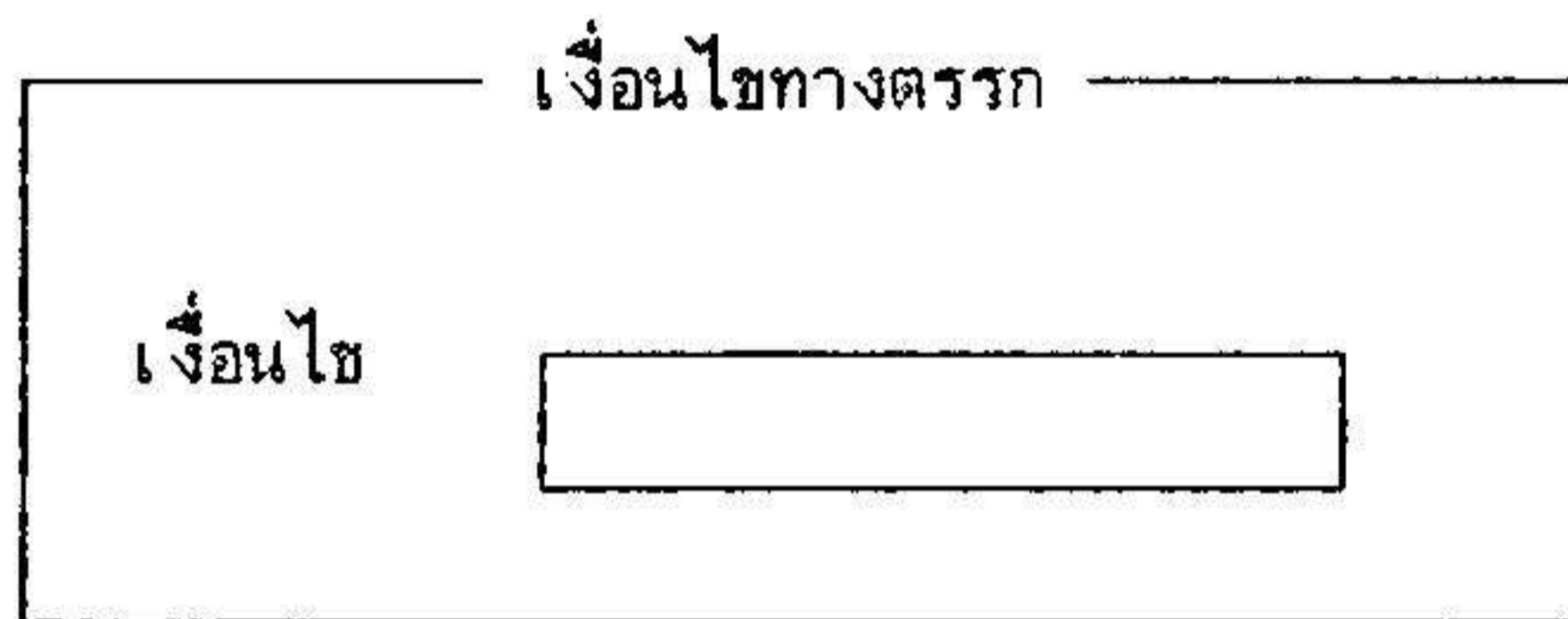
สัดส่วนของตัวอย่าง แต่ถ้า กด Enter ผ่าน เคอร์เซอร์
จะรอรับข้อมูลที่ขนาดของตัวอย่างและขนาดของประชากร

ขนาดของตัวอย่าง : ใส่จำนวนตัวเลขเพื่อแสดงขนาดของตัวอย่างที่ต้องการ

ขนาดของประชากร : ใส่จำนวนตัวเลขเพื่อแสดงขนาดของประชากรทั้งหมดที่มี

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดเลือกข้อมูล



Esc Menu

จอภาพ 1.3.3 โดยกำหนดเงื่อนไขทางตรรก

จากจอภาพ 1.3.3

เมื่อเลือกรายการเงื่อนไขทางสถิติซึ่งจะมีผลตลอด โปรแกรมแล้ว
ใส่ข้อมูล ได้ดังนี้

เงื่อนไข : ใส่เงื่อนไขทางตรรกเพื่อการคัดเลือกตัวแปร ซึ่งประกอบด้วย

- คำหรือเครื่องหมายที่แสดงความสัมพันธ์และการเปรียบเทียบ ได้แก่
EQ LT GT LE GE และ NE หรือ
= < > <= >= และ <>
- คำพิเศษทางตรรก ได้แก่ AND OR NOT
- เครื่องหมายทางคณิตศาสตร์ ได้แก่
+ - * / ** ()
- ฟังก์ชันเกี่ยวกับค่าไม่สมบูรณ์ ได้แก่
SYSMIS, MISSING VALUE

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

กำหนดลักษณะข้อมูล
เปลี่ยนรูปข้อมูล
คัดลอกข้อมูล

เงื่อนไขทางตรรก

เงื่อนไข

Esc Menu

จากจอภาพ 1.3.4 เมื่อเลือกรายการเงื่อนไขทางตรรกแบบชั่วคราว ซึ่งมี
จะมีผลชั่วคราวเฉพาะกับคำสั่ง Procedure ที่ตามมาเท่านั้น
และเงื่อนไขที่กำหนดจะต้องเป็นเงื่อนไขเดียวเท่านั้น แล้วใส่
การกำหนดข้อมูลเงื่อนไข ทำนองเดียวกันกับการกำหนดในจอ
ภาพ 1.3.3

ข้อมูล	คำสั่งทางสถิติ	คำสั่งซ้าย	อักขระไฮรน์
--------	----------------	------------	-------------

ความถี่
สถิติพรรณนา
สถิติเบื้องต้นที่แบ่งกลุ่ม
ตารางการแจกแจง
ทดสอบ T-Test
ความแปรปรวนทางเดียว
ความแปรปรวนหลายทาง
สหสัมพันธ์
วิเคราะห์ถดถอย
เขียนกราฟ
ทดสอบ Nonparametric

F1 Help F10 Exit

จอภาพ 2 คำสั่งทางสถิติ

จากจอภาพ 2 ใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- ความถี่ : จะปรากฏเป็นจอภาพ 2.1
- สถิติพรรณนา : จะปรากฏเป็นจอภาพ 2.2
- สถิติเบื้องต้นที่แบ่งกลุ่ม : จะปรากฏเป็นจอภาพ 2.3
- ตารางการแจกแจง : จะปรากฏเป็นจอภาพ 2.4
- ทดสอบ T-Test : จะปรากฏเป็นจอภาพ 2.5
- ความแปรปรวนทางเดียว : จะปรากฏเป็นจอภาพ 2.6
- ความแปรปรวนหลายทาง : จะปรากฏเป็นจอภาพ 2.7
- สหสัมพันธ์ : จะปรากฏเป็นจอภาพ 2.8

- วิเคราะห์ถดถอย : จะปรากฏเป็นจอภาพ 2.9
- เขียนกราฟ : จะปรากฏเป็นจอภาพ 2.10
- สถิติไคร้รวมมา : จะปรากฏเป็นจอภาพ 2.11

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ตัวแปร
 รวมค่าไม่สมบูรณ์
 กราฟแท่ง
 ฮิสโตแกรม
 เปอร์เซนต์ไทล์
 เอนไทล์
 ค่าสถิติ

F1 Help Esc Menu

จอภาพ 2.1 ความถี่

จากจอภาพ 2.1 เมื่อเลือกรายการความถี่ใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- ตัวแปร : จะปรากฏเป็นจอภาพ 2.1.1
- รวมค่าไม่สมบูรณ์ : จะทำการรวมค่าไม่สมบูรณ์ไปในการคำนวณด้วย

- กราฟแท่ง : จะปรากฏเป็นจอภาพ 2.1.2
- ฮิสโตแกรม : จะปรากฏเป็นจอภาพ 2.1.3
- เปอร์เซ็นไทล์ : จะปรากฏเป็นจอภาพ 2.1.4
- เอนไทล์ : จะปรากฏเป็นจอภาพ 2.1.5
- ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.1.6

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อັคตประโชชน์
--------	----------------	------------	--------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการกระจาย
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

F1 Help Esc Menu

จอภาพ 2.1.1 ชื่อตัวแปรของความถี่

จากจอภาพ 2.1.1 เมื่อเลือกรายการความถี่แล้วใส่ข้อความได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการหาความถี่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	ตัดประ โชน์
--------	----------------	------------	-------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นแบ่งกลุ่ม
 ตารางการกระจาย
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

กราฟแท่ง
 ระบุส่วนขยายหรือไม่ [Y/N] []

F1 Help Esc Menu

จอภาพ 2.1.2 ระบุส่วนขยายในกราฟแท่ง

จากจอภาพ 2.1.2 เมื่อเลือกรายการกราฟแท่งแล้วใส่ข้อมูลได้ดังนี้

โดยให้ใส่ส่วนขยาย คือ

ถ้าตอบ Y จะให้ใส่รายละเอียดของกราฟแท่งซึ่งจะปรากฏจอภาพ 2.1.2.1

ถ้าตอบ N กลับเข้าสู่รายการของการให้เลือกความถี่ (จอภาพ 2.1)

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่			
สถิติพรรณนา			
สถิติเบื้องต้นที่แบ่งกลุ่ม			
ตารางการกระจาย			
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			

กราฟแท่ง	
ค่าต่ำสุด	[]
ค่าสูงสุด	[]
ช่วงของความถี่	[]
ช่วงของความถี่ในรูปร้อยละ	[]

Esc Menu

จอภาพ 2.1.2.1 ระบุค่าต่ำสุดและสูงสุด, ความถี่ในกราฟแท่ง

จากจอภาพ 2.1.2.1 เมื่อเลือกรายการกราฟแท่งแล้วให้ใส่ข้อมูลได้ดังนี้

- ค่าต่ำสุด : ระบุค่าต่ำสุดของตัวแปรที่ต้องการเขียนกราฟ
- ค่าสูงสุด : ระบุค่าสูงสุดของตัวแปรที่ต้องการเขียนกราฟ
- ช่วงของความถี่ : ระบุช่วงความถี่ของตัวแปรซึ่งจะแสดงในแกนนอน
- ช่วงของความถี่ในรูปร้อยละ : ใช้ค่าร้อยละของความถี่ของตัวแปรซึ่งแสดงในแกนนอน

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราเปอร์เซ็นต์
--------	----------------	------------	------------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการกระจาย
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ยี่สิบโปรแกรม
 ระบุส่วนขยายหรือไม่ [Y/N] []

F1 Help Esc Menu

จอภาพ 2.1.3 ยี่สิบโปรแกรม

จากจอภาพ 2.1.3 เมื่อเลือกรายการความถี่แล้วใส่ข้อมูลได้ดังนี้

โดยให้ใส่ส่วนขยาย คือ

ถ้าตอบ Y จะให้ใส่รายละเอียดของยี่สิบโปรแกรมซึ่งจะปรากฏจอภาพ 2.1.3.1

ถ้าตอบ N กลับเข้าสู่รายการของการให้เลือกความถี่ (จอภาพ 2.1)

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
สถิติพรรณนา
สถิติเบื้องต้นที่แบ่งกลุ่ม
ตารางการกระจาย
ทดสอบ T-Test
ความแปรปรวนทางเดียว
ความแปรปรวนหลายทาง
สหสัมพันธ์
วิเคราะห์ถดถอย
เขียนกราฟ
ทดสอบ Nonparametric

ฮีสโตแกรม	
ค่าต่ำสุด	[]
ค่าสูงสุด	[]
ช่วงของความถี่	[]
ช่วงของความถี่ในรูปร้อยละ	[]

Esc Menu

จอภาพ 2.1.3.1 ระบุค่าต่ำสุด สูงสุด ความถี่ในฮีสโตแกรม

จากจอภาพ 2.1.3.1 เมื่อเลือกรายการฮีสโตแกรมแล้วให้ใส่ข้อมูลได้ดังนี้

- ค่าต่ำสุด : ระบุค่าต่ำสุดของตัวแปรที่ต้องการเขียนกราฟ
- ค่าสูงสุด : ระบุค่าสูงสุดของตัวแปรที่ต้องการเขียนกราฟ
- ช่วงของความถี่ : ระบุช่วงความถี่ของตัวแปรซึ่งจะแสดงในแกนนอน
- ช่วงของความถี่ในรูปร้อยละ : ใช้ค่าร้อยละของความถี่ของตัวแปรซึ่งแสดงในแกนนอน

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประกอบ
--------	----------------	------------	------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

เปอร์เซ็นต์ไทล์
 ตำแหน่งที่ต้องการหาค่า []

Esc Menu

จอภาพ 2.1.4 เปอร์เซ็นต์ไทล์

จากจอภาพ 2.1.4 เมื่อเลือกรายการเปอร์เซ็นต์ไทล์แล้วใส่ข้อมูล ได้ดังนี้
 ตำแหน่งที่ต้องการหาค่า : ใส่ตำแหน่งของเปอร์เซ็นต์ไทล์ที่ต้องการหาค่า

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	ยึดประ โยชน์
--------	----------------	------------	--------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric

เอนไทร์ ระบุตัวเลขที่ต้องการแบ่งส่วน []

Esc Menu

จอภาพ 2.1.5 เอนไทร์

จากจอภาพ 2.1.5 เมื่อเลือกรายการเอนไทร์แล้วใส่ข้อมูลได้ดังนี้

ระบุตัวเลขที่ต้องการแบ่งส่วน : ใส่ตัวเลขที่ต้องการแบ่งส่วนของข้อมูล

เช่น ใส่ตัวเลขเป็น 4 เมื่อต้องการหา ควอไทล์

ใส่ตัวเลขเป็น 10 เมื่อต้องการหา เดไซล์

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
<p>ความถี่</p> <p>สถิติพรรณนา</p> <p>สถิติเบื้องต้นที่แบ่งกลุ่ม</p> <p>ตารางการแจกแจง</p> <p>ทดสอบ T-Test</p> <p>ความแปรปรวนทางเดียว</p> <p>ความแปรปรวนหลายทาง</p> <p>สหสัมพันธ์</p> <p>วิเคราะห์ถดถอย</p> <p>เขียนกราฟ</p> <p>ทดสอบ Nonparametric</p>	<p>ตัวกลางเลขคณิต</p> <p>ค่าคลาดเคลื่อนมาตรฐานของค่าเฉลี่ย</p> <p>มัธยฐาน</p> <p>ฐานนิยม</p> <p>ส่วนเบี่ยงเบนมาตรฐาน</p> <p>ความแปรปรวน</p> <p>ความโค้ง</p> <p>ค่าคลาดเคลื่อนมาตรฐานของความโค้ง</p> <p>ความเบ้</p> <p>ค่าคลาดเคลื่อนมาตรฐานของความเบ้</p> <p>นิจัย</p> <p>ค่าต่ำสุด</p> <p>ค่าสูงสุด</p> <p>ผลรวม</p> <p>แสดงค่าสถิติทั้งหมด</p>		
Esc Menu			

จอภาพ 2.1.6 ค่าสถิติในความถี่

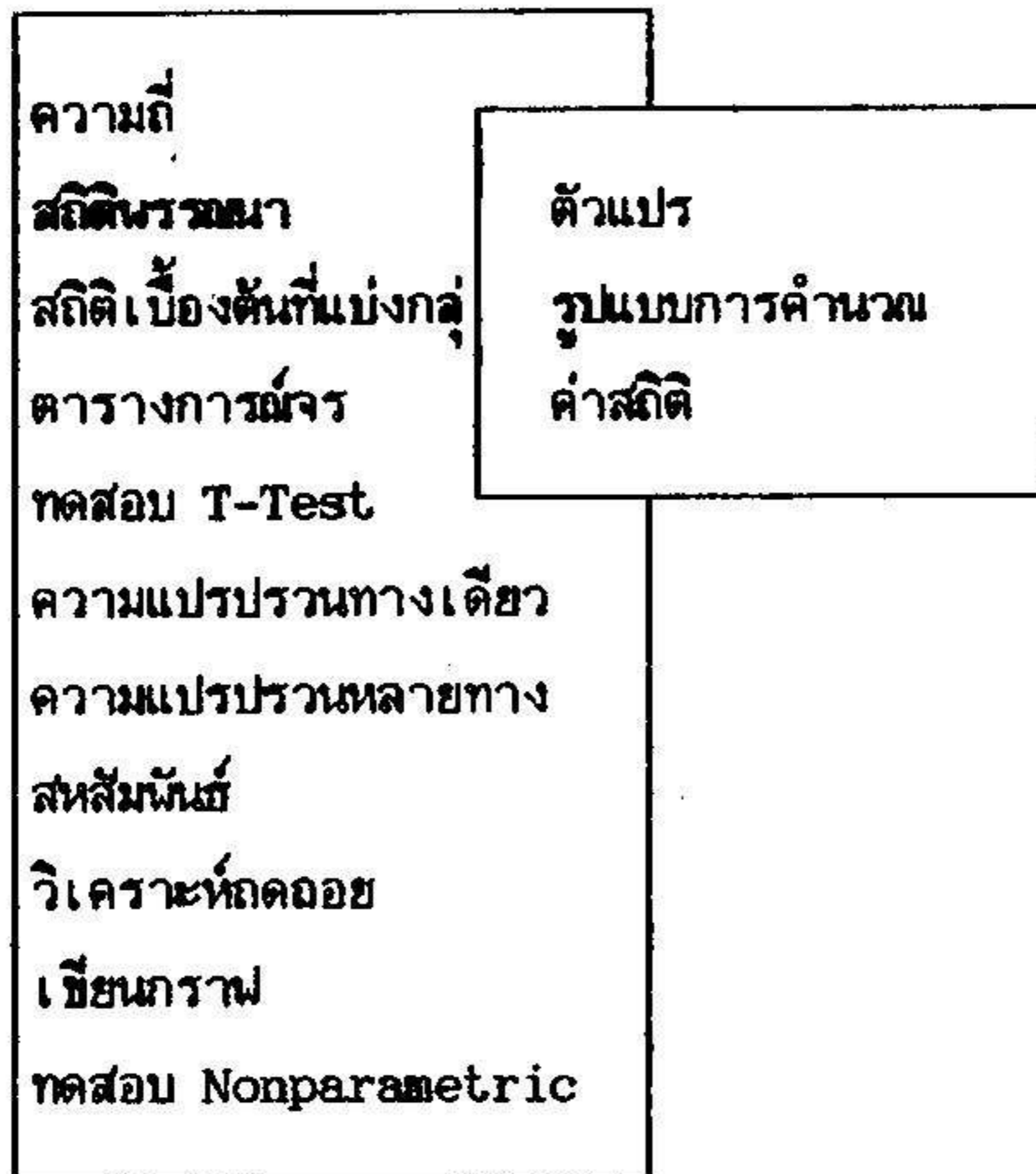
จากจอภาพ 2.1.6 เมื่อเลือกรายการความถี่แล้วใช้ลูกศรเลื่อนเพื่อเลือกค่าสถิติ แล้วกด

Enter จะปรากฏรายการให้เลือก คือ

- ตัวกลางเลขคณิต (Arithematics Mean)
- ค่าคลาดเคลื่อนมาตรฐานของค่าเฉลี่ย (Standard Error of the Mean)
- มัธยฐาน (Median)

- ฐานนิยม (Mode)
- ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)
- ความแปรปรวน (Variance)
- ความโด่ง (Skewness)
- ค่าคลาดเคลื่อนมาตรฐานของความโด่ง (Standard Error of the Skewness)
- ความเบ้ (Kurtosis)
- ค่าคลาดเคลื่อนมาตรฐานของความเบ้ (Standard Error of the Kurtosis)
- พิสัย (Range)
- ค่าต่ำสุด (Minimum)
- ค่าสูงสุด (Maximum)
- ผลรวม (Sum)
- แสดงค่าสถิติทั้งหมดของค่าสถิติข้างบนดังกล่าว (All Avariable Statistics)

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



F1 Help Esc Menu

จอภาพ 2.2 สถิติพรรณนา

จากจอภาพ 2.2 เมื่อเลือกรายการสถิติพรรณนาแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการต่อได้ดังนี้

- ตัวแปร : จะปรากฏเป็นจอภาพ 2.2.1
- รูปแบบการคำนวณ : จะปรากฏเป็นจอภาพ 2.2.2
- ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.2.3

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	ตัดทประโยชน์
--------	----------------	------------	--------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

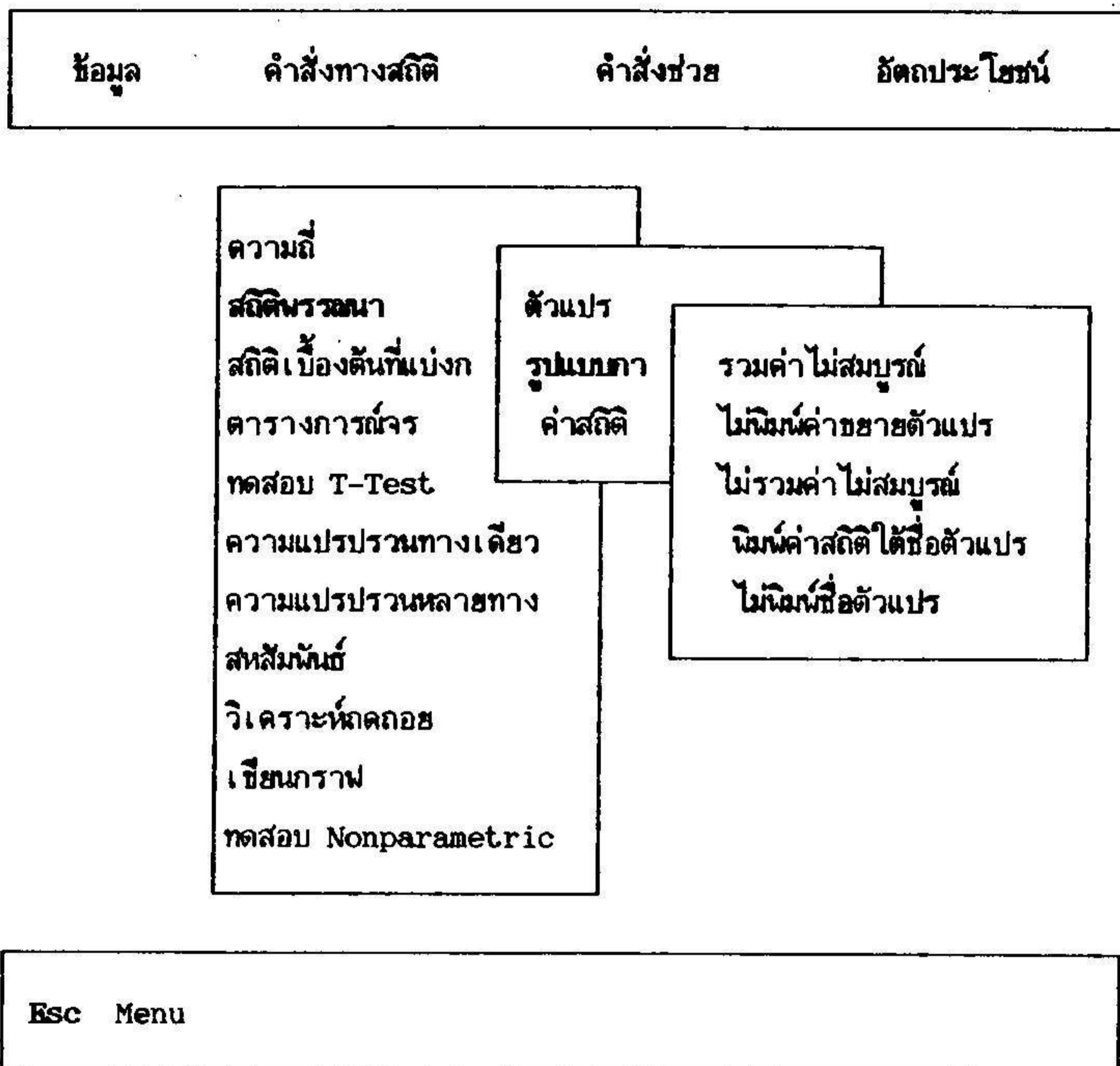
ชื่อตัวแปร

Esc Menu

จอภาพ 2.2.1 ตัวแปรในสถิติพรรณนา

จากจอภาพ 2.2.1 เมื่อเลือกรายการสถิติแล้วให้ใส่ข้อความ ได้ดังนี้

- ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการหาค่าสถิติเบื้องต้นที่แบ่งกลุ่ม

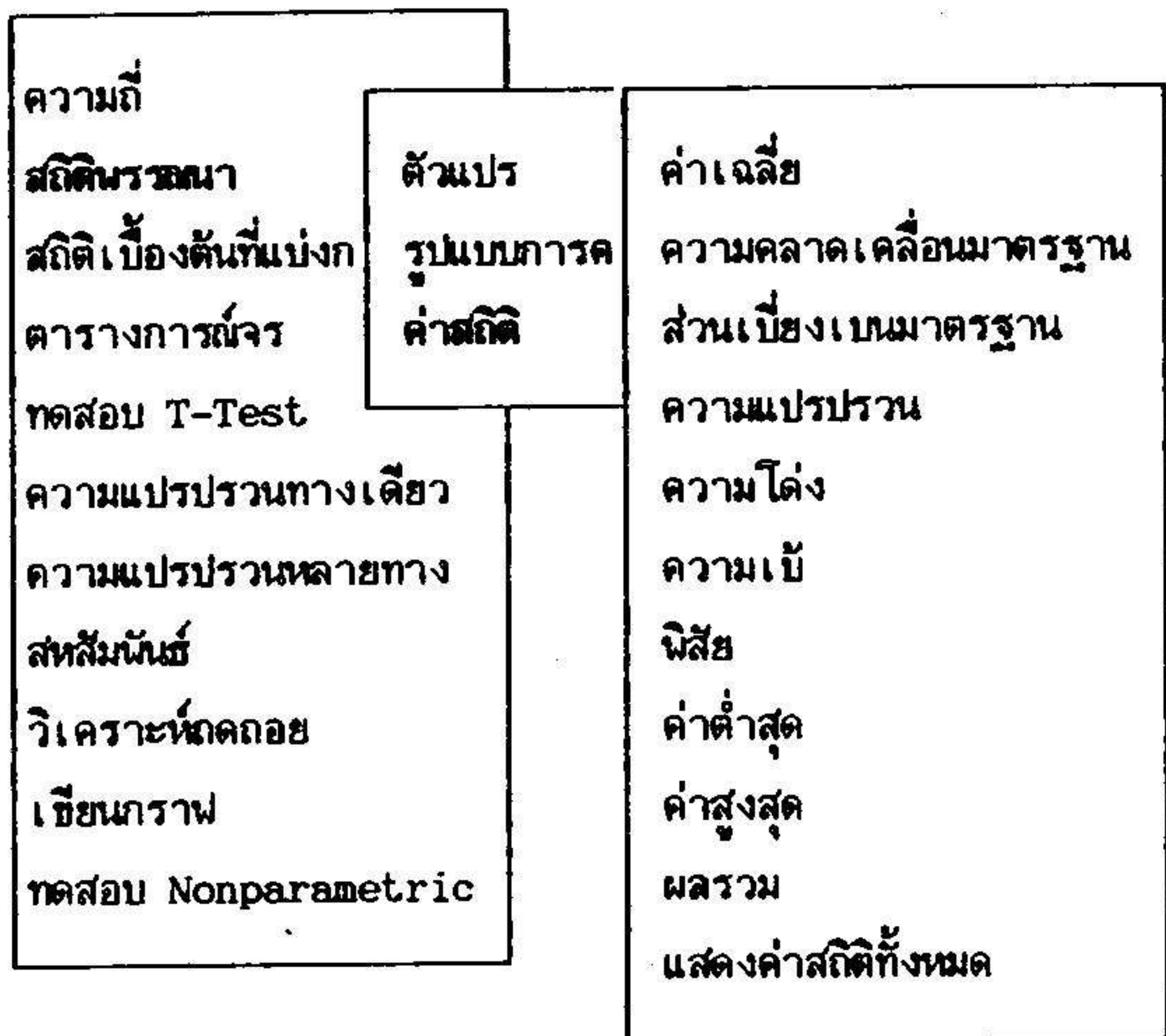


จอภาพ 2.2.2 รูปแบบของการคำนวณ

จากจอภาพ 2.2.2 เมื่อเลือกรายการรูปแบบการคำนวณได้โดยใช้ลูกศรเลื่อน เมื่อเลือกได้ข้อความที่ต้องการกด Enter โดยให้ใส่ข้อมูลของ

- รวมค่า ไม่สมบูรณ์
- ไม่พิมพ์ค่าขยายตัวแปร
- ไม่รวมค่า ไม่สมบูรณ์
- พิมพ์ค่าสถิติได้ชื่อตัวแปร
- ไม่พิมพ์ชื่อตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------



Esc Menu

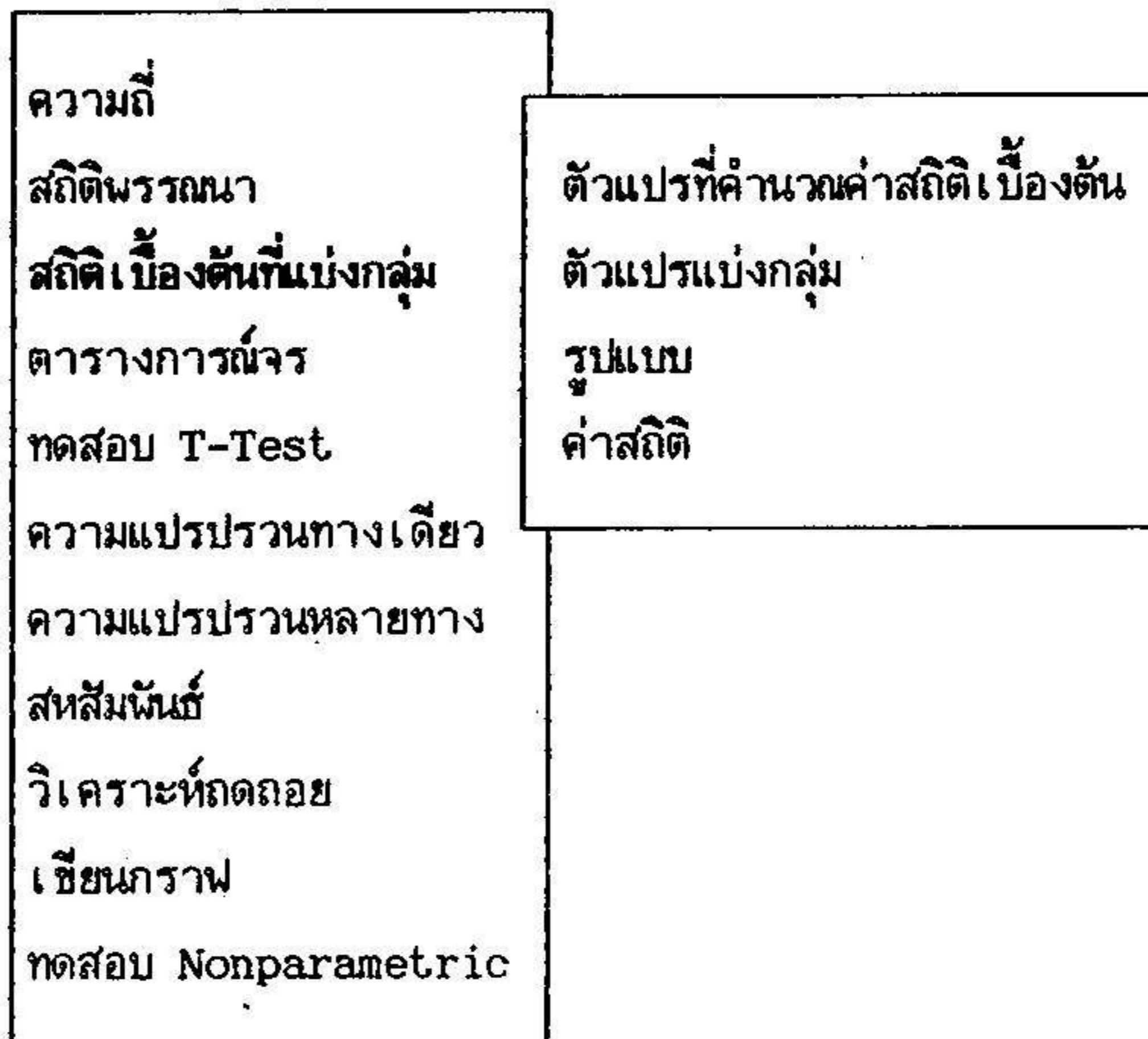
จอภาพ 2.2.3 ค่าสถิติในสถิติพรรณนา

จากจอภาพ 2.2.3 เมื่อเลือกรายการค่าสถิติของสถิติพรรณนาแล้วกด Enter จะปรากฏรายการให้เลือก คือ

- ค่าเฉลี่ย
- ความคลาดเคลื่อนมาตรฐาน
- ส่วนเบี่ยงเบนมาตรฐาน

- ความแปรปรวน
- ความโค้ง
- ความเบ้
- นิสัย
- ค่าต่ำสุด
- ค่าสูงสุด
- ผลรวม
- หาค่าสถิติทั้งหมดของสถิติข้างบนนี้

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------



F1 Help Esc Menu

จอภาพ 2.3 สถิติเบื้องต้นที่ต้องการแบ่งกลุ่ม

จากจอภาพ 2.3 เมื่อเลือกรายการสถิติเบื้องต้นแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- ตัวแปรที่คำนวณค่าสถิติเบื้องต้น : จะปรากฏเป็นจอภาพ 2.3.1
- ตัวแปรแบ่งกลุ่ม : จะปรากฏเป็นจอภาพ 2.3.2
- รูปแบบ : จะปรากฏเป็นจอภาพ 2.3.3
- ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.3.4

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric

ชื่อตัวแปร <input type="text"/>

Esc Menu

จอภาพ 2.3.1 ตัวแปรที่คำนวณค่าสถิติเบื้องต้น

จากจอภาพ 2.3.1 เมื่อเลือกรายการสถิติเบื้องต้นที่แบ่งกลุ่มแล้วใส่ข้อมูลได้ดังนี้
 ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการคำนวณค่าสถิติเบื้องต้น

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ตัวแปรแบ่งกลุ่ม
 ชื่อตัวแปร

Esc Menu

จอภาพ 2.3.2 ตัวแปรแบ่งกลุ่ม

จากจอภาพ 2.3.2 เมื่อเลือกรายการใส่ชื่อตัวแปรแล้วให้ใส่ชื่อตัวแปรที่ต้องการแบ่งกลุ่ม
ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่เป็นตัวแปรอิสระหรือตัวแปรแบ่งกลุ่ม

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประ โยชน์
--------	----------------	------------	----------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งก ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทาง ความแปรปรวนหลาย สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparam	รวมค่า ไม่สมบูรณ์ ตัดค่า ไม่สมบูรณ์ของตัวอย่าง พินิจผลรวมของกลุ่มย่อย ไม่พินิจค่าส่วนเบี่ยงเบนมาตรฐานของกลุ่มย่อย ไม่พินิจส่วนขยายของค่าตัวแปร ไม่พินิจชื่อตัวแปรแบ่งกลุ่ม ไม่พินิจค่าตัวแปรแบ่งกลุ่ม ไม่พินิจค่าของค่าเฉลี่ยในแต่ละกลุ่มย่อย พินิจค่าความแปรปรวนในแต่ละกลุ่มย่อย
--	---

Esc Menu

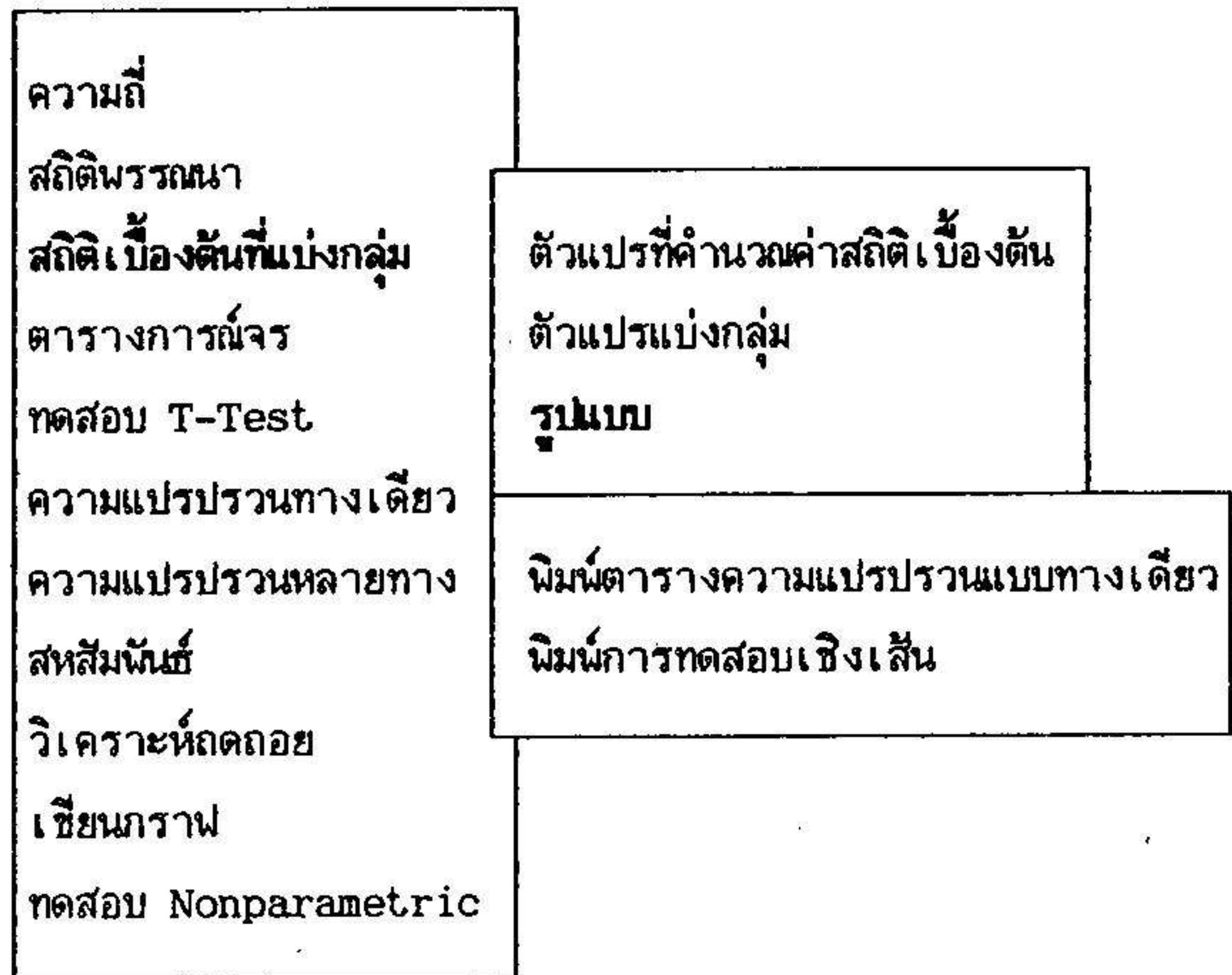
จอภาพ 2.3.3 รูปแบบในสถิติเบื้องต้น

จากจอภาพ 2.3.3 เมื่อเลือกรายการสถิติเบื้องต้นแล้วใช้ลูกศรเลื่อน กด Enter เพื่อเลือกรายการต่อไปนี้

- รวมค่า ไม่สมบูรณ์
- ตัดค่า ไม่สมบูรณ์ของตัวอย่าง
- พินิจผลรวมของกลุ่มย่อย
- ไม่พินิจค่าส่วนเบี่ยงเบนมาตรฐานของกลุ่มย่อย
- ไม่พินิจส่วนขยายของค่าตัวแปร
- ไม่พินิจชื่อตัวแปรแบ่งกลุ่ม
- ไม่พินิจค่าตัวแปรแบ่งกลุ่ม

- ไม่นิพนธ์ค่าของค่าเฉลี่ยในแต่ละกลุ่มย่อย
- นิพนธ์ค่าความแปรปรวนในแต่ละกลุ่มย่อย

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

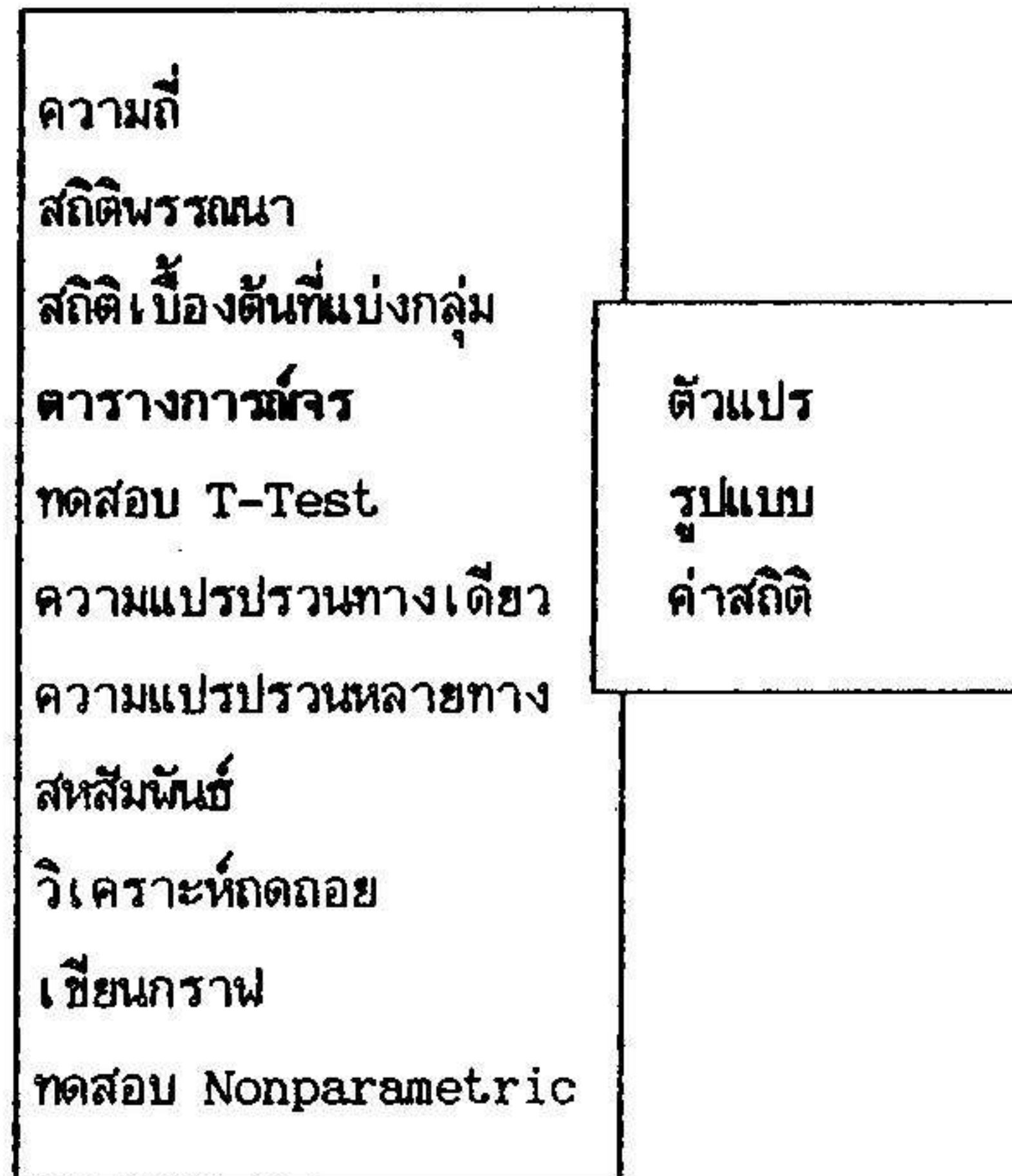


Esc Menu

จอภาพ 2.3.4 ค่าสถิติในสถิติเบื้องต้นที่แบ่งกลุ่ม

จากจอภาพ 2.3.4 เมื่อเลือกรายการสถิติเบื้องต้นที่แบ่งกลุ่มแล้วได้โดยใช้ลูกศรเลื่อนเมื่อเลือกได้ ข้อความที่ต้องการ กด Enter และสามารถเลือกรายการต่อไปได้โดย กด Enter เช่นกัน

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



F1 Help Esc Menu

จอภาพ 2.4 ตารางการแจกแจง

- จากจอภาพ 2.4 เมื่อเลือกรายการรูปแบบแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการต่อได้ดังนี้
- ตัวแปร : จะปรากฏเป็นจอภาพ 2.4.1
 - รูปแบบ : จะปรากฏเป็นจอภาพ 2.4.2
 - ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.4.3

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

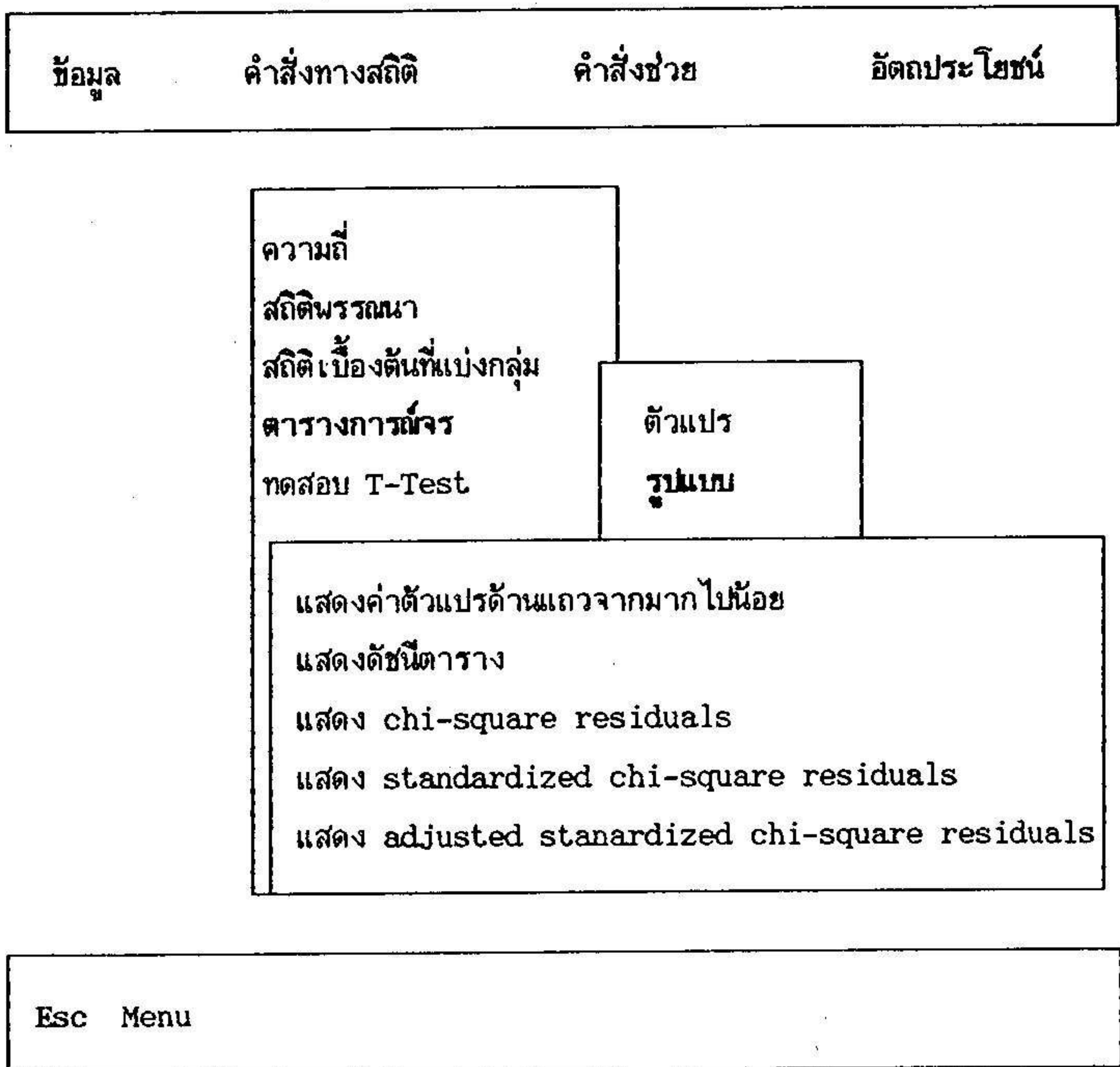
ชื่อตัวแปร

Esc Menu

จอภาพ 2.4.1 ตัวแปรในตารางการแจกแจง

จากจอภาพ 2.4.1 เมื่อเลือกรายการตารางการแจกแจงแล้วให้ใส่ข้อมูลดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการสร้างตารางแจกแจงความถี่



จอภาพ 2.4.2 รูปแบบในตารางการถ่วง

- จากจอภาพ 2.4.2 เมื่อเลือกรายการรูปแบบของตารางการถ่วง แล้วใช้ลูกศรเลื่อน เมื่อเลือกได้รายการที่ต้องการแล้วกด Enter และสามารถเลือกรายการต่อไปได้โดยกด Enter เช่นกัน รายการที่ให้เลือก คือ
- แสดงค่าตัวแปรด้านแถวจากมากไปน้อย
 - แสดงดัชนีตาราง
 - แสดง Chi-square Residuals
 - แสดง Standardized Chi-square Residuals
 - แสดง Adjusted Stanardized Chi-square Residuals

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตถประโยชน์
--------	----------------	------------	--------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test	ตัวแปร รูปแบบ
Chi-square Phi for 2x2 table, Carmer's V for large table Contingency coefficient Lambda Uncertainty coefficient Kendall's tau-b Kendall's tau-c Gamma Somer'd Eta Pearson's r	

Esc Menu

จอภาพ 2.4.3 คำสถิติในตารางการแจกแจง

จากจอภาพ 2.4.3 เมื่อเลือกรายการสถิติของตารางการแจกแจง แล้วใช้ลูกศรเลื่อน เมื่อเลือกได้รายการที่ต้องการแล้ว กด Enter และสามารถเลือกรายการต่อไปได้โดยกด Enter เช่นกัน

รายการที่ให้เลือก คือ

Chi-square

Phi for 2x2 table, Carmer's V for large table

Contingency coefficient

Lambda

Uncertainty coefficient

Kendall's tau-b

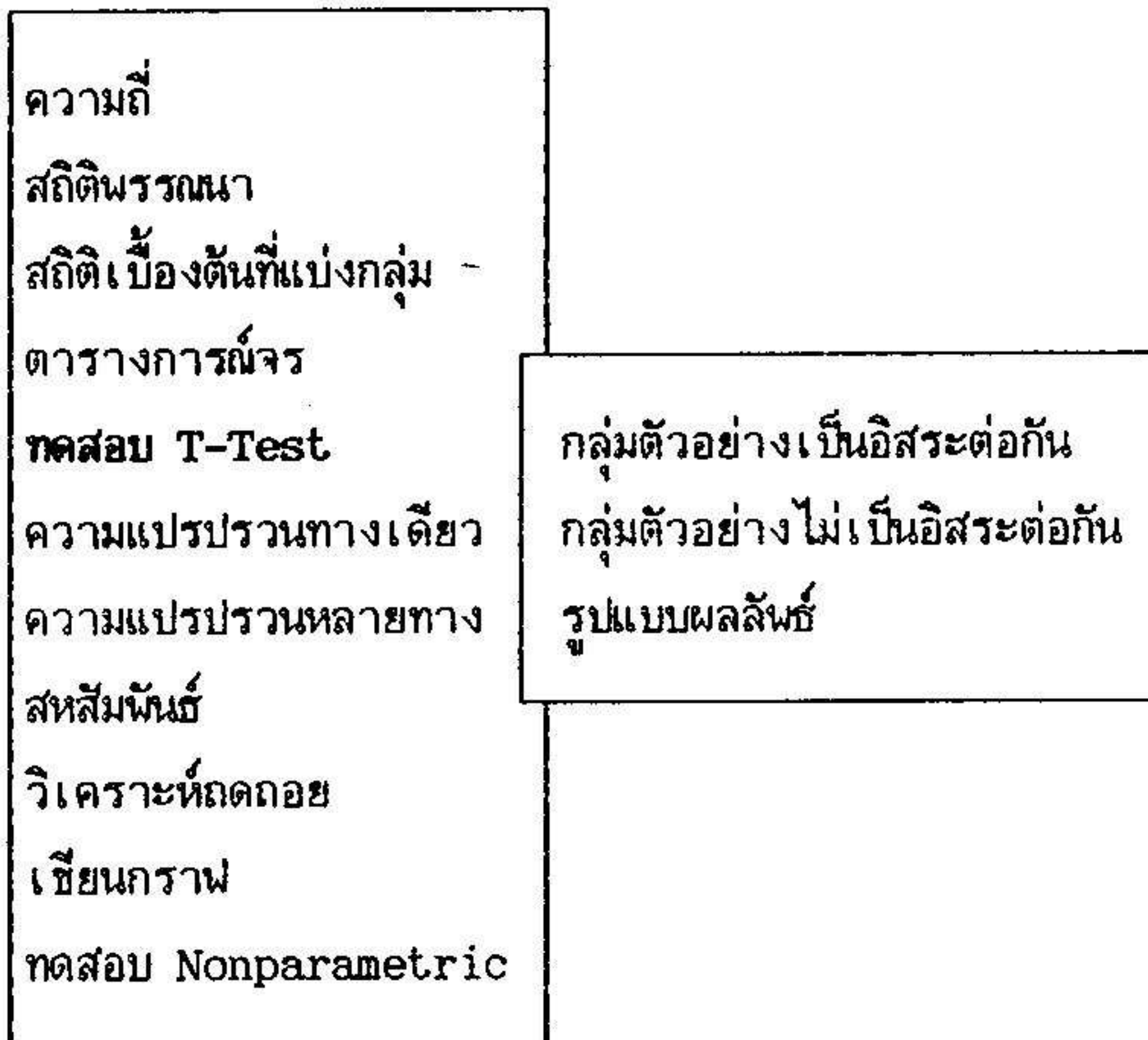
Kendall's tau-c

Gamma

Somer'd

Eta และ Pearson's r

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

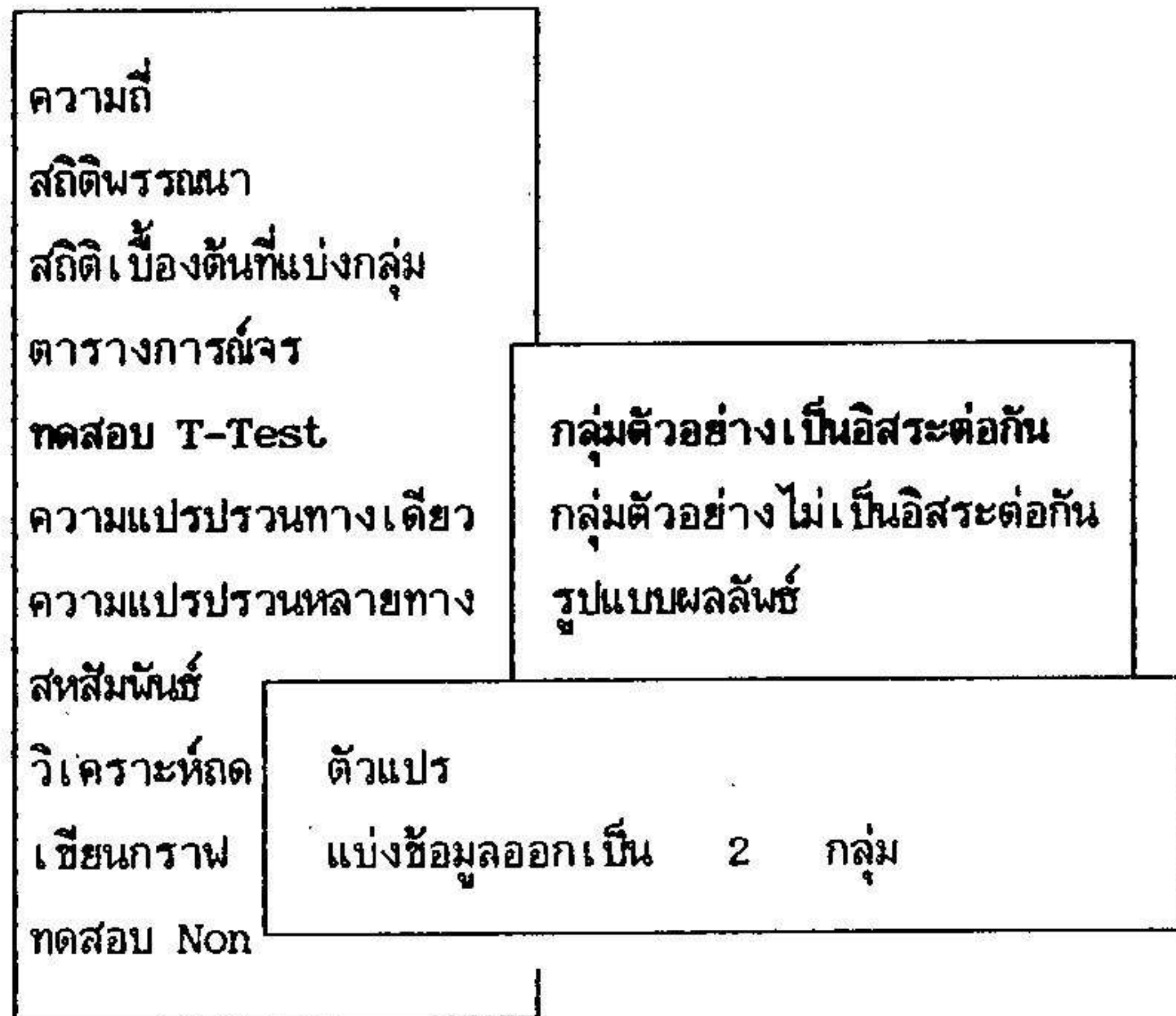


F1 Help Esc Menu

จากจอภาพ 2.5 เมื่อเลือกรายการทดสอบ T-test แล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการ
ได้ดังนี้

- กลุ่มตัวอย่างเป็นอิสระต่อกัน : จะปรากฏเป็นจอภาพ 2.5.1
- กลุ่มตัวอย่างไม่เป็นอิสระต่อกัน : จะปรากฏเป็นจอภาพ 2.5.2
- รูปแบบผลลัพธ์ : จะปรากฏเป็นจอภาพ 2.5.3

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

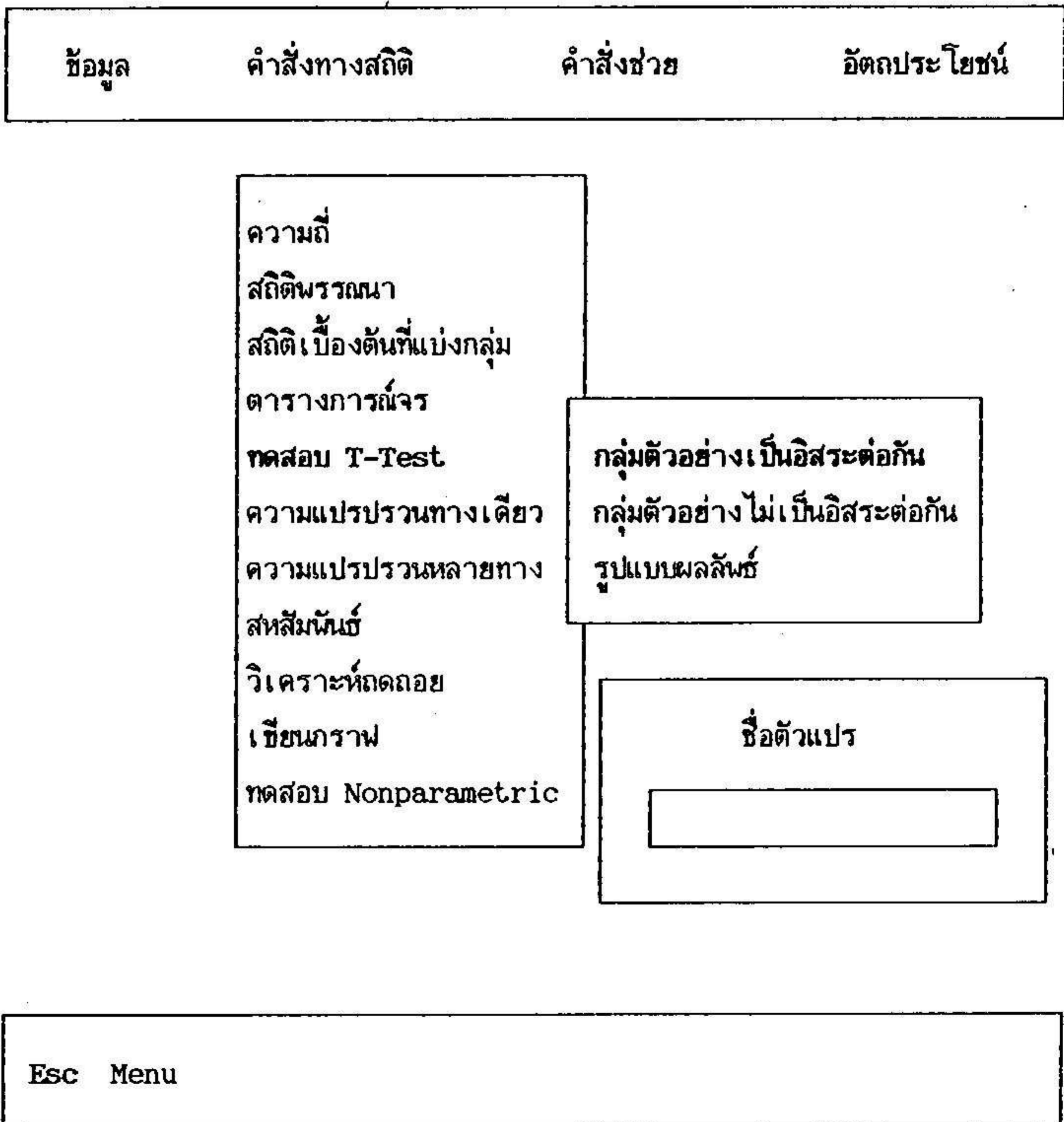


Esc Menu

จอภาพ 2.5.1 กลุ่มตัวอย่างเป็นอิสระต่อกัน

จากจอภาพ 2.5.1 เมื่อเลือกรายการกลุ่มตัวอย่างเป็นอิสระต่อกันแล้วใช้ลูกศรเลื่อนเพื่อ
รายการและใส่ข้อมูลของ

- ตัวแปร : จะปรากฏเป็นจอภาพ 2.5.1.1
- แบ่งข้อมูลเป็นสองกลุ่ม : จะปรากฏเป็นจอภาพ 2.5.1.2



จอภาพ 2.5.1.1 ตัวแปรในการทดสอบ T-Test

จากจอภาพ 2.5.1.2 เมื่อเลือกรายการกลุ่มตัวอย่างเป็นอิสระต่อกันของการทดสอบ T-test แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการนำมาแบ่งกลุ่มข้อมูล
 กลุ่มที่ 1 มีค่า : ใส่ค่าของตัวแปรที่ต้องการนำมาแบ่งกลุ่ม
 กลุ่มที่ 2 มีค่า : ใส่ค่าของตัวแปรที่ต้องการนำมาแบ่งกลุ่ม

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

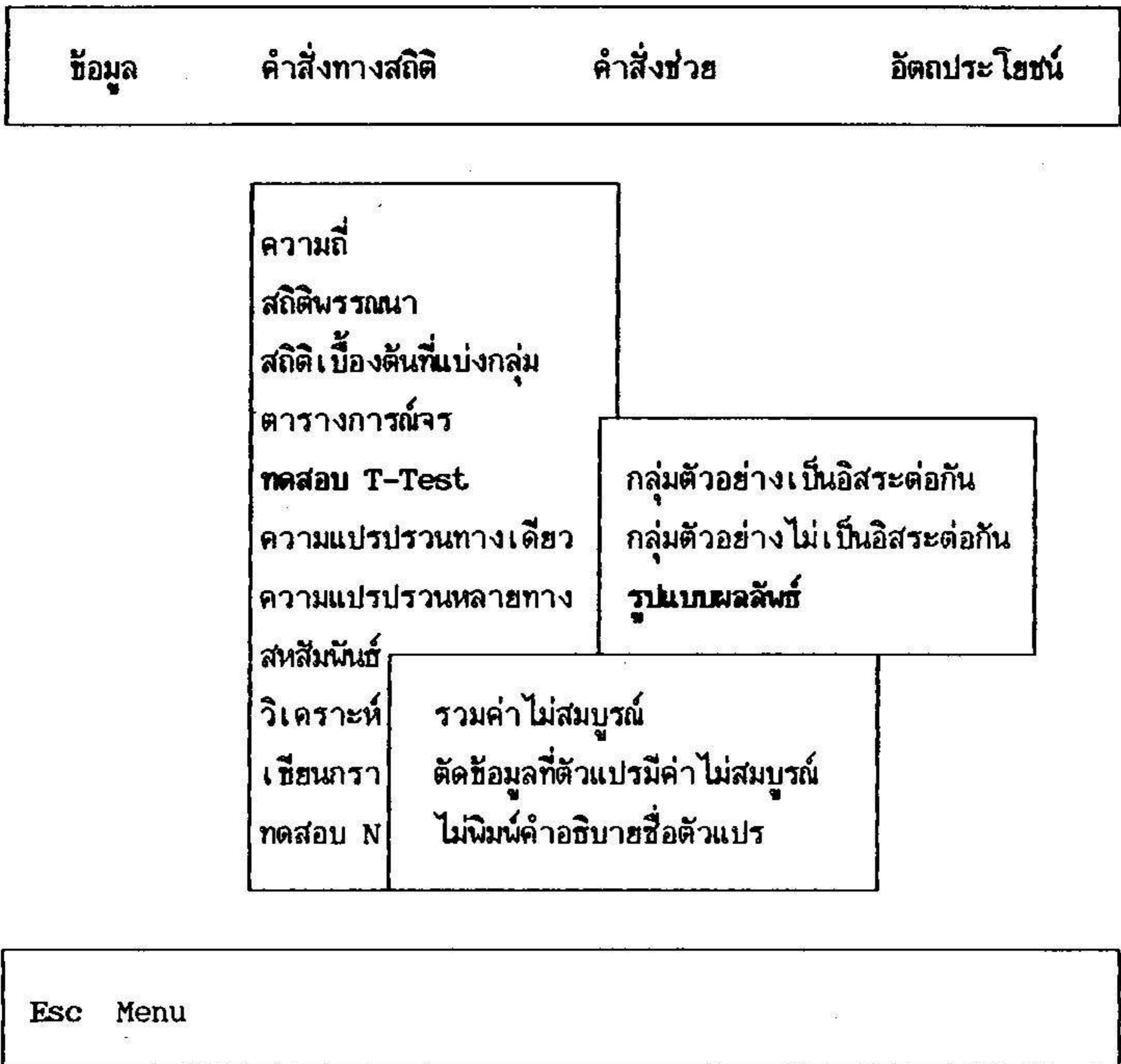
กลุ่มตัวอย่างไม่เป็นอิสระต่อกัน
 ชื่อตัวแปร

Esc Menu

จอภาพ 2.5.2 กลุ่มตัวอย่างไม่เป็นอิสระต่อกันของการทดสอบ

จากจอภาพ 2.5.2 เมื่อเลือกรายการกลุ่มตัวอย่างไม่เป็นอิสระต่อกันของการทดสอบ T-test แล้วให้ใส่ข้อมูลได้ดังนี้สามารถใส่ข้อความได้ดังนี้

ชื่อตัวแปร : ชื่อตัวแปรที่ต้องการทดสอบค่าเฉลี่ย



จอภาพ 2.5.3 รูปแบบผลลัพธ์ของการทดสอบ T-Test

จากจอภาพ 2.5.3 เมื่อเลือกรายการรูปแบบผลลัพธ์ของการทดสอบ T-test แล้วใช้ลูกศรเลื่อน เมื่อเลือกได้รายการที่ต้องการแล้วกด Enter และสามารถเลือกรายการต่อไปได้โดยกด Enter เช่นกัน รายการที่ให้เลือก คือ

- รวมค่าไม่สมบูรณ์
- ตัดข้อมูลที่ตัวแปรที่มีค่าไม่สมบูรณ์
- ไม่พิมพ์ค่าอธิบายชื่อตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
	ความถี่ สถิติพรรณนา สถิติเบื้องต้นทั้งแบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทาง ความแปรปรวนหลาย สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ สถิติไพบรณนา	ตัวแปรตาม ตัวแปรแบ่งกลุ่ม กำหนดการแบ่งกลุ่มของตัวแปรผันแปร เปรียบเทียบ, ทดสอบค่าเฉลี่ยในรูปเวกเตอร์ กำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ย รูปแบบ ค่าสถิติ	

F1 Help Esc Menu

จอภาพ 2.6 ความแปรปรวนทางเดียว

จากจอภาพ 2.6 เมื่อเลือกรายการความแปรปรวนทางเดียวแล้วเลือกรายการได้ดังนี้

- ตัวแปรตาม : จะปรากฏเป็นจอภาพ 2.6.1
- ตัวแปรแบ่งกลุ่ม : จะปรากฏเป็นจอภาพ 2.6.2
- กำหนดการแบ่งกลุ่มของตัวแปรผันแปร : จะปรากฏเป็นจอภาพ 2.6.3
- เปรียบเทียบ, ทดสอบค่าเฉลี่ยในรูปเวกเตอร์ : จะปรากฏเป็นจอภาพ 2.6.4
- กำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ย : จะปรากฏเป็นจอภาพ 2.6.5
- รูปแบบ : จะปรากฏเป็นจอภาพ 2.6.6
- ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.6.7

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ตัวแปรตาม

ชื่อตัวแปร

Esc Menu

จอภาพ 2.6.1 ตัวแปรตาม

จากจอภาพ 2.6.1 เมื่อเลือกรายการความแปรปรวนทางเดียวแล้วให้ใส่ข้อมูลได้ดังนี้
 ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการวิเคราะห์ความแปรปรวนเพื่อ
 เปรียบเทียบ และ ทดสอบค่าเฉลี่ย

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
	<div style="border: 1px solid black; padding: 5px;"> <p>ความถี่</p> <p>สถิติพรรณนา</p> <p>สถิติเบื้องต้นที่แบ่งกลุ่ม</p> <p>ตารางการแจกแจง</p> <p>ทดสอบ T-Test</p> <p>ความแปรปรวนทางเดียว</p> <p>ความแปรปรวนหลายทาง</p> <p>สหสัมพันธ์</p> <p>เขียนกราฟ</p> <p>ทดสอบ Nonparametric</p> </div>		
			<div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>ตัวแปรแบ่งกลุ่ม</p> <p>ชื่อตัวแปร</p> <div style="border: 1px solid black; height: 20px; width: 80%; margin: 5px auto;"></div> <p>ค่าต่ำสุด [] ค่าสูงสุด []</p> </div>
<div style="border: 1px solid black; padding: 5px;"> <p>Esc Menu</p> </div>			

จอภาพ 2.6.2 ตัวแปรแบ่งกลุ่มในความแปรปรวนทางเดียว

จากจอภาพ 2.6.2 เมื่อเลือกรายการตัวแปรแบ่งกลุ่มแล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร

: ใส่ชื่อตัวแปรที่ต้องการกำหนดให้เป็นตัวแปรแบ่งกลุ่ม
และระบุค่าสูงสุดและต่ำสุดด้วย

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประ โยชน์
--------	----------------	------------	----------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

กำหนดการแบ่งกลุ่มของแหล่งผันแปร []

Esc Menu

จอภาพ 2.6.3 การกำหนดการแบ่งกลุ่มของแหล่งผันแปรในความแปรปรวนทางเดียว

จากจอภาพ 2.6.3 เมื่อเลือกรายการกำหนดการแบ่งกลุ่มของแหล่งผันแปรแล้วใส่ข้อมูล
ได้ดังนี้

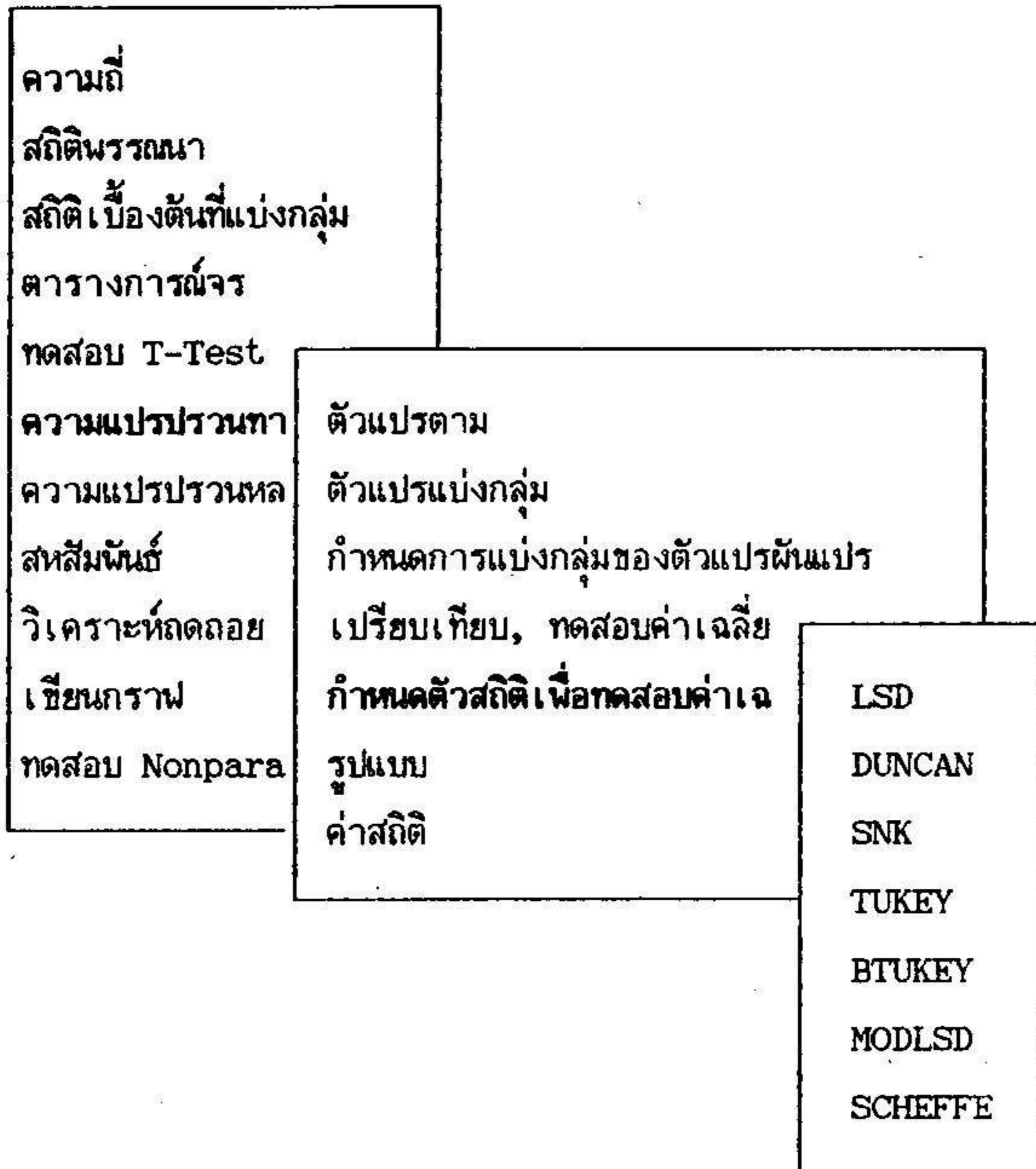
กำหนดการแบ่งกลุ่มของแหล่งผันแปร : ใส่ตัวเลขแทนจำนวนที่ต้องการแบ่งกลุ่ม

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
	ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric		เปรียบเทียบและทดสอบ ค่าเฉลี่ยในรูปเวกเตอร์ []
Esc Menu			

จอภาพ 2.6.4 การเปรียบเทียบและทดสอบ ค่าเฉลี่ยในรูปเวกเตอร์ในความแปรปรวนทางเดียว

จากจอภาพ 2.6.4 เมื่อเลือกรายการเปรียบเทียบและทดสอบ ค่าเฉลี่ยในรูปเวกเตอร์ แล้วใส่ข้อมูล ได้ดังนี้
 เปรียบเทียบ, ทดสอบค่าเฉลี่ยในรูปเวกเตอร์ :

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



Esc Menu

จอภาพ 2.6.5 กำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ย

จากจอภาพ 2.6.5 เมื่อเลือกรายการกำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ยแล้วใช้ลูกศรเลื่อนเมื่อเลือกได้รายการที่ต้องการ กด Enter และเลือกรายการต่อไปได้โดย กด Enter เช่นกัน

รายการสถิติที่ให้เลือกคือ

LSD

DUNCAN

SNK

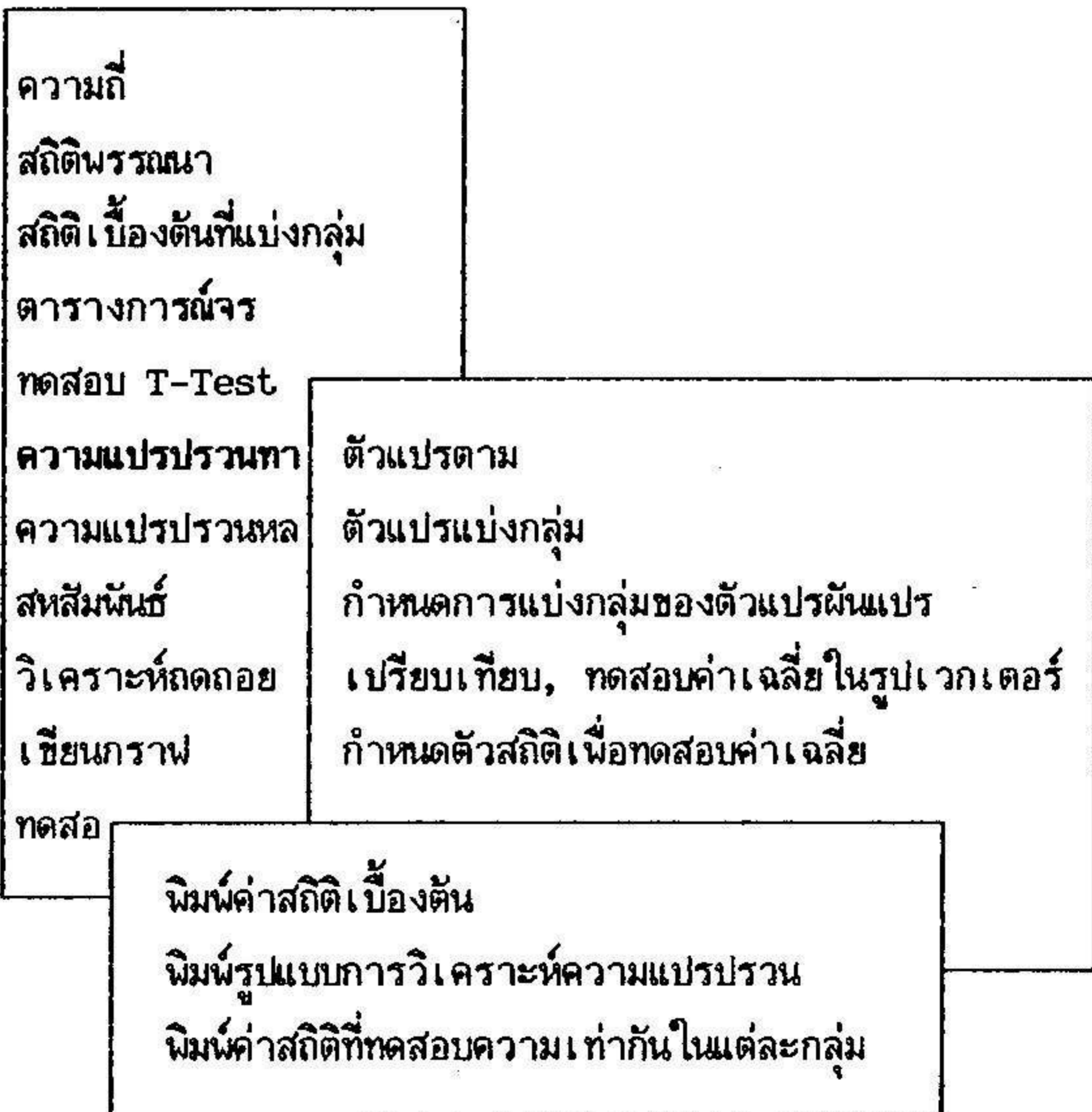
TUKEY

BTUKEY

MODLSD

SCHEFFE

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



Esc Menu

จากจอภาพ 2.6.6 เมื่อเลือกรายการรูปแบบของการวิเคราะห์ความแปรปรวนแล้วใช้ลูกศรเลื่อน เมื่อเลือกรายการที่ต้องการ กด Enter และเลือกรายการต่อไปโดยกด Enter เช่นกัน

รายการที่ให้เลือก คือ

พิมพ์ค่าสถิติเบื้องต้น

พิมพ์รูปแบบการวิเคราะห์ความแปรปรวน

พิมพ์ค่าสถิติที่ทดสอบความเท่ากันในแต่ละกลุ่ม

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
สถิติพรรณนา
สถิติเบื้องต้นที่แบ่งกลุ่ม
ตารางการแจกแจง
ทดสอบ T-Test

ความแปรปรวนทาง
ความแปรปรวนหลาย
สหสัมพันธ์
วิเคราะห์ถดถอย
เขียนกราฟ
ทดสอบ

ตัวแปรตาม
ตัวแปรแบ่งกลุ่ม
กำหนดการแบ่งกลุ่มของตัวแปรต้นแปร
เปรียบเทียบ, ทดสอบค่าเฉลี่ยในรูปเวกเตอร์
กำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ย

รวมค่า ไม่สมบูรณ์
ตัดข้อมูลที่มีค่า ไม่สมบูรณ์
ไม่พิมพ์ส่วนขยายและตัวแปร
พิมพ์ค่าสถิติเบื้องต้นในรูปเวกเตอร์

Esc Menu

จอภาพ 2.6.7 คำสถิติของการวิเคราะห์ความแปรปรวนทางเดียว

จากจอภาพ 2.6.7 เมื่อเลือกรายการค่าสถิติของการวิเคราะห์ความแปรปรวนทางเดียว โดยใช้ลูกศรเลื่อน เมื่อเลือกรายการที่ต้องการ กด Enter และเลือกรายการต่อไปได้โดยกด Enter เช่นกัน

รายการที่ให้เลือก คือ

- รวมค่า ไม่สมบูรณ์
- ตัดข้อมูลที่มีค่า ไม่สมบูรณ์
- ไม่พิมพ์ส่วนขยายและตัวแปร
- พิมพ์ค่าสถิติเบื้องต้นในรูปแบบเวกเตอร์

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric	ตัวแปรที่วิเคราะห์ความแปรปรวน ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่ม ชื่อตัวแปรที่เป็นตัวแปรร่วม รูปแบบการวิเคราะห์ความแปรปรวน ค่าสถิติ
---	--

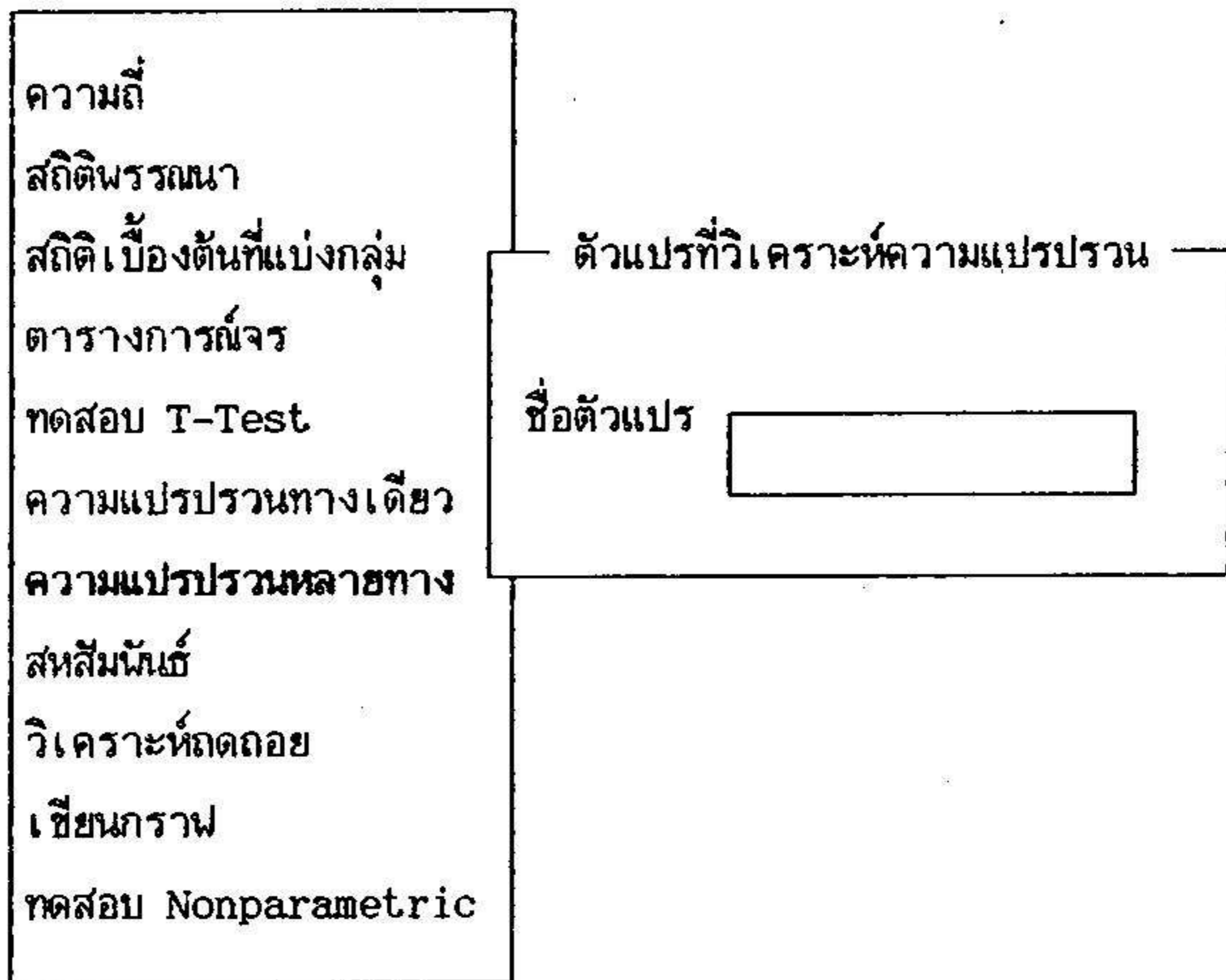
F1 Help Esc Menu

จอภาพ 2.7 ความแปรปรวนหลายทาง

จากจอภาพ 2.7 เมื่อเลือกรายการความแปรปรวนหลายทางแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- ตัวแปรที่วิเคราะห์ความแปรปรวน : จะปรากฏเป็นจอภาพ 2.7.1
- ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่ม : จะปรากฏเป็นจอภาพ 2.7.2
- ชื่อตัวแปรที่เป็นตัวแปรร่วม : จะปรากฏเป็นจอภาพ 2.7.3
- รูปแบบการวิเคราะห์ความแปรปรวน : จะปรากฏเป็นจอภาพ 2.7.4
- ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.7.5

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



Esc Menu

จอภาพ 2.7.1 ตัวแปรที่วิเคราะห์ความแปรปรวนหลายทาง

จากจอภาพ 2.7.1 เมื่อเลือกรายการตัวแปรที่วิเคราะห์ความแปรปรวนแล้วใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการวิเคราะห์ความแปรปรวนแบบหลายทาง

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric	ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่ม ชื่อตัวแปร <input type="text"/> ค่าต่ำสุด [] ค่าสูงสุด []
---	--

Esc Menu

จอภาพ 2.7.2 ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่มของความแปรปรวนหลายทาง

จากจอภาพ 2.7.2 เมื่อเลือกรายการ ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่มแล้วให้ใส่ข้อมูล
ได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการแบ่งกลุ่มพร้อมระบุค่าสูงสุดและ
ต่ำสุด

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ตัวแปรที่เป็นตัวแปรร่วม

ชื่อตัวแปร

Esc Menu

จอภาพ 2.7.3 ตัวแปรที่เป็นตัวแปรร่วมของความแปรปรวนหลายทาง

จากจอภาพ 2.7.3 เมื่อเลือกรายการตัวแปรที่เป็นตัวแปรร่วมแล้วใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการกำหนดให้เป็นตัวแปรร่วม
สำหรับการวิเคราะห์ความแปรปรวนร่วม

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
	ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลาย สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparame	รูปแบบวิเคราะห์ความแปรปรวน	รวมข้อมูลที่มีค่า ไม่สมบูรณ์ ไม่นิพนธ์ผลจากผลรวมระหว่างตัวแปร ไม่นิพนธ์ผลจากผลรวมของ 3 ตัวแปรขึ้นไป ไม่นิพนธ์ผลจากผลรวมของ 4 ตัวแปรขึ้นไป ไม่นิพนธ์ผลจากผลรวมของ 5 ตัวแปรขึ้นไป นิพนธ์ผลกระทบบหลักกับผลจากตัวแปรพร้อมกัน นิพนธ์ผลกระทบบหลักก่อนนิพนธ์บผลจากตัวแปรร่วม นิพนธ์ผลวิเคราะห์การถดถอย
Esc Menu			

จอภาพ 2.7.4 คำสถิติของความแปรปรวนหลายทาง

จากจอภาพ 2.7.4 เมื่อเลือกรายการคำสถิติของความแปรปรวนหลายทางแล้วใช้ลูกศร เลื่อนเมื่อเลือกรายการที่ต้องการ กด Enter และ เลือกรายการต่อไปโดย กด Enter

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประ โยชน์
--------	----------------	------------	----------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric	ตัวแปรที่วิเคราะห์ความแปรปรวน ตัวแปรที่ให้ เป็นตัวแปรแบ่งกลุ่ม	นิพจน์ตารางการแจกแจงแบบหลายทาง นิพจน์สัมประสิทธิ์การถดถอย นิพจน์ค่าสถิติเบื้องต้น
---	---	---

Esc Menu

จอภาพ 2.7.5 รูปแบบการวิเคราะห์ความแปรปรวนหลายทาง

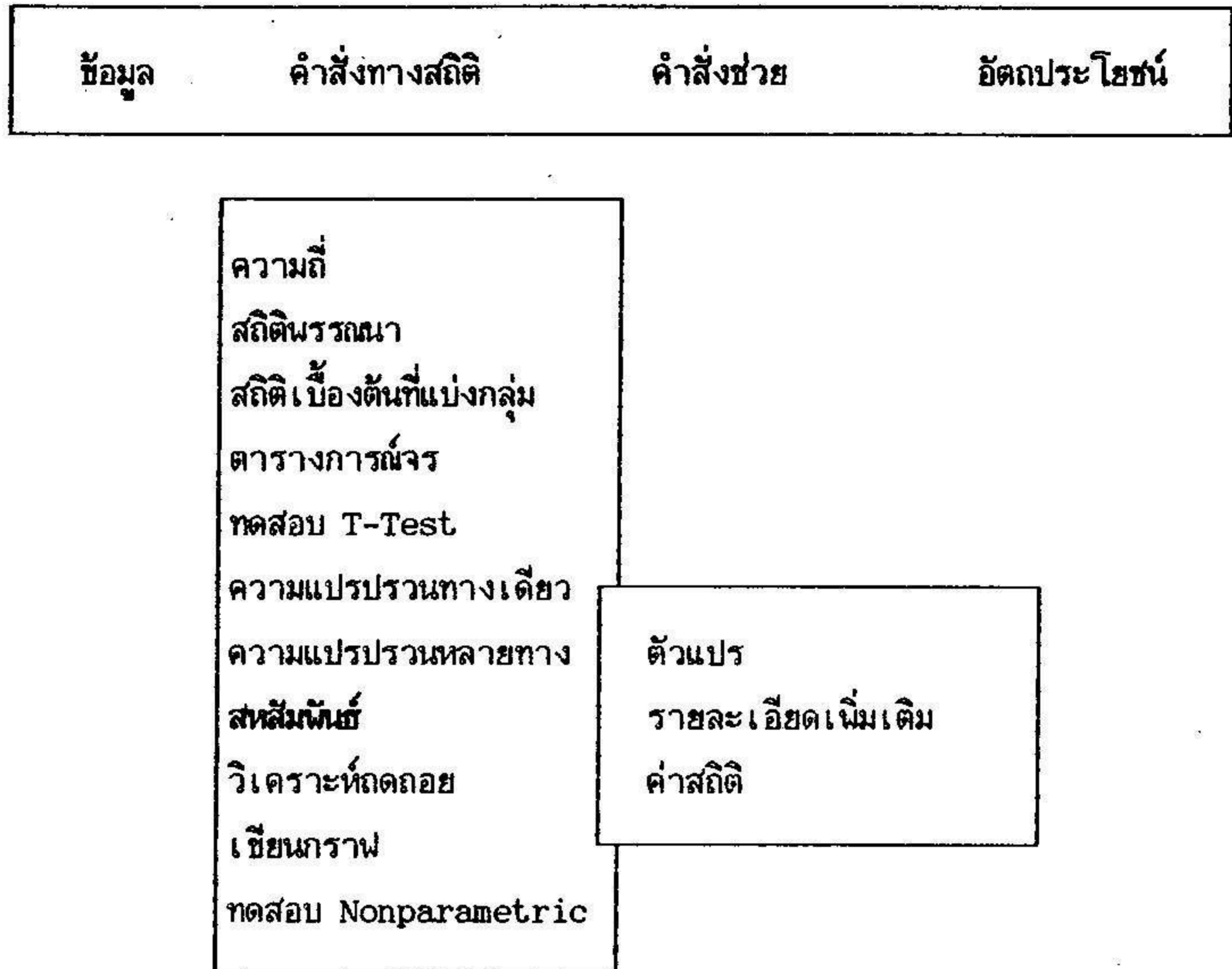
จากจอภาพ 2.7.5 เมื่อเลือกรูปแบบการวิเคราะห์ความแปรปรวนแล้ว ใช้ลูกศรเลื่อนและเมื่อเลือกรายการที่ต้องการ กด Enter และ เลือกรายการต่อไปได้โดย กด Enter เช่นกัน

รายการที่ให้เลือก คือ

นิพจน์ตารางการแจกแจงแบบหลายทาง

นิพจน์สัมประสิทธิ์การถดถอย

นิพจน์ค่าสถิติเบื้องต้น



จอภาพ 2.8 รายการสหสัมพันธ์

จากจอภาพ 2.8 เมื่อเลือกรายการสหสัมพันธ์แล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการดังนี้

- ตัวแปร : จะปรากฏจอภาพ 2.8.1
- รายละเอียดเพิ่มเติม : จะปรากฏจอภาพ 2.8.2
- คำสถิติ : จะปรากฏจอภาพ 2.8.3

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

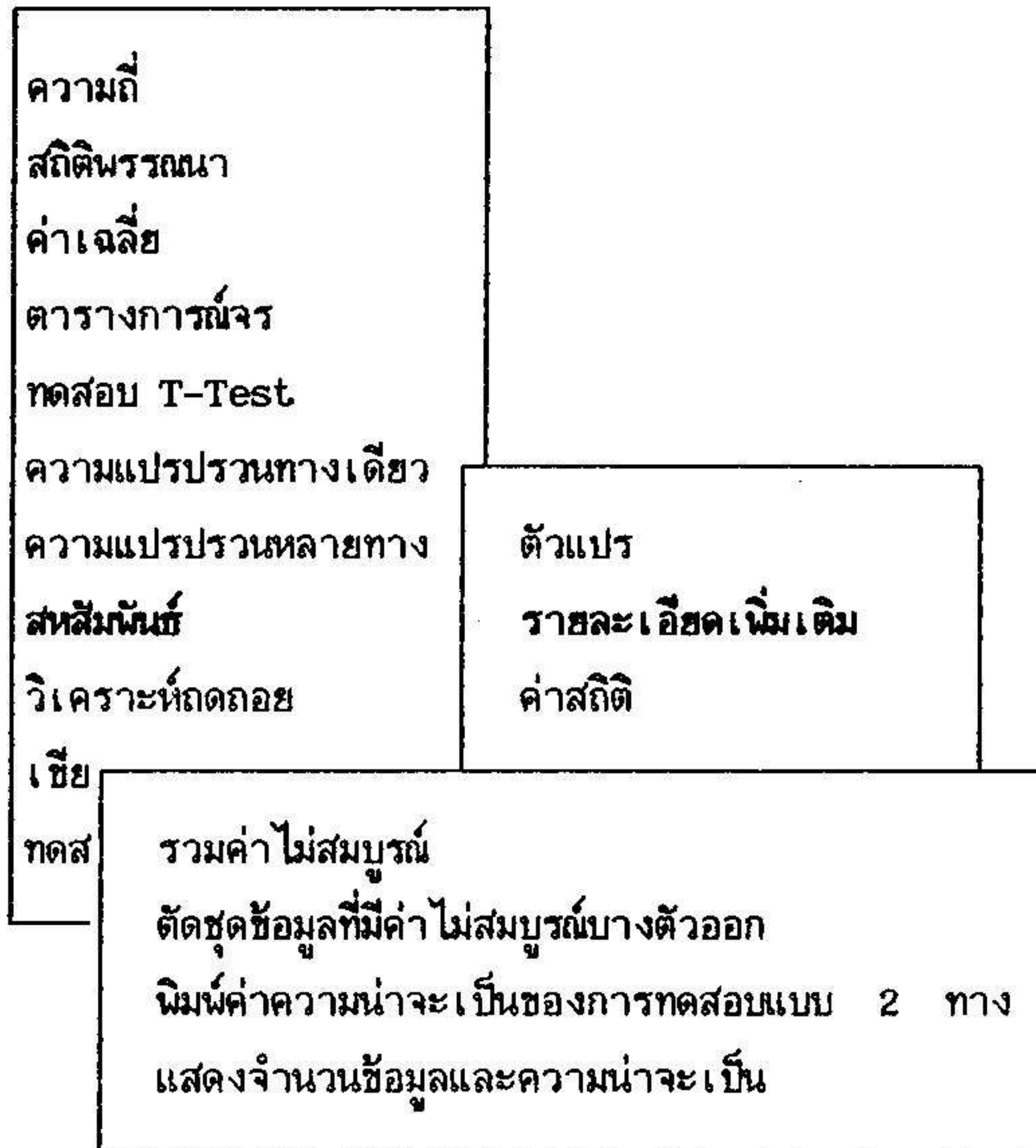
ตัวแปร	
หาความสัมพันธ์ ของตัวแปร	<input type="text"/>
กับตัวแปร	<input type="text"/>

จอภาพ 2.8.1 ตัวแปรของสหสัมพันธ์

จากจอภาพ 2.8.1 เมื่อเลือกรายการตัวแปรของสหสัมพันธ์แล้วใส่ข้อมูลได้ดังนี้

ตัวแปร : ชื่อของตัวแปรที่ต้องการหาค่าสัมประสิทธิ์สหสัมพันธ์

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

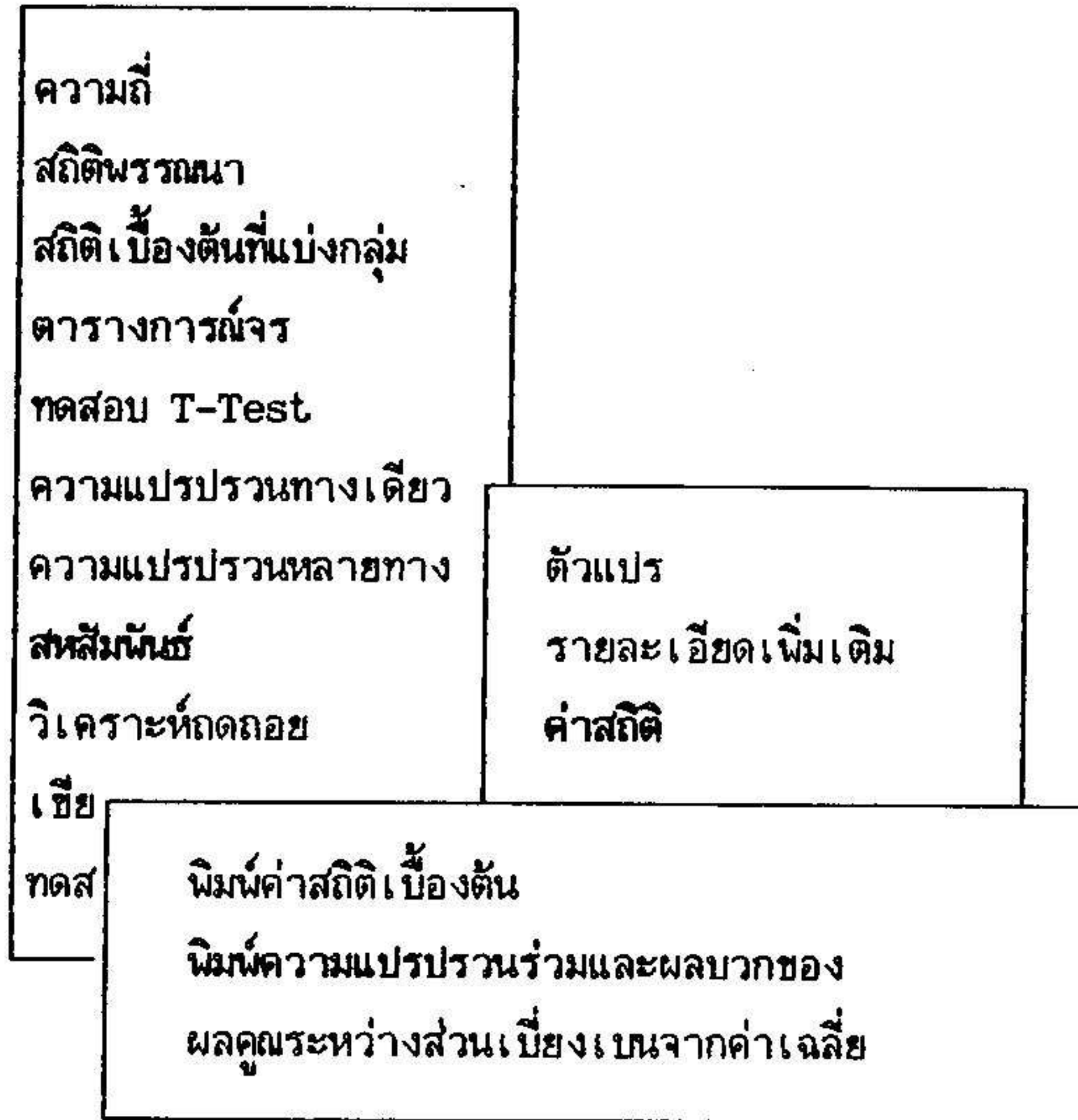


จอภาพ 2.8.2 รายละเอียดเพิ่มเติมในสหสัมพันธ์

จากจอภาพ 2.8.2 เมื่อเลือกรายการ รายละเอียดเพิ่มเติมแล้วใช้ลูกศรเลื่อน และเมื่อเลือกรายการที่ต้องการกด Enter และเลือกรายการต่อไปโดย กด Enter รายการที่ให้เลือก คือ

รวมค่าไม่สมบูรณ์
 ตัดชุดข้อมูลที่มีค่า ไม่สมบูรณ์บางตัวออก
 นิยมค่าความน่าจะเป็นของการทดสอบแบบ 2 ทาง
 แสดงจำนวนข้อมูลและความน่าจะเป็น

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

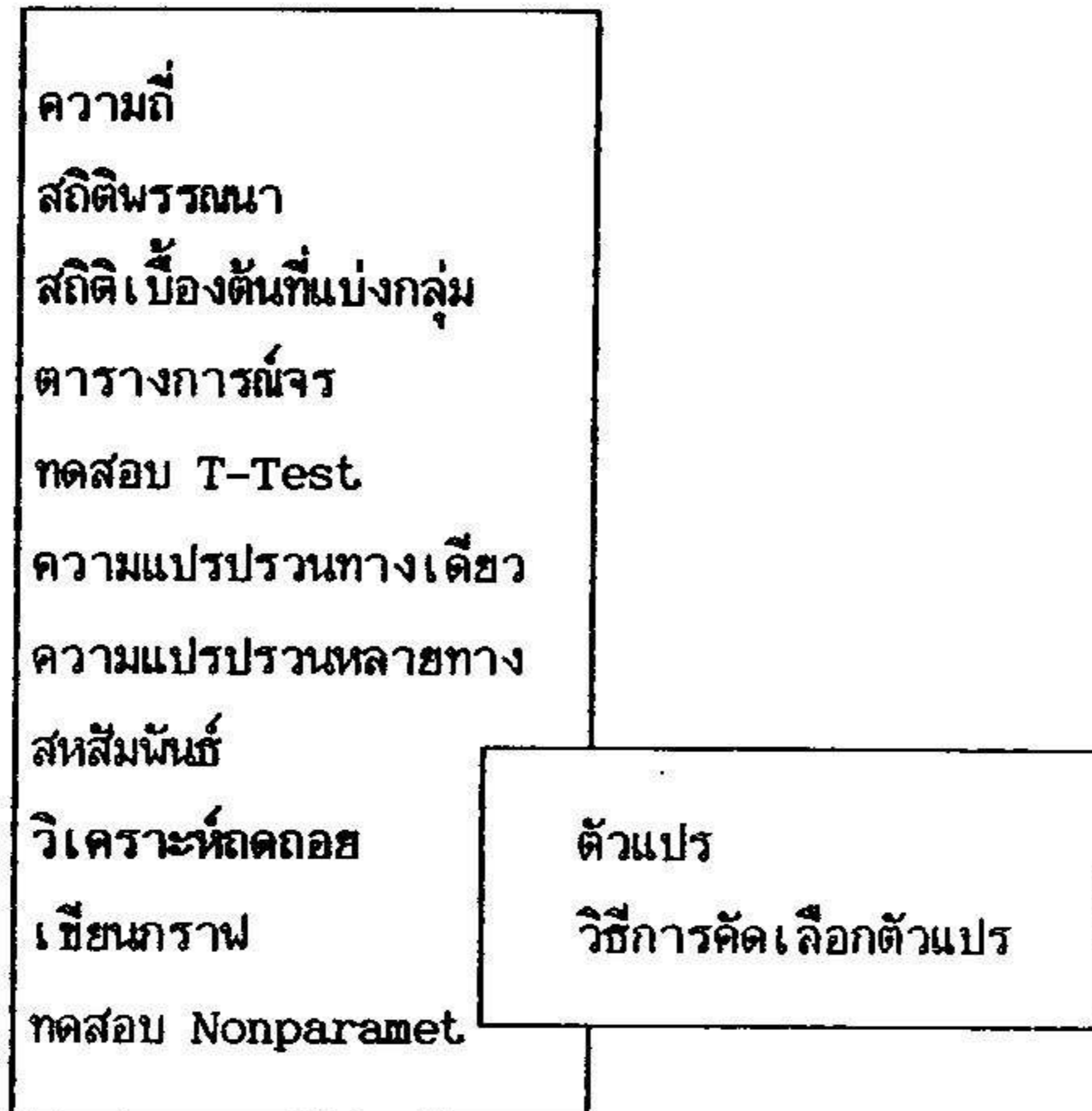


จอภาพ 2.8.3 คำสถิติในสหสัมพันธ์

จากจอภาพ 2.8.3 เมื่อเลือกรายการคำสถิติแล้วใช้ลูกศรเลื่อน เมื่อเลือกรายการที่ต้องการ กด Enter และเลือกรายการต่อไปได้โดย กด Enter รายการที่ให้เลือกคือ

- พิมพ์คำสั่งเบื้องต้น
- พิมพ์ความแปรปรวนร่วมและผลบวกของ
- ผลคูณระหว่างส่วนเบี่ยงเบนจากค่าเฉลี่ย

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	วัตถุประสงค์
--------	----------------	------------	--------------



จอภาพ 2.9 รายการวิเคราะห์ถดถอย

จากจอภาพ 2.9 เมื่อเลือกรายการวิเคราะห์ถดถอยแล้ว ใช้ลูกศรเลื่อนเพื่อเลือกรายการดังนี้

- ตัวแปร : จะปรากฏจอภาพ 2.9.1
- วิธีการคัดเลือกตัวแปร : จะปรากฏจอภาพ 2.9.2

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparametric	ตัวแปร ชื่อตัวแปร <input type="text"/> ตัวแปรตาม <input type="text"/>
---	---

จอภาพ 2.9.1 ตัวแปรของการวิเคราะห์ถดถอย

จากจอภาพ 2.9.1 เมื่อเลือกรายการตัวแปรของการวิเคราะห์ถดถอยแล้วใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ชื่อของตัวแปรที่ต้องการให้เป็นตัวแปรอิสระ

ตัวแปรตาม : ชื่อของตัวแปรที่ต้องการให้เป็นตัวแปรตาม

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประกอบ
	ความถี่ สถิติพรรณนา สถิติเบื้องต้นที่แบ่งกลุ่ม ตารางการแจกแจง ทดสอบ T-Test ความแปรปรวนทางเดียว ความแปรปรวนหลายทาง สหสัมพันธ์ วิเคราะห์ถดถอย เขียนกราฟ ทดสอบ Nonparam	ตัวแปร วิธีการคัดเลือกตัวแปร	Stepwise Forward Backward Enter Remove Test

จอภาพ 2.9.2 วิธีการคัดเลือกตัวแปรในการวิเคราะห์ถดถอย

จากจอภาพ 2.9.2 เมื่อเลือกวิธีการคัดเลือกตัวแปรได้โดยใช้ลูกศรเลื่อน เมื่อเลือกรายการที่ต้องการ กด Enter และเลือกรายการต่อไปโดย กด Enter เช่นกัน

วิธีการคัดเลือกตัวแปร คือ

- Stepwise : คัดเลือกตัวแปรอิสระออกจากสมการทีละตัวแปร
- Forward : คัดเลือกตัวแปรอิสระเข้าไปในสมการทีละตัวแปร
- Backward : คัดเลือกตัวแปรอิสระออกจาก สมการถดถอยทีละตัวแปร
- Enter : ตัวแปรอิสระเข้าไปในสมการ ได้โดยไม่ต้องใช้เกณฑ์ใด ๆ
- Remove : ตัวแปรอิสระออกจากสมการเมื่อตัวแปรนั้นผ่านเกณฑ์เข้าไปในสมการแล้ว
- Test : ทดสอบความมีนัยสำคัญ ของ ตัวแปรอิสระโดยใช้ค่าสัมประสิทธิ์การตัดสินใจ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
สถิติพรรณนา
สถิติเบื้องต้นที่แบ่งกลุ่ม
ตารางการแจกแจง
ทดสอบ T-Test
ความแปรปรวนทางเดียว
ความแปรปรวนหลายทาง
สหสัมพันธ์
วิเคราะห์ถดถอย
เขียนกราฟ
ทดสอบ Nonparametric

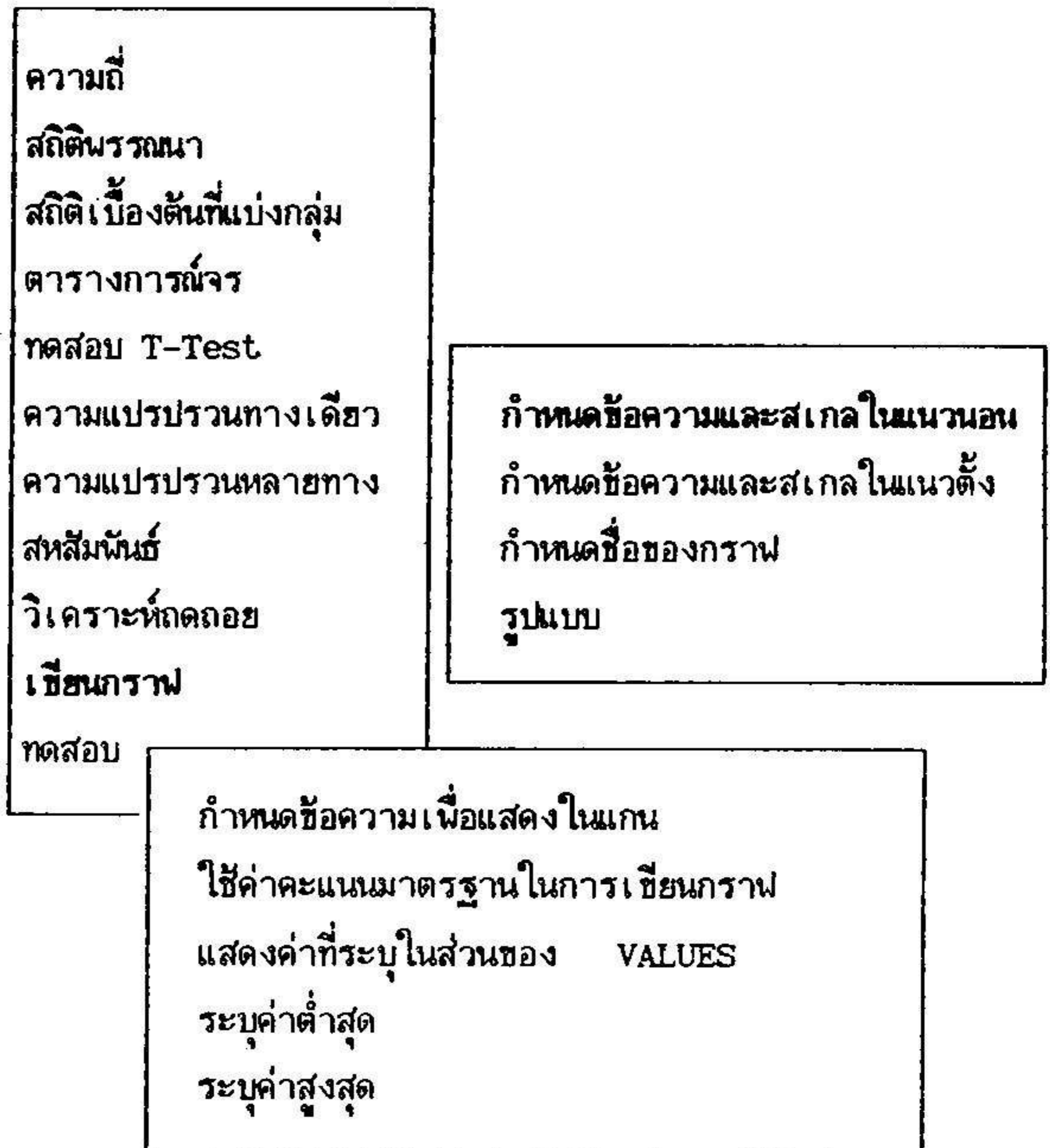
กำหนดข้อความและสเกลในแนวนอน
กำหนดข้อความและสเกลในแนวตั้ง
กำหนดชื่อของกราฟ
รูปแบบ

จอภาพ 2.10 การเขียนกราฟ

จากจอภาพ 2.10 เมื่อเลือกรายการเขียนกราฟแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- กำหนดข้อความและสเกลในแนวนอน : จะปรากฏเป็นจอภาพ 2.10.1
- กำหนดข้อความและสเกลในแนวตั้ง : จะปรากฏเป็นจอภาพ 2.10.2
- กำหนดชื่อของกราฟ : จะปรากฏเป็นจอภาพ 2.10.3
- รูปแบบ : จะปรากฏเป็นจอภาพ 2.10.4

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



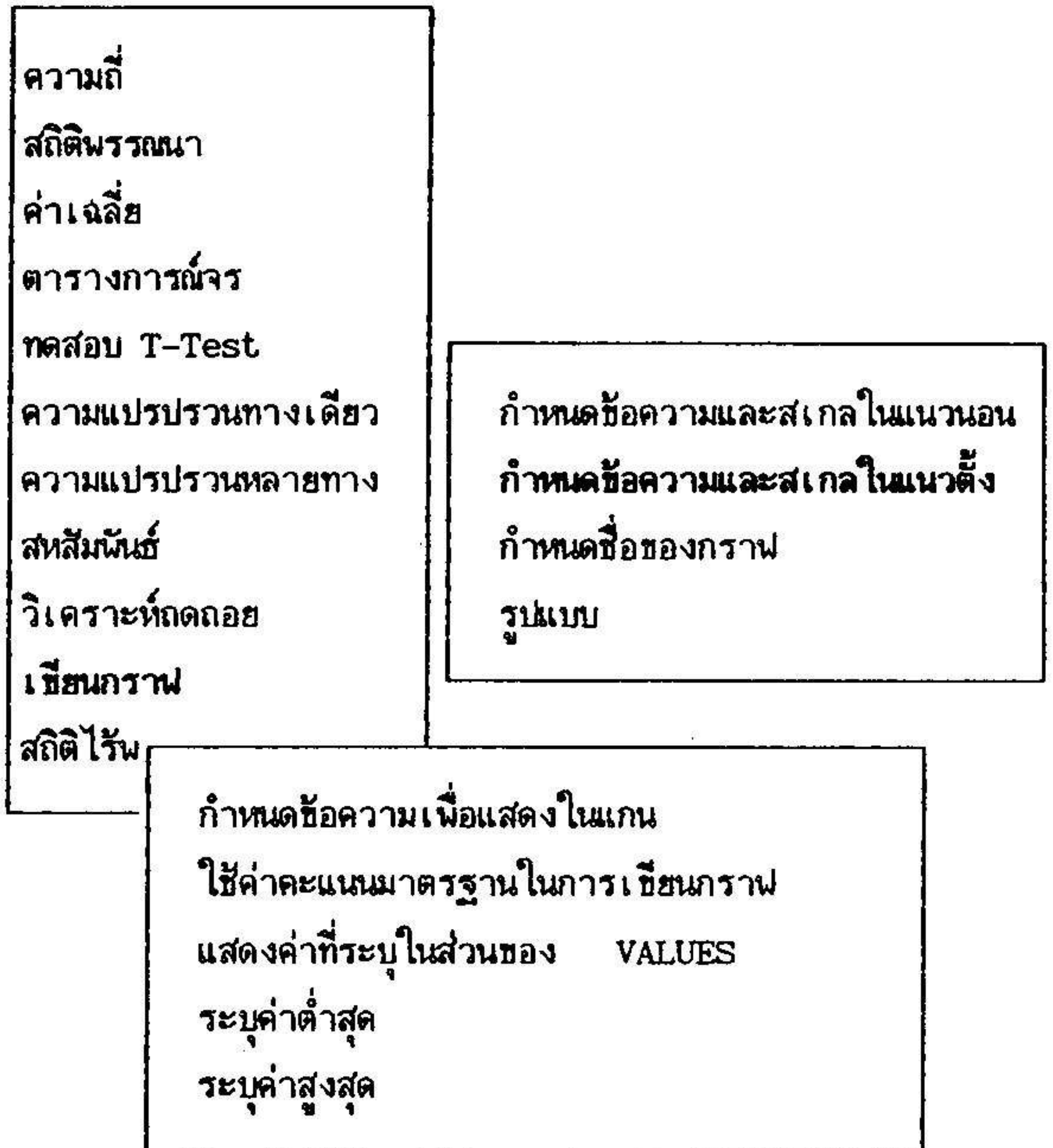
จอภาพ 2.10.1 การกำหนดข้อความและสเกลในแนวนอน

จากจอภาพ 2.10.1 เมื่อเลือกรายการกำหนดข้อความและสเกลในแนวนอนแล้ว ได้ใช้ลูกศรเลื่อนและเมื่อเลือกได้รายการที่ต้องการกด Enter และเลือกรายการต่อไปได้โดย กด Enter เช่นกัน

รายการที่ให้เลือก คือ

กำหนดข้อความเพื่อแสดงในแกน
 ใช้ค่าคะแนนมาตรฐานในการเขียนกราฟ
 แสดงค่าที่ระบุในส่วนของ VALUES
 ระบุค่าต่ำสุด
 ระบุค่าสูงสุด

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



จอภาพ 2.10.2 การกำหนดข้อความและสเกลในแนวตั้ง

จากจอภาพ 2.10.2 เมื่อเลือกรายการกำหนดข้อความและสเกลในแนวตั้งแล้วใช้ลูกศรเลื่อนและเมื่อเลือกรายการที่ต้องการ กด Enter และเลือกรายการต่อไปโดย กด Enter เช่นกัน
 รายการที่ให้เลือก คือ

- กำหนดข้อความเพื่อแสดงในแกน
- ใช้ค่าคะแนนมาตรฐานในการเขียนกราฟ
- แสดงค่าที่ระบุในส่วนของ VALUES
- ระบุค่าต่ำสุด และ ระบุค่าสูงสุด

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

กำหนดชื่อของกราฟ

จอภาพ 2.10.3 การกำหนดชื่อกราฟ

จากจอภาพ 2.10.3 เมื่อเลือกรายการกำหนดชื่อกราฟแล้วใส่ข้อมูลได้ดังนี้

กำหนดชื่อของกราฟ : ใส่ชื่อของกราฟที่ต้องกำหนด

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่			
สถิติพรรณนา			
สถิติเบื้องต้นที่แบ่งกลุ่ม			
ตารางการถัว			
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			
		ตัวแปรที่ต้องการทดสอบ	
		Binomial	
		Mcnemar	
		Wilcoxon	
		Cochran Q test	
		Frideman	
		Kendall	
		Median	
		Mann-Whitney U	
		Kolmogorov-Smirno	
		Wald-Wolfowitz	
		Kruskal-Wallis	
		รูปแบบ	
		คำสั่งสถิติ	

จอภาพ 2.11 รายการทดสอบ Nonparametric

จากจอภาพ 2.11 เมื่อเลือกรายการทดสอบ Nonparametric แล้วเลือกรายการได้ดังนี้

- ตัวแปรที่ต้องการทดสอบ : จะปรากฏเป็นจอภาพ 2.11.1
- Binomial : จะปรากฏเป็นจอภาพ 2.11.2
- Mcnemar : จะปรากฏเป็นจอภาพ 2.11.3
- Wilcoxon : จะปรากฏเป็นจอภาพ 2.11.4
- Cochran Q test : จะปรากฏเป็นจอภาพ 2.11.5
- Frideman : จะปรากฏเป็นจอภาพ 2.11.6

- Kendall : จะปรากฏเป็นจอภาพ 2.11.7
- Median : จะปรากฏเป็นจอภาพ 2.11.8
- Mann-Whitney U : จะปรากฏเป็นจอภาพ 2.11.9
- Kolmogorov-Smirno : จะปรากฏเป็นจอภาพ 2.11.10
- Wald-Wolfowitz : จะปรากฏเป็นจอภาพ 2.11.11
- Kruskal-Wallis : จะปรากฏเป็นจอภาพ 2.11.12
- รูปแบบ : จะปรากฏเป็นจอภาพ 2.11.13
- ค่าสถิติ : จะปรากฏเป็นจอภาพ 2.11.14

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	วัตถุประสงค์
--------	----------------	------------	--------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

จอภาพ 2.11.1 ชื่อตัวแปรของการทดสอบ Nonparametric

จากจอภาพ 2.11.1 เมื่อเลือกรายการทดสอบ Nonparametric แล้วใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

จอภาพ 2.11.2 รายการทดสอบ Binomial

จากจอภาพ 2.11.2 เมื่อเลือกรายการทดสอบ Binomial แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	ชื่อตัวแปร	<input type="text"/>
สถิติพรรณนา		กับตัวแปร	<input type="text"/>
สถิติเบื้องต้นที่แบ่งกลุ่ม		VALUE1	<input type="text"/>
ตารางการถัว		VALUE2	<input type="text"/>
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			

จอภาพ 2.11.3 รายการทดสอบ McNemar

จากจอภาพ 2.11.3 เมื่อเลือกรายการทดสอบ McNemar แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร, กับตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

VALUE1, VALUE2 : ใส่ค่าของตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประกอบ
--------	----------------	------------	------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

กับตัวแปร

จอภาพ 2.11.4 รายการทดสอบ Wilcoxon

จากจอภาพ 2.11.4 เมื่อเลือกรายการทดสอบ Wilcoxon แล้วให้ใส่ข้อมูลได้ดังนี้
 ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

จอภาพ 2.11.5 รายการทดสอบ Cochran Q test.

จากจอภาพ 2.11.5 เมื่อเลือกรายการทดสอบ Cochran Q test แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

จอภาพ 2.11.6 รายการทดสอบ Frideman

จากจอภาพ 2.11.6 เมื่อเลือกรายการทดสอบ Frideman แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่
 สถิติพรรณนา
 สถิติเบื้องต้นที่แบ่งกลุ่ม
 ตารางการแจกแจง
 ทดสอบ T-Test
 ความแปรปรวนทางเดียว
 ความแปรปรวนหลายทาง
 สหสัมพันธ์
 วิเคราะห์ถดถอย
 เขียนกราฟ
 ทดสอบ Nonparametric

ชื่อตัวแปร

จอภาพ 2.11.7 รายการทดสอบ Kendall

จากจอภาพ 2.11.7 เมื่อเลือกรายการทดสอบ Kendall แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

ความถี่			
สถิติพรรณนา			
สถิติเบื้องต้นที่แบ่งกลุ่ม			
ตารางการแจกแจง			
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			
	ชื่อตัวแปร	<input type="text"/>	
	กับตัวแปร	<input type="text"/>	
	VALUE1	<input type="text"/>	
	VALUE2	<input type="text"/>	

จอภาพ 2.11.8 รายการทดสอบ Median

จากจอภาพ 2.11.8 เมื่อเลือกรายการทดสอบ Median แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

VALUE : ใส่ค่าของตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราเปอร์เซ็นต์
--------	----------------	------------	------------------

ความถี่			
สถิติพรรณนา			
สถิติเบื้องต้นที่แบ่งกลุ่ม			
ตารางการแจกแจง			
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			
	ชื่อตัวแปร	<input type="text"/>	
	กับตัวแปร	<input type="text"/>	
	VALUE1	<input type="text"/>	
	VALUE2	<input type="text"/>	

จอภาพ 2.11.9 รายการทดสอบ Mann-Whitney U

จากจอภาพ 2.11.9 เมื่อเลือกรายการทดสอบ Mann-Whitney U แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

VALUE : ใส่ค่าของตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประ โยชน์
--------	----------------	------------	----------------

ความถี่			
สถิติพรรณนา			
สถิติเบื้องต้นที่แบ่งกลุ่ม			
ตารางการแจกแจง			
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			
	ชื่อตัวแปร	<input type="text"/>	
	กับตัวแปร	<input type="text"/>	
	VALUE1	<input type="text"/>	
	VALUE2	<input type="text"/>	

จอภาพ 2.11.10 รายการทดสอบ Kolmogorov-Smirnov

จากจอภาพ 2.11.10 เมื่อเลือกรายการทดสอบ Kolmogorov-Smirnov แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ

VALUE : ใส่ค่าของตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

ความถี่			
สถิติพรรณนา			
สถิติเบื้องต้นที่แบ่งกลุ่ม			
ตารางการแจกแจง			
ทดสอบ T-Test			
ความแปรปรวนทางเดียว			
ความแปรปรวนหลายทาง			
สหสัมพันธ์			
วิเคราะห์ถดถอย			
เขียนกราฟ			
ทดสอบ Nonparametric			

ชื่อตัวแปร	<input type="text"/>
กับตัวแปร	<input type="text"/>
VALUE1	<input type="text"/>
VALUE2	<input type="text"/>

จอภาพ 2.11.12 รายการทดสอบ Kruskal-Wallis

จากจอภาพ 2.11.12 เมื่อเลือกรายการทดสอบ Kruskal-Wallis แล้วให้ใส่ข้อมูลได้ดังนี้

ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการทดสอบ
 VALUE : ใส่ค่าของตัวแปร

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

แสดงตัวแปรและค่า
 เก็บข้อมูลและผลลัพธ์
 เก็บข้อมูลแบบ ASCII (Export)
 อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
 เรียงลำดับข้อมูล
 เครื่องพิมพ์

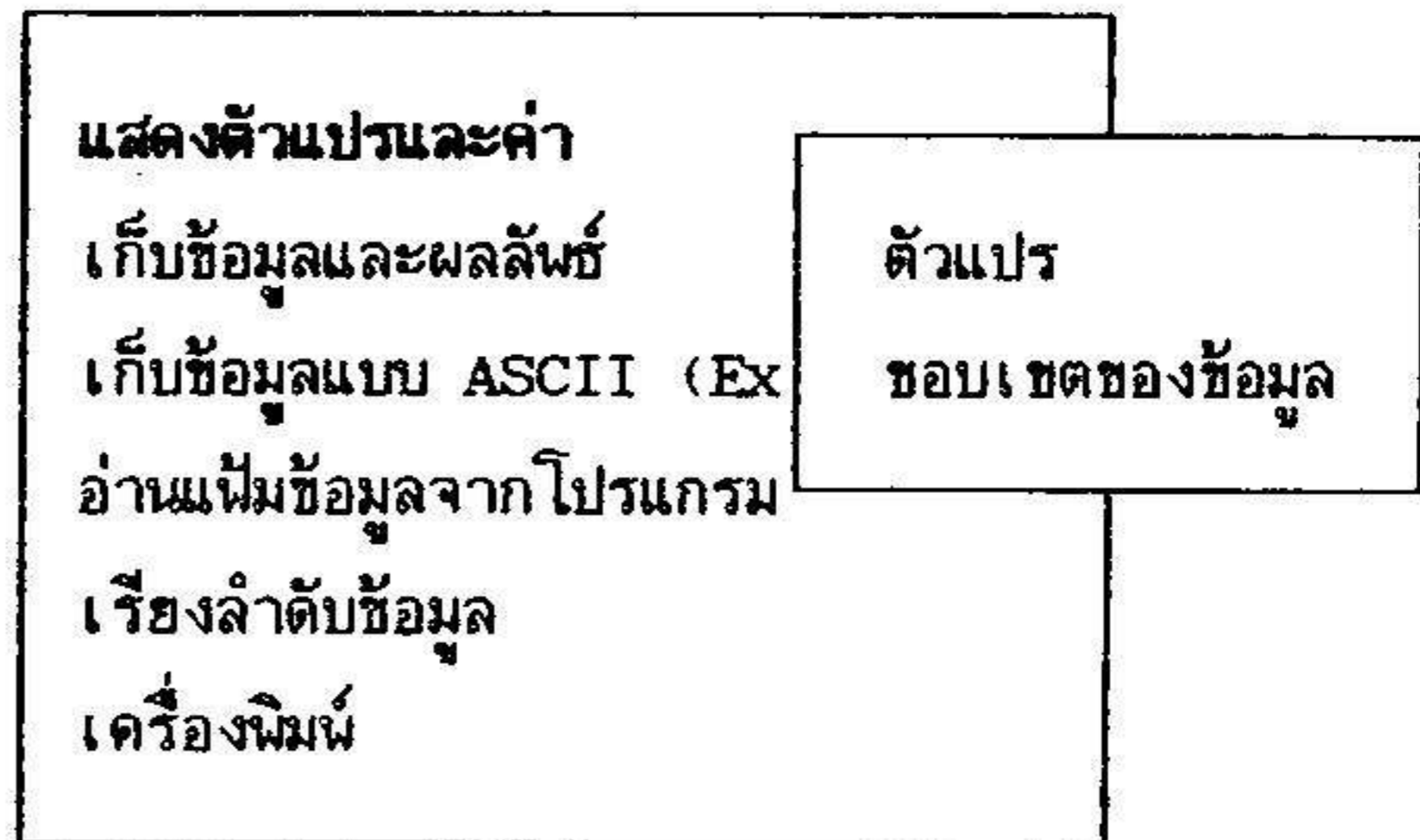
F1 Help F10 Exit

จอภาพ 3 รายการคำสั่งสถิติ

จากจอภาพ 3 เมื่อเลือกรายการคำสั่งสถิติแล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- แสดงตัวแปรและค่า : จะปรากฏเป็นจอภาพ 3.1
- เก็บข้อมูลและผลลัพธ์ : จะปรากฏเป็นจอภาพ 3.2
- เก็บข้อมูลแบบ ASCII : จะปรากฏเป็นจอภาพ 3.3
- อ่านเพิ่มข้อมูลจากโปรแกรม : จะปรากฏเป็นจอภาพ 3.4
- เรียงลำดับข้อมูล : จะปรากฏเป็นจอภาพ 3.5
- เครื่องพิมพ์ : เป็นการ set printer on ในส่วนของ
 การนำเพิ่มคำสั่งไปเรียกใช้ใน SPSS PC
 หรือ SPSS PC⁺ และ จะทำให้ทุกคำสั่งที่
 อยู่ในแฟ้มคำสั่งนี้พิมพ์ผลลัพธ์ทางเครื่องพิมพ์

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



F1 Help Esc Menu

จอภาพ 3.1 แสดงตัวแปรและค่า

จากจอภาพ 3.1 เมื่อเลือกรายการแสดงตัวแปรและค่า แล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- ตัวแปร : จะปรากฏเป็นจอภาพ 3.1.1
- ขอบเขตของข้อมูล : จะปรากฏเป็นจอภาพ 3.1.2

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

แสดงตัวแปรและค่า
 เก็บข้อมูลและผลลัพธ์
 เก็บข้อมูลแบบ ASCII (Export)
 อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
 เรียงลำดับข้อมูล
 เครื่องพิมพ์

ตัวแปร

รายชื่อตัวแปร

Esc Menu

จอภาพ 3.1.1 แสดงตัวแปร

- จากจอภาพ 3.1.1 เมื่อเลือกรายการแสดงตัวแปรแล้วใส่ข้อมูลได้ดังนี้
- รายชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการให้แสดงค่าออกมา
 ในกรณีที่ กด Enter ผ่าน จะเป็นการแสดงรายชื่อ
 ตัวแปรทั้งหมด ในขณะนั้นออกมา

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

แสดงตัวแปรและค่า
 เก็บข้อมูลและผลลัพธ์
 เก็บข้อมูลแบบ ASCII (Export)
 อ่านแฟ้มข้อมูลจากโปรแกรมอื่น (Import)
 เรียงลำดับข้อมูล
 เครื่องพิมพ์

ขอบเขตข้อมูล

ลำดับ เริ่มต้นของชุดข้อมูล	<input type="text"/>
ลำดับสุดท้ายของชุดข้อมูล	<input type="text"/>
เพิ่มชุดข้อมูลครั้งละ	<input type="text"/>

Esc Menu

จอภาพ 3.1.2 ขอบเขตข้อมูล

จากจอภาพ 3.1.2 เมื่อเลือกรายการขอบเขตข้อมูลแล้ว ใส่ข้อมูล ได้ดังนี้

- | | |
|----------------------------|---|
| ลำดับ เริ่มต้นของชุดข้อมูล | : ใส่ตัวเลขแทนชุดข้อมูลที่ต้องการเริ่มต้น |
| ลำดับสุดท้ายของชุดข้อมูล | : ใส่ตัวเลขแทนชุดข้อมูลที่ต้องการเป็นชุดสุดท้าย |
| เพิ่มชุดข้อมูลครั้งละ | : ใส่ตัวเลขแทนการเพิ่มของชุดข้อมูล |

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

เก็บข้อมูลและผลลัพธ์ลงแฟ้มข้อมูล

ชื่อแฟ้มข้อมูลใหม่

Esc Menu

จอภาพ 3.2 การเก็บข้อมูลและผลลัพธ์ลงแฟ้มข้อมูล

จากจอภาพ 3.2 เมื่อเลือกรายการเก็บข้อมูลและผลลัพธ์ลงแฟ้มข้อมูลแล้วใส่ข้อมูลได้ดังนี้

ชื่อแฟ้มข้อมูลใหม่ : ใส่ชื่อแฟ้มข้อมูลใหม่ที่ต้องการเก็บ โดยระบุ Drive ที่แฟ้มข้อมูล นั้นอยู่ เช่น c:\spss\data.doc

เมื่อใส่ชื่อแฟ้มข้อมูลเรียบร้อยแล้วจะปรากฏจอภาพ 3.2.1

เก็บข้อมูลและผลลัพธ์ลงเพิ่มข้อมูล

ชื่อเพิ่มข้อมูลใหม่

เก็บชื่อตัวแปรทั้งหมด [Y/N]

จอภาพ 3.2.1 ชื่อเพิ่มข้อมูลใหม่

ตอบ Y : แสดงว่าเก็บชื่อตัวแปรทั้งหมดให้

ตอบ N : จะปรากฏข้อความดังจอภาพ 3.2.2

เก็บข้อมูลและผลลัพธ์ลงเพิ่มข้อมูล

ชื่อเพิ่มข้อมูลใหม่

เก็บชื่อตัวแปรทั้งหมด [Y/N]

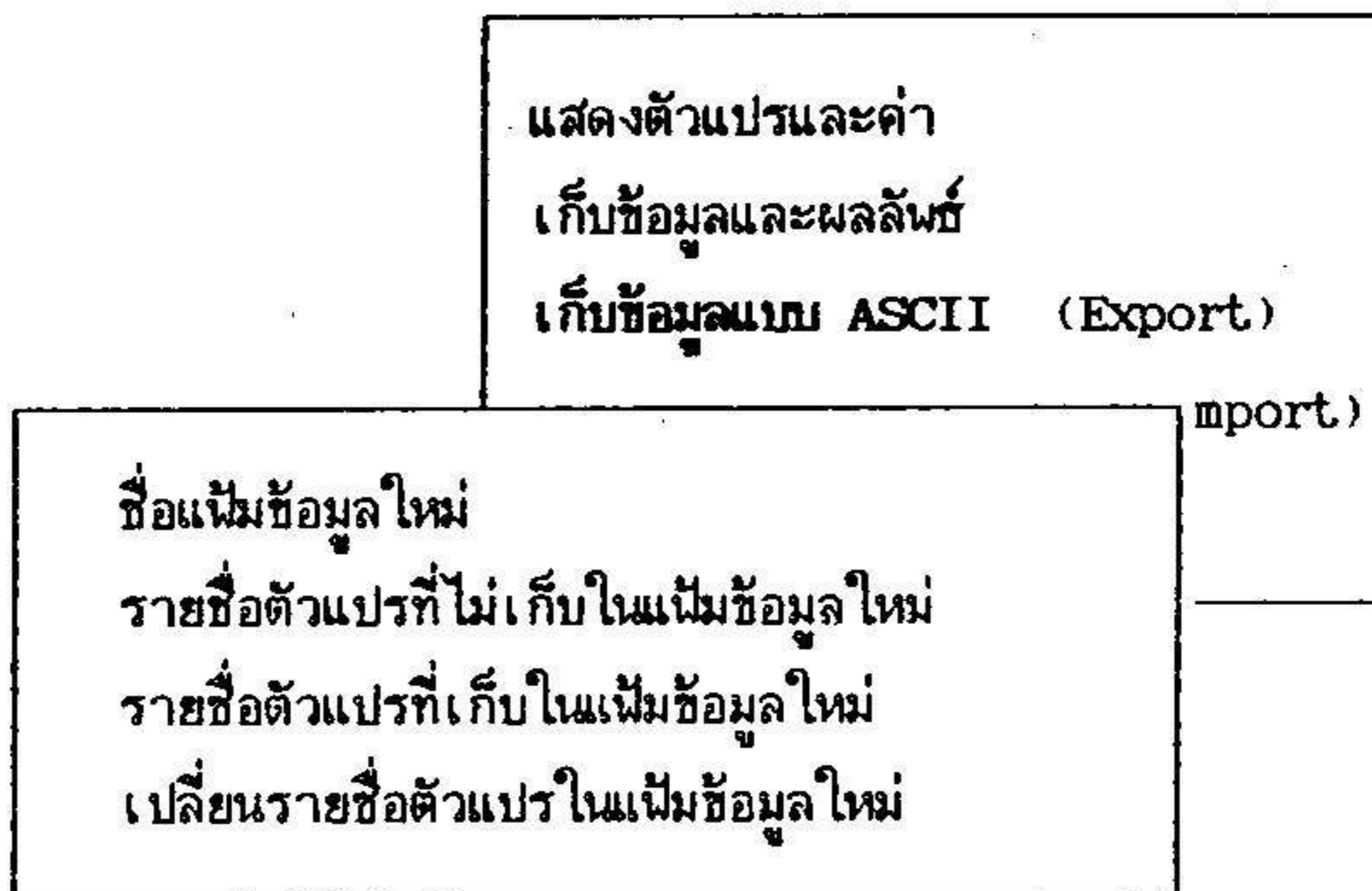
ชื่อตัวแปรที่ไม่ต้องการเก็บ

จอภาพ 3.2.2 เก็บชื่อตัวแปร

จากจอภาพ 3.2.2 ใส่ข้อมูลในรายการ

ชื่อตัวแปรที่ไม่ต้องการเก็บ : ใส่ชื่อตัวแปรที่ไม่ต้องการเก็บ

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------



F1 Help	Esc	Menu
---------	-----	------

จอภาพ 3.3 เก็บข้อมูลแบบ ASCII

จากจอภาพ 3.3 เมื่อเลือกรายการเก็บข้อมูลแบบ ASCII แล้วใช้ลูกศรเลื่อนเพื่อเลือกรายการได้ดังนี้

- | | | |
|--|---|------------------------|
| ชื่อเพิ่มข้อมูลใหม่ | : | จะปรากฏเป็นจอภาพ 3.3.1 |
| รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่ | : | จะปรากฏเป็นจอภาพ 3.3.2 |
| รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่ | : | จะปรากฏเป็นจอภาพ 3.3.3 |
| เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่ | : | จะปรากฏเป็นจอภาพ 3.3.4 |

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

แสดงตัวแปรและค่า
 เก็บข้อมูลและผลลัพธ์
 เก็บข้อมูลแบบ ASCII (Export)
 อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
 เรียงลำดับข้อมูล
 เครื่องพิมพ์

ชื่อเพิ่มข้อมูลใหม่	<input type="text"/>
---------------------	----------------------

Esc Menu

จอภาพ 3.3.1 ชื่อเพิ่มข้อมูลใหม่ของการเก็บข้อมูลแบบ ASCII

จากจอภาพ 3.3.1 เมื่อเลือกรายการเก็บข้อมูลแบบ ASCII แล้วใส่ข้อมูลได้ดังนี้
 ชื่อเพิ่มข้อมูลใหม่ : ใส่ชื่อเพิ่มข้อมูลใหม่ที่ต้องการนำ Active file
 ที่มีอยู่ไปบันทึกไว้

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

แสดงตัวแปรและค่า
 เก็บข้อมูลและผลลัพธ์
 เก็บข้อมูลแบบ ASCII (Export)
 อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
 เรียงลำดับข้อมูล
 เครื่องพิมพ์

รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่

Esc Menu

จอภาพ 3.3.2 รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่

จากจอภาพ 3.3.2 เมื่อเลือกรายการแล้วใส่ข้อมูลได้ดังนี้

รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่ : ใส่ชื่อตัวแปรที่ไม่ต้องการเก็บในเพิ่มข้อมูลใหม่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

แสดงตัวแปรและค่า

เก็บข้อมูลและผลลัพธ์

เก็บข้อมูลแบบ ASCII (Export)

อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)

เรียงลำดับข้อมูล

เครื่องหมาย

รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่

Esc Menu

จอภาพ 3.3.3 รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่

จากจอภาพ 3.3.3 เมื่อเลือกรายการแล้วใส่ข้อมูลได้ดังนี้

รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่ : ใส่ชื่อตัวแปรที่ต้องการเก็บในเพิ่มข้อมูลใหม่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

แสดงตัวแปรและค่า
เก็บข้อมูลและผลลัพธ์
เก็บข้อมูลแบบ ASCII (Export)
อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
เรียงลำดับข้อมูล
เครื่องพิมพ์

เปลี่ยนรายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่

รายชื่อตัวแปรใหม่เพิ่มข้อมูลเก่า	<input type="text"/>
รายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่	<input type="text"/>

Esc Menu

จอภาพ 3.3.4 เปลี่ยนรายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่

จากจอภาพ 3.3.4 เมื่อเลือกรายการเปลี่ยนรายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่ แล้วใส่ข้อมูลได้ดังนี้

- | | | |
|----------------------------------|---|---|
| รายชื่อตัวแปรใหม่เพิ่มข้อมูลเก่า | : | ใส่ชื่อตัวแปรที่มีอยู่ในเพิ่มข้อมูลเก่า |
| รายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่ | : | ใส่ชื่อตัวแปรที่จะเก็บในเพิ่มข้อมูลใหม่ |

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

แสดงตัวแปรและค่า
เก็บข้อมูลและผลลัพธ์
เก็บข้อมูลแบบ ASCII (Export)

Import)

ชื่อเพิ่มข้อมูลใหม่
รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่
รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่
เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่

F1 Help Esc Menu

จอภาพ 3.4 รายการเกี่ยวกับเพิ่มข้อมูล

- จากจอภาพ 3.4 เมื่อเลือกรายการแล้วใช้ลูกศรเลื่อนได้เพื่อเลือกรายการดังนี้
- | | |
|--|--------------------------|
| ชื่อเพิ่มข้อมูลใหม่ | : จะปรากฏเป็นจอภาพ 3.4.1 |
| รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่ | : จะปรากฏเป็นจอภาพ 3.4.2 |
| รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่ | : จะปรากฏเป็นจอภาพ 3.4.3 |
| เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่ | : จะปรากฏเป็นจอภาพ 3.4.4 |

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

แสดงตัวแปรและค่า
 เก็บข้อมูลและผลลัพธ์
 เก็บข้อมูลแบบ ASCII (Export)
 อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
 เรียงลำดับข้อมูล
 เครื่องพิมพ์

ชื่อเพิ่มข้อมูลใหม่

Esc Menu

จอภาพ 3.4.1 อ่านเพิ่มข้อมูลจากโปรแกรมอื่น

จากจอภาพ 3.4.1 เมื่อเลือกรายการอ่านเพิ่มข้อมูลจากโปรแกรมอื่น แล้วใส่ข้อมูลได้
 ดังนี้

ชื่อเพิ่มข้อมูลใหม่ : ใส่ชื่อเพิ่มข้อมูลใหม่ที่ต้องการนำ Active file
 เพื่อเรียกมาใช้

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

แสดงตัวแปรและค่า
เก็บข้อมูลและผลลัพธ์
เก็บข้อมูลแบบ ASCII (Export)
อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
เรียงลำดับข้อมูล
เครื่องพิมพ์

รายชื่อตัวแปรที่ไม่เรียกในเพิ่มข้อมูลใหม่

Esc Menu

จอภาพ 3.4.2 รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่

จากจอภาพ 3.4.2 เมื่อเลือกรายการแล้วใส่ข้อมูลได้ดังนี้

รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่ : ใส่ชื่อตัวแปรที่ไม่ต้องการเรียกใช้ในเพิ่มข้อมูลใหม่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตราประโยชน์
--------	----------------	------------	---------------

แสดงตัวแปรและค่า
เก็บข้อมูลและผลลัพธ์
เก็บข้อมูลแบบ ASCII (Export)
อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
เรียงลำดับข้อมูล
เครื่องพิมพ์

รายชื่อตัวแปรที่เรียกในเพิ่มข้อมูลใหม่

Esc Menu

จอภาพ 3.4.3 รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่

จากจอภาพ 3.4.3 เมื่อเลือกรายการแล้วใส่ข้อมูลได้ดังนี้

รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่ : ใส่ชื่อตัวแปรที่ต้องการเรียกใช้ในเพิ่มข้อมูลใหม่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประโยชน์
--------	----------------	------------	--------------

แสดงตัวแปรและค่า
เก็บข้อมูลและผลลัพธ์
เก็บข้อมูลแบบ ASCII (Export)
อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import)
เรียงลำดับข้อมูล
เครื่องพิมพ์

เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่

รายชื่อตัวแปรในเพิ่มข้อมูลเก่า

รายชื่อตัวแปรในเพิ่มข้อมูลใหม่

Esc Menu

จอภาพ 3.4.4 เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่

จากจอภาพ 3.4.4 เมื่อเลือกรายการเปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่ แล้วใส่ข้อมูลได้ดังนี้

รายชื่อตัวแปรในเพิ่มข้อมูลเก่า

: ใส่ชื่อตัวแปรที่มีอยู่ในเพิ่มข้อมูลเก่า

รายชื่อตัวแปรในเพิ่มข้อมูลใหม่

: ใส่ชื่อตัวแปรที่จะเรียกใช้ในเพิ่มข้อมูลใหม่

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อัตรประกอบ
--------	----------------	------------	------------

การเรียงลำดับข้อมูล

ชื่อตัวแปร

วิธีเรียงลำดับข้อมูล

* A เรียงจากน้อยไปมาก
D เรียงจากมากไปน้อย

Esc Menu

จอภาพ 3.5 การเรียงลำดับข้อมูล

- จากจอภาพ 3.5 เมื่อเลือกรายการการเรียงลำดับข้อมูล แล้วใส่ข้อมูลได้ดังนี้
- ชื่อตัวแปร : ใส่ชื่อตัวแปรที่ต้องการเรียงลำดับ
- วิธีเรียงลำดับข้อมูล : ถ้าใส่ A ข้อมูลจะเรียงลำดับจากน้อยไปมาก
ถ้าใส่ D ข้อมูลจะเรียงลำดับจากมากไปน้อย

ข้อมูล	คำสั่งทางสถิติ	คำสั่งช่วย	อรรถประโยชน์
--------	----------------	------------	--------------

Dir
Exit

F1 Help F10 Exit

จอภาพ 4 รายการอรรถประโยชน์

- จากจอภาพ 4 เมื่อเลือกรายการอรรถประโยชน์แล้วเลือกรายการได้ดังนี้
- Dir : จะแสดงรายชื่อไดเรกทอรีของดิสก์ไดรฟ์ที่ใช้งานขณะนั้น
 - Exit : ออกจากโปรแกรมนี้

ภาคผนวก ข

โปรแกรม


```

' ตารางการสำรวจ      ',
' ทดสอบ T-Test        ',
' ความแปรปรวนทางเดียว   ',
' ความแปรปรวนหลายทาง   ',
' สหสัมพันธ์          ',
' วิเคราะห์ถดถอย       ',
' เขียนกราฟ           ',
' ทดสอบ Nonparametric ');

```

```
LastChoice:11),
```

```
(Win : (27,04,67,11));
```

```
Col : (32,02,02,02,02,02,02,00,00,00,00,00);
```

```
Row : (02,01,02,03,04,05,06,00,00,00,00,00);
```

```
Msg : (' คำสั่งช่วย ',
' แสดงตัวแปรและค่า      ',
' เก็บข้อมูลและผลลัพธ์    ',
' เก็บข้อมูลแบบ ASCII (Export) ',
' อ่านเพิ่มข้อมูลจากโปรแกรมอื่น (Import) ',
' เรียงลำดับข้อมูล      ',
' เครื่องพิมพ์            ',
'', '', '', '', '');
```

```
LastChoice:6),
```

```
(Win: (51,04,64,07));
```

```
Col : (51,02,02,00,00,00,00,00,00,00,00,00);
```

```
Row : (02,01,02,00,00,00,00,00,00,00,00,00);
```

```
Msg : (' อັถถประโยชน์ ',
' Dir      ',
' Exit     ',
'', '', '', '', '', '', '', '', '');
```

```
LastChoice:2));
```

```
Const ChoiceMenu : array[1..MaxMenu] of 1..MaxChoice = (1,1,1,1);
```



```

    StatusMenu : 1..MaxMenu = 1;
    StatusChoi : 1..Maxchoi = 1;
    Finish     : Boolean     = False; { Run Flag }
var  Key      : char;
     Sub      : StrucSub;
     CurChoice : byte;
     f,fp     : text;
     st       : string;

```

```

procedure DisplayMainMenu;

```

```

var I : byte;

```

```

begin

```

```

    window(1,1,80,25);

```

```

    SetAttr(lowdisplay);

```

```

    for I := 1 to MaxMenu do

```

```

        with Menu[I] do

```

```

            begin

```

```

                gotoxy(Col[0],Row[0]);

```

```

                write(Msg[0]);

```

```

            end;

```

```

end;

```

```

procedure ChoiceActive(Old,New : byte);

```

```

begin

```

```

    with Menu[StatusMenu] do

```

```

        begin

```

```

            Window(Win[1]+1,Win[2]+1,Win[3],Win[4]);

```

```

            { Display old choice with Normal Attribute }

```

```

            Setattr(LowDisplay);

```

```

            Gotoxy(Col[old],Row[old]);

```

```

            write(Msg[old]);

```

```

    { Display new Choice with Reverse Attribute }
    SetAttr(ReverseLow);
    gotoxy(Col[New],Row[New]);
    write(Msg[New]);
    window(1,1,80,25);gotoxy(78,24);
end;
end;

procedure MenuActive(New : byte);
var i : byte;
begin
    { Display Menu Header }
    DisplayMainMenu;
    setattr(reverselow);
    with Menu[New] do
    begin
        gotoxy(Col[0],Row[0]);
        write(Msg[0]);
        { create Window }
        setwinattr(lowdisplay);
        setboxattr(lowdisplay);
        setcharattr(lowdisplay);
        SetBoxStyle(single);
        SetWinHeader('');
        WindowOpen(Win[1],Win[2],Win[3],Win[4]);
        { Display Choices of Menu }
        for i := 1 to LastChoice do
        begin
            gotoxy(Col[i],Row[i]);
            write(Msg[i]);
        end;
    end;
end;

```

```

    end;
    ChoiceActive(1,ChoiceMenu[New]);
end;

procedure MoveUp;
var CurrentChoice : byte;
begin
    CurrentChoice := ChoiceMenu[statusMenu];
    if CurrentChoice = 1 then
        ChoiceMenu[StatusMenu] := Menu[StatusMenu].LastChoice
    else
        ChoiceMenu[StatusMenu] := CurrentChoice - 1;
        ChoiceActive(CurrentChoice,ChoiceMenu[StatusMenu]);
end;

procedure MoveDown;
var CurrentChoice : byte;
begin
    CurrentChoice := ChoiceMenu[StatusMenu];
    if CurrentChoice = Menu[StatusMenu].LastChoice then
        ChoiceMenu[StatusMenu] := 1
    else
        ChoiceMenu[StatusMenu] := CurrentChoice + 1;
        ChoiceActive(CurrentChoice,ChoiceMenu[StatusMenu]);
end;

procedure MoveForward;
begin
    WindowClose;
    if StatusMenu + 1 > MaxMenu then
        StatusMenu := 1

```

```
    else
        StatusMenu := StatusMenu + 1;
        MenuActive(StatusMenu);
end;
```

```
procedure MoveBack;
```

```
begin
```

```
    WindowClose;
```

```
    if StatusMenu = 1 then
```

```
        StatusMenu := MaxMenu
```

```
    else
```

```
        StatusMenu := StatusMenu-1;
```

```
        MenuActive(StatusMenu);
```

```
end;
```

```
procedure MovetoFirstChoice;
```

```
begin
```

```
    ChoiceActive(ChoiceMenu[StatusMenu], 1);
```

```
    ChoiceMenu[StatusMenu] := 1;
```

```
end;
```

```
procedure MoveToLastChoice;
```

```
begin
```

```
    with Menu[StatusMenu] do
```

```
        begin
```

```
            ChoiceActive(ChoiceMenu[StatusMenu], LastChoice);
```

```
            ChoiceMenu[StatusMenu] := LastChoice;
```

```
        end;
```

```
end;
```

```
procedure MoveToFirstMenu;
```

```
begin
```

```
    WindowClose;
```

```
    StatusMenu := 1;
```

```
    MenuActive(1);
```

```
end;
```

```
procedure MovetoLastMenu;
```

```
begin
```

```
    WindowClose;
```

```
    StatusMenu := MaxMenu;
```

```
    MenuActive(MaxMenu);
```

```
end;
```

```
{-----}
```

```
procedure Quit;
```

```
begin
```

```
    close_win;
```

```
    clrscr;
```

```
    Finish := true;
```

```
end;
```

```
procedure Domenu1;
```

```
begin
```

```
    case ChoiceMenu[StatusMenu] of
```

```
        1 : begin Sub := Sdef;
```

```
                InitSub(Sub, StatusMenu, CurChoice);
```

```
            end;
```

```
        3 : begin Sub := Selec;
```

```
                InitSub(Sub, StatusMenu, CurChoice);
```

```
            end;
```

```

    2 : begin Sub := Transf;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
  end; { case }
end;

procedure Domenu2;
begin
  case ChoiceMenu[StatusMenu] of
    1 : begin Sub := SubFre;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    2 : begin Sub := SubDes;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    3 : begin Sub := SubMean;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    4 : begin Sub := SubCross;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    5 : begin Sub := SubT_Test;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    6 : begin Sub := SubOneway;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    7 : begin Sub := Sanova;
          InitSub(Sub, StatusMenu, CurChoice);
        end;
    8 : begin Sub := SubCorr;

```

```

        InitSub(Sub, StatusMenu, CurChoice);
    end;
9 : begin Sub := SubRegr;
        InitSub(Sub, StatusMenu, CurChoice);
    end;
10 : begin Sub := SubPlot;
        InitSub(Sub, StatusMenu, CurChoice);
    end;
11 : begin Sub := SubNPar;
        InitSub(Sub, StatusMenu, CurChoice);
    end;
end; { case }
end;

```

```

procedure Domenu3;
var s : string;
begin
    case ChoiceMenu[StatusMenu] of
        1 : begin Sub := Lis;
                InitSub(Sub, StatusMenu, CurChoice);
            end;
        2 : begin close_win;
                Psave;
                MenuActive(3);
            end;
        3,4 : begin Sub := Export;
                InitSub(Sub, StatusMenu, CurChoice);
            end;
        5 : begin close_win;
                Sort_case;
                MenuActive(3);
            end;
    end;
end;

```

```

        end;
    6 : begin assign(f,'c:\sptool\ex.sps');
        s := 'c:\sptool\ex0.sps';
        assign(fp,s);
        erase(fp);
        rename(f,s);
        assign(fp,'c:\sptool\ex.sps');
        st := 'set print on.';
        makefi(st,fp);
        reset(f);
        while not eof(f) do
            begin readln(f,st);
                    makefi(st,fp);
            end;
        end;
    end; { case }
end;

procedure Domenu4;
begin
    case ChoiceMenu[StatusMenu] of
        1 : begin windowclose;
                window(1,4,79,22);
                setattr(lowdisplay);
                DisplayFileName;
                MenuActive(4);
            end;
        2 : quit;
    end; {case}
end;

```



```
procedure DoMenu;
begin
    DisplayFuncKey;
    CurChoice := ChoiceMenu[StatusMenu];
    case statusMenu of
        1 : Domenu1;
        2 : Domenu2;
        3 : Domenu3;
        4 : Domenu4;
    end; { case }
end;
```

```
procedure DoF1;
var i:integer;
begin
    save_scr;
    case statusMenu of
        1 : Read_txt('c:\sptool\help\mdat.txt');
        2 : Read_txt('c:\sptool\help\proc.txt');
        3 : Read_txt('c:\sptool\help\hcom.txt');
        4 : Read_txt('c:\sptool\help\uses.txt');
    end; { case }
    readln; gotoxy(78,25);
    get_oldscr;
end;
```

```
procedure TestKey(Key: char);
begin
    if FuncKey then
        case (Key) of
            Home_Key : MoveToFirstMenu;
```

```

        End_Key    : MoveToLastMenu;
        Up_Key     : MoveUp;
        Lt_Key     : MoveBack;
        Rt_Key     : MoveForward;
        Dn_Key     : MoveDown;
        PgUp_Key  : MoveToFirstChoice;
        PgDn_Key  : MoveToLastChoice;
        F1_Key    : DoF1;
        F10_Key   : Quit;
    end { of Case }
else
    if Key = Return_Key Then
        DoMenu;
end;

```

```

begin { Main Menu }
    ClrScr;
    DirectVideo := True;
    CheckSnow   := True;
    Read_txt('c:\sptool\help\logo.txt');
    delay(2500);
    Read_txt('c:\sptool\help\intro.txt');
    gotoxy(76,25);readln;
    clrscr;
    InitWin;    { Display Menu }
    setwinattr(lowdisplay);
    setboxattr(lowdisplay);
    setcharattr(lowdisplay);
    setboxstyle(single);
    WindowOpen(1,1,79,3);

```

```
WindowOpen ( 1, 23, 79, 25 );  
DisplayFuncKey;  
DisplayMainMenu;      {OpenScreen;}  
MenuActive(1);  
repeat                { Loop read control key }  
    ReadFuncKey (Key) ;  
    Testkey (key) ;  
until Finish ;  
{ CloseScreen;}  
end.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\logo.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```


Πηληγον KEYBORD.PAS

```

Unit keyboard;
interface
uses Crt,Dos;
var Key      : char;
    FuncKey  : Boolean;
    st       : string;
    intnum   : integer;
    realnum  : real;
    code     : integer;

const Return_Key   = #13; Esc_Key   = #27;
      Shift_Tab    = #15;
      F1_Key       = #59; F10_Key  = #68;
      Home_Key     = #71; End_Key   = #79;
      Up_Key       = #72; Lt_Key    = #75;
      Rt_Key       = #77; Dn_Key    = #80;
      PgUp_Key     = #73; PgDn_Key  = #81;
      Ins_Key      = #82; Del_Key   = #83;

procedure ReadFuncKey(var Key  : char);
procedure ReadChar(var St:String;Count:integer);
procedure ReadChar1(var St :String; Count:integer);
procedure ReadReal(var value:real;count:integer;var code :integer);
procedure ReadInt(var value:integer;count:integer;var code :integer);
procedure checkword(var line : string; var word,cnt : byte);

implementation
procedure ReadFuncKey(var Key  : Char);
begin

```



```

Key := ReadKey;
if Key = #0 then
begin
    FuncKey := True;
    Key := ReadKey;
end
else
    FuncKey := False;
end;

{===== ReadChar =====}
procedure ReadChar(var St : String;Count:integer);
var Ch : Char;
    Strg : String;
    Ptr : integer;
    Quit : boolean;
procedure Delete;
begin
    if Ptr <> 1 then
        begin Ptr := Ptr-1;
            write(Ch);
            write(' ');
            write(Ch);
            Strg := copy(Strg,1,length(Strg)-1);
        end
    else write(^G);
end;
procedure Character;
begin
    if Ptr>Count then
        write(^G)
    else if (strg[1] in['0'..'9']) then

```

```

Key := ReadKey;
if Key = #0 then
begin
    FuncKey := True;
    Key := ReadKey;
end
else
    FuncKey := False;
end;

{===== ReadChar =====}
procedure ReadChar(var St : String;Count:integer);
var Ch : Char;
    Strg : String;
    Ptr : integer;
    Quit : boolean;
procedure Delete;
begin
    if Ptr <> 1 then
        begin Ptr := Ptr-1;
            write(Ch);
            write(' ');
            write(Ch);
            Strg := copy(Strg,1,length(Strg)-1);
        end
    else write(^G);
end;
procedure Character;
begin
    if Ptr>Count then
        write(^G)
    else if (strg[1] in['0'..'9']) then

```

```

        begin
            write(^G);
            Strg := Strg;
        end
    else begin
        write(Ch);
        Strg := Strg+Ch;
        Ptr := Ptr+1;end;
end;
begin
    Ptr := 1;
    Strg := '';
    Quit := false;
    repeat
        Ch := ReadKey;
        Strg[Ptr] := ch;
        case Ch of
            ^[ : begin Strg := #13; Quit := true;end;
            ^M : begin if strg = '' then strg := #13; Quit := true; end;
            ^H : Delete;
            ' '..'z' : Character;
        end;
    until Quit;
    St := Strg;
end;
{===== ReadChar1=====}
procedure ReadChar1(var St : String;Count:integer);
var Ch : Char;
    Strg : String;
    Ptr : integer;
    Quit : boolean;

```

```
procedure Delete;
```

```
begin
```

```
  if Ptr <> 1 then
```

```
    begin Ptr := Ptr-1;
```

```
      write(Ch);
```

```
      write(' ');
```

```
      write(Ch);
```

```
      Strg := copy(Strg,1,length(Strg)-1);
```

```
    end
```

```
  else write('^G');
```

```
end;
```

```
procedure Character;
```

```
begin
```

```
  if Ptr>Count then
```

```
    write('^G)
```

```
  else begin
```

```
    write(Ch);
```

```
    Strg := Strg+Ch;
```

```
    Ptr := Ptr+1;
```

```
  end;
```

```
end;
```

```
begin
```

```
  Ptr := 1;
```

```
  Strg := '';
```

```
  Quit := false;
```

```
  repeat
```

```
    Ch := ReadKey;
```

```
    Strg[ptr] := ch;
```

```
    case Ch of
```

```
      ^[ : begin Strg := #13; Quit := true;end;
```

```

    ^M : begin if strg = '' then strg := #13; Quit := true; end;
    ^H : Delete;
    ' '..'z' : Character;
end;
until Quit;
St := Strg;
end;

```

```

Procedure ReadReal(var value:real;count:integer;var code:integer);

```

```

Var MinusFlag,DecimalFlag : boolean;

```

```

    ch : char;

```

```

    strg : string;

```

```

    ptr : integer;

```

```

    quit : boolean;

```

```

Procedure Delete;

```

```

Begin

```

```

    if Ptr <> 1 then

```

```

        begin

```

```

            ptr := ptr-1;

```

```

            write(ch); gotoxy(wherex,wherey);write(' '); write(ch);

```

```

            if strg = '-' then

```

```

                MinusFlag := true;

```

```

            if strg[ptr] = '.' then

```

```

                DecimalFlag := true;

```

```

            strg := Copy(Strg,1,length(strg)-1);

```

```

        end

```

```

    else

```

```

        write(^G);

```

```

End;

```

```

Procedure Point;

```

```

Begin

```

```
    if DecimalFlag then
      begin
        write(ch);
        ptr := ptr+1;
        strg:=strg+ch;
        decimalFlag := false;
      end;
    End;
  Procedure Minus;
  Begin
    if MinusFlag and (ptr=1) then
      begin
        write(ch);
        ptr := ptr+1;
        strg:=strg+ch;
        minusFlag := false;
      end
    else
      write(^G);
    End;
  Procedure Number;
  Begin
    if ptr > count then
      write(^G)
    else
      begin
        write(ch);
        ptr := ptr+1;
        strg:=strg+ch;
      end;
    End;
```

Begin

```

ptr := 1;
strg := '';
value := 0.0;
minusFlag := true;
decimalFlag := true;
quit := false;
repeat
  ch := readkey;
  case ch of
    '[' : begin
            strg := #27;
            quit := true;
          end;
    'M' : quit := true;
    'H' : Delete;
    '-' : Minus;
    '.' : Point;
    '0'..'9': Number;
  else
    Write('^G');
  end; { case }
until Quit;
if Strg[1] = '.' then
  strg := '0' + strg;
if Strg[length(strg)] = '.' then
  strg := strg + '0';
Val(strg,value,code);
if code <> 0 then
  code := 255;
if (Strg = #27) and (Strg = '') then code :=1;

```

End;

{===== End of Check Real =====}

{Procedure for checking integer number. If not integer number, will make noise}

Procedure ReadInt(var value:integer;count:integer;var code :integer);

Var MinusFlag : boolean;

ch : char;

strg : string;

ptr : integer;

quit : boolean;

Procedure Delete;

Begin

if Ptr <> 1 then

begin

ptr := ptr-1;

write(ch); gotoxy(wherex,wherey);write(' '); write(ch);

if strg = '-' then

MinusFlag := true;

strg := Copy(Strg,1,length(strg)-1);

end

else

write(^G);

End;

Procedure Minus;

Begin

if (MinusFlag) and (Ptr=1) then

begin

write(ch);

ptr := ptr+1;


```
        strg:=strg+ch;
        minusFlag := false;
    end
    else
        write(^G);
End;
Procedure Number;
Begin
    if ptr > count then
        write(^G)
    else
        begin
            write(ch);
            ptr := ptr+1;
            strg:=strg+ch;
        end;
End;
Begin
    ptr := 1;
    strg := '';
    minusFlag := true;
    quit := false;
    repeat
        ch := readkey;
        case ch of
            ^[ : begin
                    strg := #27;
                    quit := true;
                end;
            ^M : quit := true;
            ^H : Delete;
```

```

    '-' : Minus;
    '0'..'9': Number;
else
    Write(^G);
end; { case }
until Quit;
Val(strg,value,code);
if code <> 0 then
    code := 255;
if (Strg = #27) and (Strg = '') then
    code := 1;
End;

procedure checkword(var line: string;var word,cnt : byte);
var k : integer;
begin
    word := 1;cnt := 0;
    for k := 1 to length(line) do
        begin
            if (line[k] <> ' ') and (line[k-1] = ' ') then
                if line[k] in['0'..'9','a'..'z','A'..'Z','/'] then
                    if (line[k-2] <> ' ') and (line[k+1] <> ' ') then
                        word := word + 1;
                    if word = 5 then
                        cnt := k-1;
                end;
        end;
    end;

begin
end.
```

โปรแกรม WIN.PAS

```
{Window Procedures
```

```
WindowOpen :- Make Window and draw box
WindowClose :- Remove Window
SetBoxStyle :- Select type of Box Window
SetWinHeader :- Set Header of Window
SetWinAttr :- Set Attribute of Box Window
SetHeadAttr :- Set Attribute of Header
SetCharAttr :- Set Attribute of Character in Window
SetWindow :- Set Window configuration
```

```
}
```

```
Unit win;
```

```
interface
```

```
uses Crt,Dos;
```

```
type ScreenLine = array[1..80] of integer;
```

```
ScreenArray = array[1..25] of ScreenLine;
```

```
ScreenBlock = array[1..2000] of integer;
```

```
WindowLink = ^WindowControlBlock;
```

```
WindowControlBlock = record
```

```
    X1,Y1,X2,Y2 : integer ;{Window boundary }
```

```
    X,Y : integer; { Cursor Location}
```

```
    ID : byte;
```

```
    BackLink : WindowLink;
```

```
    ScreenContents: ScreenBlock;
```

```
end;
```

```
HeadString = String[80];
```

```
const NoDisplay = $00;
```

```
LowDisplay = $07;
```

```
HighDisplay = $0F;
```

```

    UnderLineLow = $01;
    UnderLineHigh = $09;
    ReverseLow = $70;
    ReverseHigh = $78;
    ColorSeg = $B800;
    MonoSeg = $B000;
const Single = 1;
    Double = 2;
    BoxStyle      : byte = Single;
    AttrOfBox     : byte = HighDisplay;
    AttrOfWindow  : byte = HighDisplay;
    AttrOfHeader  : byte = HighDisplay;
    AttrOfChar    : byte = HighDisplay;
    HeaderOfWindow : HeadString = '';
    TypeOfBox     : array [1..2,1..8] of char =
                    ((#196,#179,#179,#196,#218,#191,#192,#217),
                    (#205,#186,#186,#205,#201,#187,#200,#188));

var VideoSeg      : word;
    Crttype       : byte Absolute $0040:$0049;
    ActiveWindow  : WindowLink;
    ScreenPtr     : ^ScreenArray;
    FixedSize     : integer;
    WindowCount   : byte;
    ErrorWindow   : byte;

procedure WindowBox(x1,y1,x2,y2:byte);
procedure WindowOpen(x1,y1,x2,y2:byte);
procedure WindowClose;
procedure InitWin;
procedure SetBoxStyle(Name:byte);

```

```

procedure SetWinHeader(St :HeadString);
procedure SetWinAttr(Name : byte);
procedure SetBoxAttr(Name : byte);
procedure SetHeadAttr(Name : byte);
procedure SetCharAttr(Name : byte);
procedure SetAttr(Attrib :byte);
procedure save_scr;
procedure get_oldscr;

```

implementation

```

procedure SetAttr(Attrib:byte);
begin
    TextAttr := Attrib;
end;

procedure WindowBox(x1,y1,x2,y2 : byte);
const
    Top      = 1; Left   = 2;
    Right   = 3; Bottom = 4;
    UpLeft  = 5; UpRight= 6;
    LoLeft  = 7; LoRight= 8;
var
    X,Y : byte;
begin
    window(x1,y1,x2,y2);
    SetAttr(AttrOfWindow);
    clrscr;
    window(1,1,80,25);
    SetAttr(AttrOfBox);
    { Top}
    gotoxy(x1,y1);
    write(.TypeOfBox[BoxStyle,UpLeft]);
    for X := X1 + 1 to X2 - 1 do

```

```

    write(.TypeOfBox[BoxStyle,Top]);
write(.TypeOfBox[BoxStyle,UpRight]);
{Sides}
for Y := y1+1 to y2-1 do
begin
    Gotoxy(x1,Y); Write(.TypeOfBox[BoxStyle,Left]); { Left side}
    Gotoxy(x2,Y); write(.TypeOfBox[BoxStyle,Right]);{ Right side}
end;
{ Bottom}
Gotoxy(x1,y2);
write(.TypeOfBox[BoxStyle,LoLeft]);
for X := x1+1 to x2-1 do
    write(.TypeOfBox[BoxStyle,Bottom]);
write(.TypeOfBox[BoxStyle,LoRight]);
{ Make it the current Window and Locate cursor to top}
SetAttr(attrOfHeader);
gotoxy((x1+x2-Length(HeaderOfWindow)) div 2,y1);
write(HeaderOfWindow);
window(x1+1,y1+1,x2-1,y2-1);
SetAttr(AttrOfChar);
end; { Proc. BoxWin}

```

```
{ Procedure WindowOpen
```

```
    Purpose      :- Make Window and draw box
```

```
    Input        :- Coordinate XY
```

```
    OutPut       :- none
```

```
}
```

```
procedure WindowOpen(x1,y1,x2,y2 : byte);
```

```
var    Block      : WindowLink;
```

```
    LineLength , WindowSize,
```

```
    i              : integer;
```

```

y                                     : byte;

begin

  LineLength := x2-x1+1;
  WindowSize := LineLength*(y2-y1+1)*2+FixedSize;
  {Check Window Valid }
  if (x2>80) or (y2>25) or (x2-x1<2) or (y2-y1<2) then
    ErrorWindow := 1
  else
    if (abs(MemAvail) < WindowSize) then
      ErrorWindow := 2
    else
      ErrorWindow := 0;
  if ErrorWindow = 0 then
    begin
      GetMem(Block, WindowSize);
      Block^.x1 := x1;
      Block^.x2 := x2;
      Block^.y1 := y1;
      Block^.y2 := y2;
      Block^.X := WhereX;
      Block^.Y := whereY;
      Block^.BackLink := ActiveWindow;
      ActiveWindow := Block;
      WindowCount := WindowCount + 1;
      Block^.ID := WindowCount;
      I := 1;
      for Y := y1 to y2 do
        begin
          Move(ScreenPtr^[Y,x1], Block^.ScreenContents[I], LineLength*2);
          I := I + LineLength;
        end;
    end;

```

```

        WindowBox(x1,y1,x2,y2);
    end;
end;

{ Procedure WindowClose
  Purpose  :- Remove Window
  Input    :- none
  Output   :- none
}

procedure WindowClose;
var Block      : WindowLink;
    LineLength ,WindowSize,
    I           : integer;
    y           : byte;
begin
    if ActiveWindow <> nil then
    begin
        Block      := ActiveWindow;
        LineLength := Block^.x2-Block^.x1+1;
        WindowSize := LineLength*(Block^.y2-Block^.y1+1)*2+FixedSize;
        I := 1;
        for Y := Block^.y1 to Block^.y2 do
        begin
            Move(Block^.ScreenContents[I],ScreenPtr^[Y,Block^.x1],LineLength*2)
            I := I + LineLength;
        end;
        ActiveWindow := Block^.BackLink;
        if ActiveWindow = nil then
            Window(1,1,80,25)
        else
            with ActiveWindow^ do Window(x1+1,y1+1,x2-1,y2-1);
    end;
end;

```



```
        gotoxy(Block^.X,Block^.Y);
        Freemen(Block,WindowSize);
    end;
end;

procedure InitWin;
var Mode : byte absolute $0040:$0049;
begin
    ActiveWindow := nil;
    FixedSize := SizeOf(WindowControlBlock)-SizeOf(ScreenBlock);
    if (Mode = 2) or (Mode = 3) then
        ScreenPtr := Ptr($B800,0)
    else
        ScreenPtr := Ptr($B000,0);
    window(1,1,80,25);
    windowCount := 0;
end;

procedure SetBoxStyle(Name : byte);
begin
    BoxStyle := Name;
end;

procedure SetWinHeader(St : HeadString);
begin
    HeaderOfWindow := st;
end;

procedure SetWinAttr(Name : byte);
begin
    AttrOfWindow := Name;
```

end;

```
procedure SetBoxAttr(Name : byte);
```

```
begin
```

```
    AttrOfBox := Name;
```

```
end;
```

```
procedure SetHeadAttr(Name : byte);
```

```
begin
```

```
    AttrOfHeader := Name;
```

```
end;
```

```
procedure SetCharAttr(Name : byte);
```

176

```
begin
```

```
    AttrOfChar := Name;
```

```
end;
```

```
procedure save_scr;
```

```
var i:integer;
```

```
begin
```

```
    for i:=0 to 3999 do
```

```
        begin
```

```
            mem[$b400:i]:=mem[$b000:i];
```

```
            if ((i mod 2) = 0) then      {clear old screen}
```

```
                mem[$b000:i]:= $00
```

```
            else
```

```
                mem[$b000:i]:= $07;
```

```
            end;
```

```
end;
```

```
procedure get_oldscr;
var i : integer;
begin
    for i:=0 to 3999 do
        mem[$b000:i]:=mem[$b400:i];
    end;

procedure IdentifyCrt;
begin
    case CrtType of
        0..3 : VideoSeg := ColorSeg;
        7   : VideoSeg := MonoSeg;
    end;
end;

begin
    IdentifyCrt;
end. { of unit }
```

ໂປຣແກຣມ WINDO.PAS

```

unit windo;
interface
uses crt,dos;
type screen80 = array[1..25,1..80,1..2]of char;
var monoscreen : screen80 Absolute $b000:$0000;
    monosave    : screen80;
    Cursormode  : integer Absolute $0040:$0060;
    Vport       : integer Absolute $0040:$0063;
    attr,row,col : byte;
procedure wind_close(ch :char;x1,y1,x2,y2 : byte);
procedure open1(col1,col2,col3,col4,row1,row2,row3,row4:byte);
implementation

procedure winrow(ch :char;x1,y1,x2,y2,attr : byte);
begin
    for row := y1 to y2 do
        for col := x1 to x2 do
            begin
                monoscreen[row,col,1] := ch;
                monoscreen[row,col,2] := chr(attr);
            end;
        end;
    end;

end;

procedure wincol(ch:char;x1,y1,x2,y2,attr:byte);
begin
    for col := x1 to x2 do
        for row := y1 to y2 do
            begin
                monoscreen[row,col,1] := ch;
            end;
        end;
    end;
end;

```

```

        monoscreen[row,col,2] := chr(attr);
    end;
end;
procedure plotch(ch:char;col,row,attr:byte);
begin
    monoscreen[row,col,1] := ch;
    monoscreen[row,col,2] := chr(attr);
end;
procedure wind_close(ch:char;x1,y1,x2,y2:byte);
begin
    for row := y1 to y2 do
        for col := x1 to x2 do
            begin
                monoscreen[row,col,1] := ch;
                monoscreen[row,col,2] := #07;
            end;
        end;
    end;
end;

procedure open1(col1,col2,col3,col4,row1,row2,row3,row4:byte);
begin
    plotch(#218,col1,row1,7);
    winrow(#196,col2,row1,col3,row1,7);
    plotch(#191,col4,row1,7);
    plotch(#192,col1,row4,7);
    wincol(#179,col1,row2,col1,row3,7);
    wincol(#179,col4,row2,col4,row3,7);
    winrow(#196,col2,row4,col3,row4,7);
    plotch(#217,col4,row4,7);
end;
begin
end.

```

โปรแกรม LDIR.PAS

```

unit Ldir;
interface
uses Dos,crt;

procedure DisplayFileName;
implementation
procedure ConvertTime( Time : Longint;
                      var Day,Month,Year,Hour,Min,Sec : String);
var Dt : DateTime;
begin
    Unpacktime(time,Dt);
    str(Dt.Day :2,Day);
    str(Dt.Month :2,Month);
    str(Dt.Year :4,Year);
    str(Dt.Hour :2,Hour);
    str(Dt.Min :2,Min);
    str(Dt.Sec :2,Sec);
    if Day[1] = ' ' then Day[1] := '0';
    if Month[1] = ' ' then Month[1] := '0';
    if Hour[1] = ' ' then Hour[1] := '0';
    if Min[1] = ' ' then Min[1] := '0';
    if Sec[1] = ' ' then Sec[1] := '0';
end;

procedure DisplayDir(var j : byte;var DirInfo : SearchRec);
var Day,Month,Year,Hour,Min,Sec : String;
begin
    with DirInfo do begin
        ConvertTime(Time,Day,Month,Year,Hour,Min,Sec);
        Gotoxy( 5,j);write(name);
    end;
end;

```

```

if size = 0 then
  begin
    gotoxy(25,j);write('<PSU>');
  end
else
  Gotoxy(25,j);write(Size:8);
  Gotoxy(35,j);write(Day,'-',Month,'-',Year,' ');
  Gotoxy(48,j);write(Hour,':',Min,':',Sec);
end;
writeln;
end;
procedure Dir(Mask : string;var DirInfo : SearchRec);
var j : byte;
begin
  j := 2;
  FindFirst(Mask,AnyFile,DirInfo);
  while DosError = 0 do
    begin FindNext(DirInfo);
      if (DosError = 0) then
        begin
          if j = 2 then
            begin
              gotoxy(5,2);write('Directory of *.* ');
              j := j + 2;
            end;
          DisplayDir(j,DirInfo);
          j := j + 1 ;
        end;
      if (j = 18) or (DosError = 18) then
        begin gotoxy(5,19);
          write('press Enter when ready ...');readln; clrscr;
        end;
    end;
end;

```

```
        j := 2; end;
```

```
    end;
```

```
end;
```

```
procedure DisplayFileName;
```

```
var Path      : string;
```

```
    DirInfo   : SearchRec;
```

```
begin
```

```
    clrscr;
```

```
    Path := '*.*';
```

```
    Dir(Path, DirInfo);
```

```
end;
```

```
begin
```

```
end.
```


ໂປຼແກຼມ FI.PAS

```

unit fi;
interface
uses crt,windo;
type anystr = string[80];
var fva : text;
    fil : string[25];
    line,stfi : anystr;
procedure makefi(stfi : string;var fva :text);
implementation
function create(var fva : text):boolean;
begin
    {$I-}
    reset(fva); {$I+}
    create := IOresult = 0 ;
end;
procedure makefi(stfi : string; var fva: text);
begin
    if create(fva) then
        begin
            append(fva);
            writeln(fva,stfi);
            close(fva);
        end
    else
        begin
            rewrite(fva);
            writeln(fva,stfi);close(fva);end;end;begin
end.

```

โปรแกรม DISPLAY.PAS

```
unit Display;
```

```
interface
```

```
Uses      Crt,Win;
```

```
Type      Screen80 = array [1..25,1..80,1..2] of char;
```

```
           ScreenArray = array [1..25] of String[80];
```

```
Var        MonoScreen : Screen80 Absolute $B000:$000;
```

```
           ColoScreen : Screen80 Absolute $B800:$000;
```

```
           CharScreen : ScreenArray;
```

```
           CrtType     : Byte Absolute $0040:$0049;
```

```
           x,y,len     : byte;
```

```
           ScrFile     : File of ScreenArray;
```

```
           f           : text;
```

```
procedure Read_txt(txt_name : string);
```

```
implementation
```

```
Procedure Read_File( XX : string);
```

```
Begin
```

```
    Assign(ScrFile,XX);
```

```
    Reset(ScrFile);
```

```
    Read(ScrFile,CharScreen);
```

```
    close(ScrFile);
```

```
End;
```

```
Procedure Display_File;
```

```
Begin
```

```
    for y := 1 to 25 do
```

```
        begin
```

```
            len := Length(CharScreen[y]);
```

```
for x := 1 to len do
  if (CrtType = 7) then
    begin
      MonoScreen[y,x,1] := CharScreen[y][x];
      MonoScreen[y,x,2] := #15;
    end
  else
    begin
      ColoScreen[y,x,1] := CharScreen[y][x];
      ColoScreen[y,x,2] := #15;
    end;
  end;
end;

End;

procedure Read_txt(txt_name : string);
begin
  Read_File(txt_name);
  Display_File;
end;

Begin
End.
```

TRANSPAS.PAS

```

unit Trans;
interface
uses crt,keyboard,win,windo,fi;
type vlink = ^vlist;
      vlist = record  vname  : string[20];
                    vpnt   : vlink;
                    end;

var  Hvar,LastP,Pvar,Fvar  :vlink;
     f   : text;
     stfi : string;

procedure SelectN;
procedure SelecSamp;
procedure SelecLogi(var stlog : string);
procedure TrChange;
procedure Trcompute;
procedure Trif;
procedure Trcount;
procedure Trweight;
procedure Def_file(var st,str : string);
procedure Defix(var strfi : string);
procedure Defree(var strfi : string);
procedure DefMiss;
procedure DefValue;
procedure Defnme_var;
implementation
procedure pnter(valink : vlink);
begin
    if Hvar = nil then

```

```

        Hvar := valink
    else
        LastP^.vpnt := valink;
        LastP      := valink;
        LastP^.vpnt := nil;
    end;

procedure SelectN;
var  Inte : integer;
     stn,sint : string;
begin
    open1(27,28,73,74,6,7,9,10);
    gotoxy(35,6);writeln(' รายการกำหนดจำนวนชุดข้อมูล ');
    gotoxy(28,8);writeln(' จำนวนข้อมูลที่คัดเลือกมาวิเคราะห์ ');
    open1(62,63,70,71,7,8,8,9);
    gotoxy(64,8);readInt(Inte,6,code);
    str(inte,sint);
    if inte <> 0 then
        begin assign(f,'c:\sptool\ex.sps');
              stn := 'n = ' + sint + '.';
              makefi(stn,f);
        end;
    wind_close(' ',27,6,74,10);
end;

procedure SelecSamp;
var  Ratio : real;
     size,size1 : integer;
     strat,ssize,ssize1,stsamp : string;
begin
    open1(27,28,73,74,6,7,15,16);
    gotoxy(35,6);writeln(' วิธีการสุ่มตัวอย่าง ');

```

```

gotoxy(28,8);writeln('สัดส่วนตัวอย่าง');open1(48,49,63,64,7,8,8,9);
gotoxy(28,11);writeln('ขนาดของตัวอย่าง');open1(48,49,63,64,10,11,11,12);
gotoxy(28,14);writeln('ขนาดของประชากร');
open1(48,49,63,64,13,14,14,15);
gotoxy(50,8);ReadReal(Ratio,5,code);
if Ratio = 0.0 then
begin
gotoxy(50,11);ReadInt(Size,8,code);
str(size,ssize);
if Size <> 0 then
begin gotoxy(50,14);ReadInt(Size1,8,code);
str(size1,ssize1);
stsamp := 'sam ' + ssize + ' from ' + ssize1 + '.'
end;
end
else
begin str(ratio:4:2, strat);
stsamp := 'sam ' + strat + '.';
end;
if (Ratio <> 0.0) or (Size <> 0) then
begin assign(f,'c:\sptool\ex.sps');
makefi(stsamp,f); end;
wind_close(' ',27,6,74,16);
end;
procedure SelecLogi(var stlog : string);
var Names : string;
begin
open1(27,28,73,74,6,7,9,10);
gotoxy(34,6);writeln('เงื่อนไขทางตรรก ');
gotoxy(28,8);writeln('เงื่อนไข');open1(38,39,70,71,7,8,8,9);
gotoxy(40,8);readchar1(Names,30);

```

```

        assign(f,'c:\sptool\ex.sps');
        makefi(stlog,f);
    end;
wind_close(' ',27,6,74,10);
end;
procedure TrChange;
type datalink = ^datalist;
    datalist = record names      : string[10];
                   oldcode     : string[3];
                   newcode     : string[3];
                   next        : datalink;
    end;
var na,firstN,lastN      : datalink;
    h      : ^integer;
    stcod  : string;
procedure pnter1(datarec : datalink);
begin
    if FirstN = nil then
        FirstN := datarec
    else
        LastN^.next := datarec;
        LastN := datarec;
        LastN^.next := nil;
    end;
end;
procedure nchange;
var nme,olds,news      : string;
begin
    new(na);
    gotoxy(50,8);readchar(nme,10);
    if nme <> #13 then
        begin

```

```

    gotoxy(50,11);readchar1(Olds,3);
    gotoxy(50,14);readchar1(News,3);
end
else begin Olds := ''; News := ''; end;
with na^ do
begin names := nme;
      oldcode := olds;
      newcode := news;
end;
pnter1(na);
gotoxy(50,8);wind_close(' ',50,8,63,8);
gotoxy(50,11);wind_close(' ',50,11,63,11);
gotoxy(50,14);wind_close(' ',50,14,63,14);
end;
procedure outcode(fst : datalink);
var prenme : string[10];
begin
  stcod := 'reco '+fst^.names;
  prenme := fst^.names;
  repeat
    if fst^.names = prenme then
      if fst^.oldcode[1] in['0'..'9'] then
        stcod := stcod+' ('+fst^.oldcode+' = '+fst^.newcode+')'
      else
        stcod := stcod+' ("'+fst^.oldcode+'" = "'+fst^.newcode+'")'
    else
      begin
        makefi(stcod,f);
        stcod := '/'+fst^.names;
        if fst^.oldcode[1] in['0'..'9'] then
          stcod := stcod+' ('+fst^.oldcode+' = '+fst^.newcode+')'

```



```

    else
        stcod := stcod + ' (' + fst^.oldcode + ' = "' + fst^.newcode + '" )';
    end;
    prenme := fst^.names;
    fst := fst^.next;
until fst^.next = nil;
stcod := stcod + '.';
makefi(stcod, f);
end;
begin
    open1(27, 28, 73, 74, 6, 7, 15, 16);
    gotoxy(35, 6); writeln('      เปลี่ยนค่าของตัวแปร ');
    gotoxy(32, 8); writeln('ชื่อของตัวแปร'); open1(48, 49, 63, 64, 7, 8, 8, 9);
    gotoxy(32, 11); writeln('ค่าหรือรหัสเดิม'); open1(48, 49, 63, 64, 10, 11, 11, 12);
    gotoxy(32, 14); writeln('ค่าหรือรหัสใหม่'); open1(48, 49, 63, 64, 13, 14, 14, 15);
    mark(h);
    FirstN := nil;
    repeat
        nchange;
    until Na^.Names = #13 ;
    if FirstN^.Names <> #13 then
        begin assign(f, 'c:\sptool\ex.sps');
            outcode(firstN);
        end;
    wind_close(' ', 27, 6, 74, 16);
    release(h);
end;
procedure Trcompute;
type comlink = ^comlist;
    comlist = record name : string[10];
                comp : string[32];

```

```

                                next : comlink;
                                end;
var  first,last,compute  : comlink;
     Names,Math,stcom  : string;
     htop              : ^integer;
procedure pnter2(com  : comlink);
begin
    if first = nil then
        First := com
    else
        Last^.next := com;
        Last := com;
        Last^.next := nil;
    end;
end;
procedure Pcomp;
begin
    new(compute);
    gotoxy(41,12);readchar(Names,10);
    if Names <> #13 then
        begin
            gotoxy(41,15);readchar1(Math,32);
        end
    else Math := '';
    with compute^ do
        begin name := names;
              comp := math;
        end;
    pnter2(compute);
    gotoxy(41,12); wind_close(' ',41,12,76,12);
    gotoxy(41,15); wind_close(' ',41,15,76,15);
end;

```

```

procedure out_comp(fst : comlink);
begin
  repeat
    stcom := 'comp ' + fst^.name + ' = ' + fst^.comp + '.';
    makefi(stcom,f);
    fst := fst^.next;
  until fst^.next = nil;
end;

begin
  open1(15,16,77,78,10,11,16,17);
  gotoxy(22,10); writeln('   สร้างตัวแปรใหม่โดยการคำนวณ ');
  gotoxy(17,12); writeln('ชื่อตัวแปรใหม่'); open1(39,40,76,77,11,12,12,13);
  gotoxy(17,15); writeln('นิพจน์ทางคณิตศาสตร์');
  open1(39,40,76,77,14,15,15,16);
  mark(htop);
  first := nil;
  repeat
    pcomp;
  until compute^.Name = #13 ;
  if first^.Name <> #13 then
    begin assign(f,'c:\sptool\ex.sps');
           out_comp(first);
    end;
  wind_close(' ',15,10,78,17);
  release(htop);
end;

procedure Trif;
type  iflink  = ^iflist;
      iflist  = record
                                vif  : string;
                                vlog : string;

```

```

                                val  : string;
                                next  : iflink;
                                end;
var  last,first,va   : iflink;
     htop           : ^integer;
procedure Pif;
var  Logi,Names,Value : string;
begin
    new(va);
    gotoxy(50,8);readchar(Names,10);
    if Names <> #13 then
        begin
            gotoxy(50,11);readchar(Logi,15);
            gotoxy(55,14);readchar1(Value,15);
        end
    else begin Logi := ''; Value := ''; end;
    with va^ do
        begin  vif  := names;
                vlog := Logi;
                val  := Value;
        end;
    if first = nil then
        First := va
    else
        Last^.next := va;
    Last := va;
    Last^.next := nil;
    wind_close(' ',50,8,63,8);
    wind_close(' ',50,11,63,11);
    wind_close(' ',55,14,66,14);
end;
```

```

begin
  open1(27,28,73,74,6,7,15,16);
  gotoxy(32,6);writeln(' สร้างตัวแปรใหม่โดยเงื่อนไข ');
  gotoxy(28,8);writeln(' ชื่อตัวแปรใหม่ ');open1(48,49,63,64,7,8,8,9);
  gotoxy(28,11);writeln(' นิพจน์ทางตรรก ');open1(48,49,63,64,10,11,11,12);
  gotoxy(28,14);writeln(' ค่าหรือนิพจน์ที่กำหนดให้ ');
  open1(53,54,66,67,13,14,14,15);
  mark(htop);
  First := nil;
  repeat pif until va^.vif = #13;
  wind_close(' ',27,6,74,16);
  release(htop);
end;

procedure Trcount;
type cntlink = ^cntlist;
   cntlist = record name : string[10];
                   vi   : string[19];
                   valu : string[15];
                   next : cntlink;
           end;
var htop : ^integer;
    first,last,vcount : cntlink;
    stcou : string;
procedure Pcount;
var Names,Ifs,Value : string;
begin
  new(vcount);
  gotoxy(54,12);readchar(Names,10);
  if Names <> #13 then
    begin
      gotoxy(54,15);readchar(Ifs,19);

```

```

        gotoxy(54,18);readchar1(Value,15);
    end
else begin Ifs := ''; value := ''; end;
with vcount^ do
    begin name := names;
          vi := ifs;
          valu := Value;
    end;
if first = nil then
    First := vcount
else
    Last^.next := vcount;
Last := vcount;
Last^.next := nil;
wind_close(' ',54,12,69,12);
wind_close(' ',54,15,74,15);
wind_close(' ',54,18,74,18);
end;
procedure out_count(fst : cntlink);
var pren : string[10];
begin
    stcou := 'count ';
    pren := fst^.name;
    repeat
        if fst^.name = pren then
            if fst^.valu[1] in['0'..'9'] then
                stcou := stcou+fst^.name+' = '+fst^.vi+' ('+fst^.valu+') '
            else
                stcou := stcou+fst^.name+' = '+fst^.vi+' ("'+fst^.valu+'" ) '
            else
                begin

```

```

    makefi(stcou,f);
    if fst^.valu[1] in ['0'..'9'] then
        stcou := '/'+fst^.name+' = '+fst^.vi+' ('+fst^.valu+') '
    else
        stcou := '/'+fst^.name+' = '+fst^.vi+' ("'+fst^.valu+'") ';
    end;
    pren := fst^.name;
    fst := fst^.next;
until fst^.next = nil;
stcou := stcou + '.';
makefi(stcou,f);
end;
begin
    open1(12,13,76,77,10,11,19,20);
    gotoxy(23,10);writeln('      สร้างตัวแปรใหม่โดยการนับ ');
    gotoxy(14,12);writeln('ชื่อตัวแปรใหม่');open1(52,53,69,70,11,12,12,13);
    gotoxy(14,15);writeln('รายชื่อตัวแปรที่เป็นเงื่อนไขในการนับ');
    open1(52,53,74,75,14,15,15,16);
    gotoxy(14,18);writeln('กำหนดค่ารายชื่อตัวแปร');
    open1(52,53,74,75,17,18,18,19);
    mark(htop);
    first := nil;
    repeat Pcount until vcount^.name = #13;
    if first^.name <> #13 then
        begin assign(f,'c:\sptool\ex.sps');
            out_count(first);
        end;
    wind_close(' ',12,10,77,20);
    release(htop);
end;
procedure Trweight;

```

```

var htop : ^integer;
    stweigh : string;
procedure Pweight ;
var Names : string;
begin
    new(Pvar);
    gotoxy(45,8);readchar(Names,10);
    Pvar^.vname := names;
    pnter(Pvar);
    wind_close(' ',45,8,63,8);
end;
procedure outweigh(fst : vlink);
begin
    repeat
        if fst^.vname <> 'off' then
            stweigh := 'weig by ' + fst^.vname + '.';
        else
            stweigh := 'weig off.';
        makefi(stweigh,f);
        fst := fst^.vpnt;
    until fst^.vpnt = nil;
end;
begin
    open1(27,28,69,70,6,7,9,10);
    gotoxy(33,6);writeln('      กำหนดน้ำหนักของตัวแปร ');
    gotoxy(28,8);writeln('ชื่อตัวแปร');open1(43,44,63,64,7,8,8,9);
    Hvar := nil;
    mark(htop);
    repeat Pweight until Pvar^.vname = #13;
    if Hvar^.vname <> #13 then
        begin assign(f,'c:\sptool\ex.sps');

```



```

        outweight(hvar);
    end;
wind_close(' ',27,6,70,10);
release(htop);
end;
procedure Def_file(var st,str : string);
begin
    open1(22,23,75,76,10,11,14,15);
    open1(41,42,70,71,12,13,13,14);
    gotoxy(24,11);write('เก็บเพิ่มข้อมูล แบบ ASCII หรือ System file [A/S] [ ]');
    gotoxy(24,13);write('ชื่อเพิ่มข้อมูล');
    gotoxy(74,11);readchar(str,1);
    if str <> #13 then
        begin
            gotoxy(43,13);readchar(st,20);
        end
    else st := #13;
    wind_close(' ',22,10,76,15);
end;
{-----}
procedure defix(var strfi : string);
type fixlink = ^fixlist;
    fixlist = record fname : string[15];
                    ftype : string[4];
                    fpos  : string[8];
                    next  : fixlink;
    end;
var first,last,p : fixlink;
    h : ^integer;
procedure win_fix;
begin

```

```

gotoxy(12,13);writeln('ชื่อด้านแปร');open1(34,35,55,56,12,13,13,14);
gotoxy(12,16);writeln('ประเภทข้อมูลของตัวแปร *');
open1(40,41,55,56,15,16,16,17);
gotoxy(12,19);writeln('ตำแหน่งคอลัมน์ของตัวแปร');
open1(40,41,55,56,18,19,19,20);
gotoxy(14,20); writeln('* a : ตัวอักษร หรือ ');
gotoxy(14,21); writeln('ใส่จำนวนทศนิยม ถ้าจำนวนทศนิยมเป็น 0: เคาะผ่าน');
end;
procedure Pfix;
var Names,Types,Posi,ch : string;
    Name1,Type1,Posi1 : string;
begin
    new(p);
    gotoxy(36,13);readchar(Names,15);
    if names <>#13 then
        begin
            gotoxy(42,16);readchar1(Types,4);
            gotoxy(42,19);readchar1(Posi,8);
            wind_close(' ',12,12,62,21);
            gotoxy(14,13);write('ชื่อด้านแปร      = ',Names);
            gotoxy(14,15);
            if types <> #13 then
                if types[1] <> 'a' then
                    write('ประเภทข้อมูล      = numeric ทศนิยม ',types)
                else write('ประเภทข้อมูล      = a')
            else write('ประเภทข้อมูล      = numeric');
            gotoxy(14,17);write('ตำแหน่งคอลัมน์ = ',posi);
            gotoxy(20,19);write('ข้อมูลถูกต้องหรือไม่ [Y/N] [ ]');
            gotoxy(48,19);readchar(ch,1);
            if ch[1] in['n','N'] then
                begin

```

```

    wind_close(' ',12,12,62,21);
    win_fix;
    gotoxy(36,13);readchar(Name1,15);
    if name1 <> #13 then
        names := name1
    else begin gotoxy(36,13);write(names);end;
    gotoxy(42,16);readchar1(Type1,4);
    if type1 <> #13 then
        types := type1
    else begin gotoxy(42,16);write(types);end;
    gotoxy(42,19);readchar1(Posi1,8);
    if Posi1 <> #13 then
        posi := posi1
    else begin gotoxy(42,19);write(posi);end;
    end;
end
else begin types := ''; Posi := ''; end;
with p^ do
    begin fname := names;
        ftype := types;
        fpos := posi;
    end;
if first = nil then
    First := p
else
    Last^.next := p;
Last := p;
Last^.next := nil;
wind_close(' ',12,12,62,21);
end;
procedure outfix(first : fixlink);

```

```

var i : integer;
begin
  strfi := strfi + '/';
  i := 0;
  repeat
    if first^.ftype = #13 then
      strfi := strfi + first^.fname + ' ' + first^.fpos + ' '
    else
      strfi := strfi + first^.fname + ' ' + first^.fpos + '(' + first^.ftype + ')'
      first := first^.next;
      i := i + 1;
      if (i in [3,7,10,14,18,22,26]) then
        begin makefi(strfi,f);
              strfi := '';
        end;
      if (i in [3,7,10,14,18,22,26]) and (first^.fname = #13) then
        strfi := '';
    until first^.next = nil;
    strfi := strfi + '.';
    makefi(strfi,f);
end;
begin
  open1(10,11,62,63,11,12,21,22);
  gotoxy(22,11);writeln(' กำหนดตำแหน่งคงที่ ');
  mark(h);
  First := nil;
  repeat win_fix;pfix; until p^.fname = #13 ;
  if (First^.fname <> #13) and (strfi <> '') then
    begin assign(f,'c:\sptool\ex.sps');
          erase(f);
          outfix(first);
    end;
end;

```

```

    end;
    wind_close(' ',10,11,63,22);
    release(h);
end;
{-----}
procedure defree(var strfi : string);
type   freelnk  = ^freelist;
       freelist = record   freenme  : string[15];
                           freetype : string[6];
                           next     : freelnk;
                       end;
var    first,last,freeva : freelnk;
       h      : ^integer;
procedure win_free;
begin
    gotoxy(13,13);writeln(' ชื่อตัวแปร');open1(30,31,55,56,12,13,13,14);
    gotoxy(13,16);writeln(' ประเภทตัวแปร* ');open1(30,31,55,56,15,16,16,17);
    gotoxy(13,18);writeln('* A สำหรับตัวอักษร หรือ ระบุความกว้างของตัวแปรที่');
    gotoxy(13,19);writeln(' เป็นข้อความ');
end;
procedure Pfree;
var   Names,Types,ch   : string;
       Name1,Type1    : string;
begin
    new(freeva);
    gotoxy(32,13);readchar(Names,15);
    if Names <> #13 then
        begin
            gotoxy(32,16);readchar1(Types,6);
            repeat
                wind_close(' ',13,12,62,19);

```

```

gotoxy(15,13);write('ชื่อตัวแปร = ',names);
gotoxy(15,15);
if types <> #13 then
  if types[1] <> 'a' then
    write('ประเภทตัวแปร = a',types)
  else write('ประเภทตัวแปร = a1 ')
else write('ประเภทตัวแปร = numeric');
gotoxy(20,17);write('ข้อมูลถูกต้องหรือไม่ [Y/N] [ ]');
gotoxy(48,17);readchar(ch,1);
if ch[1] in['n','N'] then
  begin
    wind_close(' ',13,13,62,19);
    win_free;
    gotoxy(32,13);readchar(Name1,15);
    if Name1 <> #13 then
      names := name1
    else begin gotoxy(32,13);write(names); end;
    gotoxy(32,16);readchar1(Type1,6);
    if Type1 <> #13 then
      types := type1
    else begin gotoxy(32,16);write(types); end;
  end;
until ch[1] in['y','Y'];
end
else types := '';
with freeva^ do
  begin  freeenme := names;
        freetype := types;
  end;
if first = nil then
  First := Freeva

```

```

else
  Last^.next := freeva;
Last := freeva;
Last^.next := nil;
wind_close(' ',13,13,62,19);
end;
procedure outfree(first : freelnk);
var i : integer;
begin
  strfi := strfi + 'free /';
  i := 1;
  repeat
    if first^.freetype = 'a' then
      strfi := strfi + first^.freenme + ' (' + first^.freetype + '1) '
    else if first^.freetype = #13 then
      strfi := strfi + first^.freenme + ' '
    else
      strfi := strfi + first^.freenme + ' (a'+ first^.freetype + ') '
    first := first^.next;
    i := i + 1;
    if (i in[4,10,15,20,25]) then
      begin makefi(strfi,f);
        strfi := '';
      end;
  until first^.next = nil;
  strfi := strfi + '.';
  makefi(strfi,f);
end;
begin
  open1(10,11,62,63,11,12,19,20);
  gotoxy(18,11);writeln(' กำหนดตำแหน่งคอลัมน์แบบอิสระ');

```

```

first := nil;
repeat win_free; Pfree; until freeva^.freenme = #13;
if (first^.freenme<>#13) and (strfi <> '')then
  begin assign(f,'c:\sptool\ex.sps');
         erase(f);
         outfree(first);
  end;
wind_close(' ',10,11,63,20);
release(h);
end;

procedure DefMiss;
type Mislink = ^Mislist;
   Mislist = record MisName : String[15];
                MisId   : String[10];
                next    : Mislink;
   end;

var lst,frst,Misva : Mislink;
    h : ^integer;
    stmis : string;

procedure Pmiss;
var Names,Id : string;
begin
  new(Misva);
  gotoxy(42,13);readchar(Names,15);
  if Names <> #13 then
    begin
      gotoxy(42,16);readchar1(Id,10);
    end
  else Id := '';
  with Misva^ do

```



```

begin  MisName := Names ; MisId := Id; end;
if frst = nil then
  Frst := Misva
else
  Lst^.next := Misva;
Lst := Misva;
Lst^.next := nil;
wind_close(' ',42,13,58,13);wind_close(' ',42,16,58,16);
end;
procedure outmiss(fst : mislink);
var i : integer;
begin
  stmis := 'mis val ';
  i := 1;
  repeat
    if i = 1 then
      if fst^.Misid[1] in['0'..'9'] then
        stmis := stmis + fst^.Misname + ' (' + fst^.Misid + ' ) '
      else
        stmis := stmis + fst^.Misname + ' (" + fst^.MisId + ") '
      else
        if fst^.Misid[1] in['0'..'9'] then
          stmis := stmis + '/' + fst^.Misname + ' (' + fst^.Misid + ' ) '
        else
          stmis := stmis + '/' + fst^.Misname + ' (" + fst^.MisId + ") '
        i := i+1;
        fst := fst^.next;
      until  fst^.next = nil;
      stmis := stmis + '.';
      makefi(stmis,f);
    end;
end;

```

```

begin
  open1(10,11,64,65,11,12,17,18);
  gotoxy(20,11);writeln('      ค่าไม่สมบูรณ์ให้กับตัวแปร ');
  gotoxy(13,13);writeln(' ชื่อตัวแปร');open1(40,41,58,59,12,13,13,14);
  gotoxy(13,16);writeln(' รหัสแทนค่าที่ไม่สมบูรณ์');
  open1(40,41,58,59,15,16,16,17);
  mark(h);
  Frst := nil;
  repeat Pmiss until Misva^.MisName = #13;
  if Frst^.MisName <> #13 then
    begin assign(f,'c:\sptool\ex.sps');
           outmiss(frst);
    end;
  wind_close(' ',10,11,65,18);
  release(h);
end;

procedure DefValue;
type Delink = ^Delist;
   Delist = record Denme : string[15];
                Deval  : string[6];
                Delab  : string[10];
                next   : Delink;
            end;

var top : ^integer;
    frst,lst,Dvalue : Delink;
    strval : string;

procedure Pvalu ;
var Names,Valu,Add : string;
begin
  new(Dvalue);
  gotoxy(37,13);readchar(Names,15);

```

```

if Names <> #13 then
  begin gotoxy(37,16);readchar1(valu,6);
        gotoxy(37,19);readchar(Add,10);
  end
else begin valu := ''; Add := ''; end;
with Dvalue^ do
  begin Denme := Names;
        Deval := Valu;
        Delab := Add;
  end;
if frst = nil then
  Frst := Dvalue
else
  Lst^.next := Dvalue;
Lst := Dvalue;
Lst^.next := nil;
wind_close(' ',37,13,58,13);wind_close(' ',37,16,58,16);
wind_close(' ',37,19,58,19);
end;
procedure outval(frst : Delink);
var prenme : string;
begin
  strval := 'val label '+ frst^.denme;
  prenme := frst^.denme;
  repeat
    if frst^.denme = prenme then
      if frst^.deval[1] in['0'..'9'] then
        strval := strval + ' '+ frst^.deval + ' "' + frst^.delab + '" '
      else
        strval := strval + ' "' + frst^.deval + '" "' + frst^.delab + '" '
    else

```

```

    if frst^.deval[1] in['0'..'9'] then
    strval := strval + '/' + frst^.denme + ' ' + frst^.deval +
        ' "' + frst^.delab + '" ' else
    strval := strval + '/' + frst^.denme + ' "' + frst^.deval +
        '" "' + frst^.delab + '" ';
    frst := frst^.next; prenme := frst^.denme;
until frst^.next = nil;
strval := strval + '.';
makefi(strval, f);
end;
begin
    open1(10, 11, 64, 65, 11, 12, 20, 21);
    gotoxy(20, 11); writeln('      รายละเอียดเกี่ยวกับค่าตัวแปร ');
    gotoxy(13, 13); writeln('ชื่อตัวแปร '); open1(35, 36, 60, 61, 12, 13, 13, 14);
    gotoxy(13, 16); writeln('ค่าของตัวแปร'); open1(35, 36, 60, 61, 15, 16, 16, 17);
    gotoxy(13, 19); writeln('ส่วนขยายของตัวแปร');
    open1(35, 36, 60, 61, 18, 19, 19, 20);
    mark(top);
    frst := nil;
    repeat Pvalu until Dvalue^.Denme = #13;
    if (frst^.Denme <> #13) and (frst^.Deval <> #13) then
        begin assign(f, 'c:\sptool\ex.sps');
            outval(frst);
        end;
    wind_close(' ', 10, 11, 65, 21);
    release(top);
end;
procedure Defnme_var;
type Delink = ^Delist;
    Delist = record    Dename : string[15];
                        DeAdd  : String[30];

```

```

                                next   : Delink;

                                end;

var   top   : ^integer;
      frst,lst,Denme  : Delink;
      strvar : string;

procedure Pname ;
var   Names,Add   : string;
begin
  new(Denme);
  gotoxy(37,13);readchar(Names,15);
  if Names <> #13 then
    begin
      gotoxy(37,16);readchar(Add,30);
    end
  else Add := '';
  with Denme^ do
    begin   Dename  := Names;
           DeAdd   := Add;
    end;
  if frst = nil then
    Frst := Denme
  else
    Lst^.next := Denme;
  Lst := Denme;
  Lst^.next := nil;
  wind_close(' ',37,13,67,13);wind_close(' ',37,16,67,16);
end;

procedure outvar(frst : delink);
var   i : integer;
begin
  i := 1;

```

```

strvar := 'var lab ';
repeat
  if i = 1 then
    strvar := strvar + frst^.dename + ' "' + frst^.deadd + '" '
  else
    strvar := strvar + '/' + frst^.dename + ' "' + frst^.deadd + '" ';
  i := i+1;
  frst := frst^.next;
until frst^.next = nil;
strvar := strvar + '.';
makefi(strvar,f);
end;
begin
  open1(14,15,70,71,11,12,17,18);
  gotoxy(24,11);writeln(' รายละเอียดของชื่อตัวแปร ');
  gotoxy(16,13);writeln('ชื่อตัวแปร');open1(35,36,67,68,12,13,13,14);
  gotoxy(16,16);writeln('ส่วนขยายของตัวแปร');open1(35,36,67,68,15,16,16,17);
  mark(top);
  frst := nil;
  repeat Pname until Denme^.Dename = #13;
  if (frst^.Dename <> #13) and (frst^.DeAdd <> #13) then
    begin assign(f,'c:\sptool\ex.sps');
      outvar(frst);
    end;
  wind_close(' ',14,11,71,18);
  release(top);
end;
begin
end.

```

โปรแกรม PROC.PAS

```

unit proc;
interface
uses crt, win, keyboard, windo, fi;
type vlink = ^vlist;
      vlist = record  vname : string[10];
                    vpnt  : vlink;
      end;
var  LastP : vlink;
     f     : text;
procedure Anovar(var stano : string);
procedure devi(var stano : string);
procedure Covar(var stano : string);
procedure Corrvar(var stcorr:string);
procedure Crossvar(var stcros:string;var j:byte);
procedure Descript_var(var stdes : string;var j : byte);
procedure Frevar(var st : string;var j : byte);
procedure Fregraph(var st : string);
procedure Fpercent(var st : string);
procedure Meanvar(var stmean1 : string;var j:byte);
procedure MeanGr(var stmean2 : string);
procedure Onewayvar(var st : string);
procedure OnewayGr(var st : string);
procedure OnewayVec(var st : string);
procedure plot_var(var st : string);
procedure Regression_var(var streg : string);
procedure Tind_va(var str1 : string);
procedure T2gr(var str2 : string);
procedure Tdep_va(var stdep : string);
procedure Npar_with(var stpar : string);

```

```

procedure Npva(var stn : string);
procedure Nby(var stn : string);
procedure Remo_var(var stn : string);
implementation
procedure pnter(var hvar:vlink;valink : vlink);
begin
    if Hvar = nil then
        Hvar := valink
    else
        LastP^.vpnt := valink;
    LastP      := valink;
    LastP^.vpnt := nil;end;
procedure anovar(var stano : string);
var Names : string;anvar,hvar : vlink;top : ^integer;
begin
    open1(36,37,75,76,7,8,10,11);
    gotoxy(39,7);writeln('ตัวแปรที่วิเคราะห์ความแปรปรวน');
    gotoxy(38,9);writeln('ชื่อตัวแปร');open1(50,51,70,71,8,9,9,10);
    mark(top);hvar := nil;
    repeat
        new(anvar);
        gotoxy(52,9);readchar(Names,10);
        anvar^.vname := names;pnter(hvar,anvar);
        wind_close(' ',52,9,70,9);
    until anvar^.vname = #13;
    if hvar^.vname <> #13 then
        begin
            stano := 'ano var ';
            repeat
                stano := stano + hvar^.vname + ' ';
                hvar := hvar^.vpnt;
            until hvar = nil;
        end;
    end;
end;

```



```

    until hvar^.vpnt = nil;end;
wind_close(' ',36,7,76,11);release(top);end;
procedure devi(var stano : string);
type devlink = ^devlist;
    devlist = record devar : string[10];
                devmax : integer;
                devmin : integer;
                next   : devlink;
            end;
var Names,dmax,dmin : string;
    mins,maxs : integer;
    top       : ^integer;
    head,last,dev : devlink;
begin
    open1(36,37,75,76,7,8,12,13);
    gotoxy(40,7);writeln('ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่ม');
    gotoxy(38,9);writeln('ชื่อตัวแปร');open1(55,56,68,69,8,9,9,10);
    gotoxy(38,11);writeln('ค่าต่ำสุด      [      ]');
    gotoxy(38,12);writeln('ค่าสูงสุด      [      ]');
    mark(top);
    head := nil;
    repeat
        new(dev);
        gotoxy(57,9);readchar(Names,10);
        if Names <> #13 then
            begin gotoxy(57,11);readint(mins,10,code);
                    gotoxy(57,12);readint(maxs,10,code);
            end
        else begin mins := 0; maxs := 0; end;
    until wind_close(' ',57,9,68,9);
    wind_close(' ',57,11,66,11);
end;

```

```

wind_close(' ',57,12,66,12);
with dev^ do
    begin devar := names;
           devmax := maxs;
           devmin := mins;
    end;
if Head = nil then
    Head := Dev
else
    Last^.next := dev;
    Last      := dev;
    Last^.next := nil;
until dev^.devar = #13;
if head^.devar <> #13 then
    stano := 'by ';
    repeat
        str(head^.devmin,dmin);
        str(head^.devmax,dmax);
        stano := stano+head^.devar+'('+dmin+', '+dmax+') ';
        head := head^.next;
    until head^.next = nil;
wind_close(' ',36,7,76,13);
release(top);
end;
procedure covar(var stano : string);
var Names : string;
    top    : ^integer;
    cov,hvar : vlink;
begin
    open1(36,37,75,76,7,8,10,11);
    gotoxy(39,7);writeln('ตัวแปรที่เป็นตัวแปรร่วม');

```

```

gotoxy(38,9);writeln('ชื่อตัวแปร');open1(53,54,68,69,8,9,9,10);
mark(top);
hvar := nil;
repeat
    new(cov);
    gotoxy(55,9);readchar(Names,10);
    cov^.vname := names;
    pnter(hvar,cov);
    wind_close(' ',55,9,67,9);
until cov^.vname = #13;
if hvar^.vname <> #13 then
    begin
        stano := 'with ';
        repeat
            stano := stano + hvar^.vname + ' ';
            hvar := hvar^.vpnt;
        until hvar^.vpnt = nil;
    end;
wind_close(' ',36,7,76,11);
release(top);
end;
procedure Corrvar(var stcorr : string);
var top : ^integer;
    Pvar,Fvar,head,fst : vlink;
procedure Cor_va;
var va,val : string;
begin
    new(Pvar);
    gotoxy(56,9);readchar(va,10);
    Pvar^.vname := va;
    pnter(fst,Pvar);

```

```

if (va = #13) and (fst^.vname <> #13) then
  repeat
    new(Fvar);
    gotoxy(56,12);readchar(va1,10);
    Fvar^.vname := va1;
    pnter(head,Fvar);
    wind_close(' ',56,12,69,12);
  until Fvar^.vname = #13 ;
  wind_close(' ',56,9,69,9);
end;
procedure out_corr(fst,head:vlink);
begin
  stcorr := 'corr ';
  repeat
    stcorr := stcorr + fst^.vname + ' ';
    fst := fst^.vpnt;
  until fst^.vpnt = nil ;
  if head^.vname <> #13 then
    begin
      stcorr := stcorr + 'with ';
      repeat
        stcorr := stcorr + head^.vname + ' ';
        head := head^.vpnt;
      until head^.vpnt = nil ;
    end;
end;
begin
  open1(36,37,72,73,7,8,13,14);
  gotoxy(44,7);write(' ตัวแปร ');
  gotoxy(38,9);write('หาความสัมพันธ์');
  gotoxy(38,10);write('ของตัวแปร');

```

```

open1(54,55,69,70,8,9,9,10);
gotoxy(38,12);write('กับตัวแปร');
open1(54,55,69,70,11,12,12,13);
mark(top);
fst := nil; head := nil;
repeat Cor_va until Pvar^.vname = #13;
if fst^.vname <> #13 then
    out_corr(fst,head);
wind_close(' ',36,7,73,14);
release(top);
end;
procedure Crossvar(var stcros : string;var j : byte);
var htop      : ^integer;
    Cross,Fvar,head,hvar : vlink;
procedure getlink;
var va,val : string;
begin
    new(Cross);
    gotoxy(58,9);readchar(va,10);
    Cross^.vname := va;
    pnter(hvar,Cross);
    if va = #13 then
        repeat
            new(Fvar);
            gotoxy(58,12);readchar(val,10);
            Fvar^.vname := val;
            pnter(head,Fvar);
            wind_close(' ',58,12,68,12);
        until Fvar^.vname = #13 ;
    wind_close(' ',58,9,68,9);
end;

```

```

procedure out_cros(fst,head:vlink);
begin
  repeat
    stcros := stcros + fst^.vname + ' ';
    fst := fst^.vpnt;
until fst^.vpnt = nil ;
if head^.vname <> #13 then
  begin
    stcros := stcros + 'by ';
    repeat
      stcros := stcros+ head^.vname + ' ';
      head := head^.vpnt;
    until head^.vpnt = nil ;
  end;
end;
begin
  mark(htop);
  open1(36,37,71,72,7,8,13,14);
  gotoxy(45,7);write(' ตัวแปร ');
  gotoxy(38,9);write('ตัวแปรทางด้านแนว');
  gotoxy(38,12);write('ตัวแปรทางด้านหลัก');
  open1(56,57,69,70,8,9,9,10);
  open1(56,57,69,70,11,12,12,13);
  Hvar := nil;
  head := nil;
  repeat
    getlink;
  until Cross^.vname = #13;
  if hvar^.vname <> #13 then
    begin
      if j = 1 then

```

```

        stcros := 'cros tab = '
    else
        stcros := '/';
    out_cros(hvar,head);
end;
wind_close(' ',36,7,72,14);
release(htop);
end;
procedure Descript_var(var stdes : string;var j : byte);
var  htop    : ^integer;
     Pvar,hvar : vlink;
     va : string;
begin
    mark(htop);
    open1(38,39,60,61,7,8,10,11);
    gotoxy(44,7);write(' ชื่อตัวแปร ');
    open1(41,42,55,56,8,9,9,10);
    Hvar := nil;
    repeat
        new(Pvar);
        gotoxy(43,9);readchar(va,10);
        Pvar^.vname := va;
        pnter(hvar,Pvar);
        wind_close(' ',43,9,54,9);
    until Pvar^.vname = #13;
    if hvar^.vname <> #13 then
        begin
            if j = 1 then
                stdes := 'des '
            else stdes := '';
            repeat

```

```

        stdes := stdes + hvar^.vname + ' ';
        hvar := hvar^.vpnt;
    until hvar^.vpnt = nil;
end;
release(htop);
wind_close(' ',36,7,66,11);
end;
procedure Frevar(var st : string;var j : byte);
var  htop    : ^integer;
     Fvar,hvar : vlink;
procedure getlink;
var  va : string;
begin
    new(Fvar);
    gotoxy(42,10);readchar(va,10);
    Fvar^.vname := va;
    pnter(hvar,Fvar);
    wind_close(' ',42,10,53,10);
end;
begin
    st := '';
    mark(htop);
    open1(38,39,58,59,8,9,11,12);
    gotoxy(43,8);writeln(' ชื่อตัวแปร ');
    Hvar := nil;
    open1(40,41,55,56,9,10,10,11);
    repeat
        getlink;
    until Fvar^.vname = #13;
    if Hvar^.vname <> #13 then
        begin

```



```

    if j = 1 then
        st := 'fre '
    else st := '';
    repeat
        st := st + Hvar^.vname + ' ';
        Hvar := Hvar^.vpnt;
    until Hvar^.vpnt = nil;
    end;
    release(htop);
    wind_close(' ',36,8,59,12);
end;

procedure Fregraph(var st : string);
var  maxf,minf,rangf,rperf : real;
     maxfre,minfre,rngfre,perf : string;
begin
    open1(36,37,75,76,8,9,12,13);
    st := '';
    gotoxy(45,8); writeln(' กราฟพื้นที่ ');
    gotoxy(39,9); writeln('ค่าต่ำสุด [ ]');
    gotoxy(39,10); writeln('ค่าสูงสุด [ ]');
    gotoxy(39,11); writeln('ช่วงของความถี่ [ ]');
    gotoxy(39,12); writeln('ช่วงของความถี่ในรูปแบบร้อยละ [ ]');
    gotoxy(66,9); readreal(minf,7,code);
    if minf <> 0.0 then
        begin
            gotoxy(66,10); readreal(maxf,7,code);
            gotoxy(66,11); readreal(rangf,7,code);
            str(maxf:07:2,maxfre);
            str(minf:07:2,minfre);
            str(rangf:07:2,rngfre);
            st := 'min('+ minfre +' ) max('+ maxfre + ' ) ';
        end;
    end;
end;

```

```

    if (rangf = 0) then
        begin
            gotoxy(66,12);readreal(rperf,5,code);
            str(rperf:05:2,perf);
            st := st + 'perc(' + perf + ')';
        end
    else
        st := st + 'freq(' + rngfre + ')';
    end;
    wind_close(' ',36,8,76,13);
end;

procedure Fpercent(var st:string);
type perlnk = ^perlist;
    perlist = record    Freal : real;
                        next  : perlnk;
                    end;
var num : real;
    perc,head,last : perlnk;
    htop : ^integer;
    streal : string;
begin
    mark(htop);
    head := nil;
    open1(36,37,74,75,8,9,9,10);
    gotoxy(38,9);write('ตำแหน่งที่ต้องการหาค่า [      ]');
    repeat
        new(perc);
        gotoxy(64,9);readreal(num,8,code);
        perc^.Freal := num;
        if head = nil then
            head := perc

```

```

    else
        last^.next := perc;
    last := perc;
    last^.next := nil;
    wind_close(' ',64,9,71,9);
until perc^.Freal = 0.0;
if head^.Freal <> 0.0 then
    repeat
        str(head^.Freal:8:2,streal);
        st := st + streal + ' ';
        head := head^.next;
    until head^.next = nil;
wind_close(' ',36,8,75,10);
release(htop);
end;
procedure Meanvar(var stmean1 : string;var j :byte);
var htop : ^integer;
    Mvar,hvar : vlink;
procedure getlink;
var v : string;
begin
    new(Mvar);
    gotoxy(42,10);readchar(v,10);
    Mvar^.vname := v;
    pnter(hvar,Mvar);
    wind_close(' ',42,10,52,10);
end;
begin
    open1(37,38,57,58,7,8,11,12);
    mark(htop);
    gotoxy(43,8);write(' ชื่อตัวแปร ');

```

```

open1(40,41,53,54,9,10,10,11);
Hvar := nil;
repeat
  getlink;
until Mvar^.Vname = #13;
if hvar ^.vname <> #13 then
  begin
    if j = 1 then
      stmean1 := 'mean '
    else stmean1 := '';
    repeat
      stmean1 := stmean1 + hvar^.vname + ' ';
      hvar := hvar^.vpnt;
    until hvar^.vpnt = nil;
  end;
release(htop);
wind_close(' ',37,7,58,12);
end;

procedure MeanGr(var stmean2 : string);
type Mlink = ^Mlist;
      Mlist = record Mname : string[10];
                  Mpnt : Mlink;
            end;
var Grvar,GrHead,LastGr : Mlink;
    htop : ^integer;
    Group,hvar : vlink;

procedure LinkSubGr;
var v : string;
begin
  new(Grvar);
  gotoxy(53,11);readchar(v,10);

```

```

Grvar^.Mname := v;
if GrHead = nil then
  GrHead := Grvar
else
  LastGr^.Mpnt := Grvar;
LastGr      := Grvar;
LastGr^.Mpnt := nil;
wind_close(' ',53,11,71,11);
end;

procedure LinkGr;
var v : string;
begin
  new(Group);
  gotoxy(53,11);readchar(v,10);
  Group^.Vname := v;
  pnter(hvar,Group);
  wind_close(' ',53,11,71,11);
end;

procedure SubGrvar(alp : string;var stmean2 : string);
begin
  wind_close(' ',38,10,77,13);
  open1(51,52,71,72,10,11,11,12);
  gotoxy(44,09);write(' ตัวแปรแบ่งกลุ่มย่อย ');
  gotoxy(38,11); write(' ชื่อตัวแปร ');
  GrHead := nil;
  repeat
    LinkSubGr;
  until Grvar^.Mname = #13;
  if GrHead^.Mname <> #13 then
    begin
      stmean2 := stmean2 +'by ';
    end;
  end;

```

```

repeat
    stmean2 := stmean2 + Grhead^.Mname + ' ';
    GrHead := GrHead^.Mpnt;
until GrHead^.Mpnt = nil;
end;
gotoxy(38,13); write('ต้องการตัวแปรแบ่งกลุ่มย่อยหรือไม่[Y/N] ');
readchar(alp,1);
if alp[1] in['Y','y'] then
    SubGrvar(alp,stmean2);
end;
procedure listGrvar;
var alp : string;
begin
    open1(51,52,71,72,10,11,11,12);
    gotoxy(44,09);write('   ตัวแปรแบ่งกลุ่ม   ');
    gotoxy(38,11); write(' ชื่อตัวแปร ');
    Hvar := nil;
    repeat
        LinkGr;
    until Group^.Vname = #13;
    if hvar^.vname <> #13 then
        begin
            stmean2 := 'by ';
            repeat
                stmean2 := stmean2 + hvar^.vname + ' ';
                hvar := hvar^.vpnt;
            until hvar^.vpnt = nil;
            gotoxy(38,13); write('ต้องการตัวแปรแบ่งกลุ่มย่อยหรือไม่(Y/N) ');
            readchar(alp,1);
            if alp[1] in['Y','y'] then
                SubGrvar(alp,stmean2);
        end;
    end;
end;

```

```

        end;
end;
begin
    open1(36,37,77,78,09,10,13,14);
    mark(htop);
    listgrvar;
    release(htop);
    wind_close(' ',36,09,78,14);
end;
procedure Onewayvar(var st : string);
var htop : ^integer;
    Ovar,hvar : vlink;
procedure getlink;
var v : string;
begin
    new(Ovar);
    gotoxy(43,10);readchar(v,10);
    Ovar^.vname := v;
    pnter(hvar,Ovar);
    wind_close(' ',43,10,53,10);
end;
begin
    open1(36,37,63,64,8,9,11,12);
    mark(htop);
    open1(40,41,59,60,9,10,10,11);
    gotoxy(44,8);write('ชื่อตัวแปร ');
    Hvar := nil;
    repeat
        getlink;
    until Ovar^.Vname = #13;
    if hvar^.vname <> #13 then

```

```

begin
  st := 'oneway';
  repeat
    st := st+ hvar^.vname + ' ';
    hvar:= hvar^.vpnt;
  until hvar^.vpnt = nil;
end;
release(htop);
wind_close(' ',36,8,64,12);
end;
procedure OnewayGr(var st : string);
var htop : ^integer;
    minst,maxst : string;
    va: string;
procedure LinkGrOne;
var MaxOne,MinOne : real;
begin
  MinOne := 0.0;
  MaxOne := 0.0;
  gotoxy(42,11);write(' ');readchar(va,10);
  if va <> #13 then
    begin
      gotoxy(48,13);write(' ');readreal(MinOne,7,code);
      gotoxy(68,13);write(' ');readreal(MaxOne,7,code);
      str(MinOne:7:2,Minst);
      str(MaxOne:7:2,Maxst);
      st := 'by ' + va+ '( ' + minst + ', ' + maxst + ' )';
    end;
end;
end;
begin
  open1(36,37,77,78,8,9,13,14);

```



```

open1(40,41,60,61,10,11,11,12);
gotoxy(44,9);write('   ชื่อตัวแปร ');
gotoxy(37,13);write('ค่าต่ำสุด [           ]');
gotoxy(57,13);write('ค่าสูงสุด [           ]');
linkGrOne;
wind_close(' ',36,8,78,14);

end;

procedure OnewayVec(var st : string);
type conlnk = ^conlist;
      conlist = record cont : real;
                  next : conlnk;
                end;

var vec : real;
    vecst : string;
    head,last,cotra : conlnk;
    htop : ^integer;

procedure getlink;
begin
    new(cotra);
    gotoxy(63,11);readreal(vec,10,code);
    cotra^.cont := vec;
    if Head = nil then
        Head := Cotra
    else
        Last^.next := Cotra;
    Last      := Cotra;
    Last^.next := nil;
    wind_close(' ',63,11,68,11);

end;

begin
    open1(36,37,75,76,9,10,11,12);

```

```

gotoxy(38,10);write('เปรียบเทียบและทดสอบ ');
gotoxy(38,11);write('ค่าเฉลี่ยในรูปเวกเตอร์ [      ]');
mark(htop);
head := nil;
repeat
  getlink;
until cotra^.cont = 0.0;
if head^.cont <> 0.0 then
  st := '/con = ';
  repeat
    str(head^.cont:4:2,vecst);
    st := st + vecst + ' ';
    head := head^.next;
  until head^.next = nil;
release(htop);
wind_close(' ',36,9,76,12);
end;
procedure plot_var(var st : string);
var pvar,hvar,fvar,head,gvar,first : vlink;
    htop      : ^integer;
    ch        : string;
procedure getlink;
var v,v1,v2 : string;
begin
  new(pvar);
  gotoxy(52,10);readchar(v,10);
  pvar^.vname := v;
  pnter(hvar,pvar);
  wind_close(' ',52,10,66,10);
  if (v = #13) and (hvar^.vname <> #13) then
    begin

```

```

repeat
  new(Fvar);
  gotoxy(52,13);readchar(v1,10);
  Fvar^.vname := v1;
  pnter(head,Fvar);
  wind_close(' ',52,13,66,13);
until Fvar^.vname = #13 ;
gotoxy(38,16);write('ต้องการตัวแปรควบคุมหรือไม่[Y/N] [ ]');
gotoxy(71,16);readchar(ch,1);
if ch[1] in['y','Y'] then
  begin
    wind_close(' ',38,16,72,16);
    gotoxy(38,16);write('ชื่อตัวแปรควบคุม');
    open1(54,55,67,68,15,16,16,17);
    repeat
      new(gvar);
      gotoxy(56,16);readchar(v2,10);
      gvar^.vname := v2;
      pnter(first,gvar);
      wind_close(' ',56,16,66,16);
    until gvar^.vname = #13 ;
  end
else first^.vname := #13;
end;
end;
begin
  open1(36,37,72,73,8,9,17,18);
  mark(htop);
  open1(50,51,67,68,9,10,10,11);
  open1(50,51,67,68,12,13,13,14);
  gotoxy(38,10);write('ชื่อตัวแปร ');
  gotoxy(38,13);write('กับตัวแปร ');

```

```

Hvar := nil;
head := nil; first := nil;
repeat
  getlink;
until pvar^.Vname = #13;
if hvar^.vname <> #13 then
  begin
    st := 'plot ';
    repeat
      st := st + hvar^.vname + ' ';
      hvar:= hvar^.vpnt;
    until hvar^.vpnt = nil;
    st := st + 'with ';
    repeat
      st := st + head^.vname + ' ';
      head := head^.vpnt;
    until head^.vpnt = nil;
    if first^.vname <> #13 then
      begin st := st + 'by ';
        repeat st := st + first^.vname + ' ';
          first := first^.vpnt;
        until first^.vpnt = nil;
      end;
    end;
  release(htop);
  wind_close(' ',36,8,73,18);
end;
procedure Regression_var(var streg : string);
var htop : ^integer;
    head,Revar,Redep,hvar : vlink;
procedure getDep;

```

```

var   Depen : string;
begin
    new(ReDep);
    gotoxy(51,12);write(' ');readchar(Depen,8);
    ReDep^.vname := Depen;
    pnter(head,reddep);
    wind_close(' ',51,12,62,12);
end;

procedure  getRevar;
var   Reqr : string;
begin
    new(Revar);
    gotoxy(51,9);write(' ');readchar(Reqr,8);
    Revar^.vname := Reqr;
    pnter(hvar,Revar);
    wind_close(' ',51,9,62,9);
end;

procedure  listvar(var streg : string);
begin
    gotoxy(38,9);write('ชื่อตัวแปร ');open1(50,51,63,64,8,9,9,10);
    gotoxy(38,12);write('ตัวแปรตาม');open1(50,51,63,64,11,12,12,13);
    Hvar := nil;
    repeat
        getRevar;
    until Revar^.vname = #13;
    if Hvar^.vname <> #13 then
    begin  Head := nil;
        repeat
            getDep;
        until Redep^.vname = #13;
        streg := 'regr var = ';
    end;
end;

```

```

    repeat
        streg := streg + hvar^.vname + ' ';
        hvar := hvar^.vpnt;
    until hvar^.vpnt = nil;
    if head^.vname <> #13 then
    begin streg := streg+'depen ';
        repeat
            streg := streg + head^.vname+' ';
            head := head^.vpnt;
        until head^.vpnt = nil;
    end;
end;
end;
begin
    open1(36,37,70,71,7,8,13,14);
    gotoxy(45,7);write(' ตัวแปร ');
    mark(htop);
    listvar(streg);
    release(htop);
    wind_close(' ',36,7,71,14);
end;
procedure Tind_va(var str1 : string);
var top : ^integer;
    Tva,hvar : vlink;
procedure getlink;
var v : string;
begin
    new(Tva);
    gotoxy(40,15);readchar(v,10);
    Tva^.vname := v;
    pnter(hvar,tva);

```

```

    wind_close(' ',40,15,61,15);
end;
begin
    open1(36,37,64,65,13,14,16,17);
    gotoxy(39,13);write(' ชื่อตัวแปร ');
    open1(38,39,62,63,14,15,15,16);
    mark(top);
    Hvar := nil;
    repeat
        getlink;
    until Tva^.Vname = #13;
    if Hvar^.vname <> #13 then
        str1 := '/vari  ';
        repeat
            str1 := str1+hvar^.vname+' ';
            hvar := hvar^.vpnt;
        until hvar^.vpnt = nil;
    release(top);
    wind_close(' ',36,13,65,17);
end;
procedure T2gr(var str2 : string);
procedure getlink;
var  v,grval1,grval2 : string;
     valu1 ,valu2  : integer;
begin
    gotoxy(59,15);readchar(v,10);
    if v <> #13 then
        begin
            gotoxy(59,18);readint(valu1,5,code);
            if valu1 <> 0 then
                begin

```

```

        gotoxy(59,19);readint(valu2,5,code);
    end;
end
else begin valu1 := 0 ; valu2 := 0; end;
str(valu1,grval1);
str(valu2,grval2);
str2 := 't-test gro ';
if valu1 <> 0 then
    str2 := str2+v+ ('+grval1+', '+grval2+') '
else
    str2 := str2+v;
end;
begin
    open1(36,37,71,72,13,14,20,21);
    gotoxy(38,15);write('ชื่อตัวแปร ');
    open1(57,58,69,70,14,15,15,16);
    gotoxy(38,18);write('กลุ่มที่ 1 มีค่า');
    gotoxy(38,19);write('กลุ่มที่ 2 มีค่า');
    open1(57,58,69,70,17,18,19,20);
    getlink;
    wind_close(' ',36,13,72,21);
end;
procedure Tdep_va(var stdep : string);
var top : ^integer;
    Tva1,hvar : vlink;
procedure getlink;
var v : string;
begin
    new(Tva1);
    gotoxy(54,13);readchar(v,10);
    Tva1^.vname := v;

```



```

    pnter(hvar,tva1);
    wind_close(' ',54,13,65,13);
end;
begin
    open1(36,37,72,73,11,12,14,15);
    gotoxy(38,11);write('กลุ่มตัวอย่างไม่เป็นอิสระต่อกัน');
    gotoxy(38,13);write(' ชื่อตัวแปร');
    open1(52,53,66,67,12,13,13,14);
    mark(top);
    Hvar := nil;
    repeat
        getlink;
    until Tva1^.Vname = #13;
    if hvar^.vname <> #13 then
        begin stdep := 't-test pair ';
            repeat
                stdep := stdep + hvar^.vname + ' ';
                hvar := hvar^.vpnt;
            until hvar^.vpnt = nil;
        end;
    release(top);
    wind_close(' ',36,11,73,15);
end;

procedure Npar_with(var stpar : string);
var head,hvar,nva1,nva2 : vlink;
    htop : ^integer;
procedure getlink(var nva,first : vlink;y:byte);
var v : string;
begin
    repeat
        new(nva);

```

```

    gotoxy(52,y);readchar(v,10);
    nva^.vname := v;
    pnter(first,nva);
    wind_close(' ',52,y,65,y);
until nva^.vname = #13
end;
begin
    open1(36,37,71,72,8,9,14,15);
    open1(50,51,67,68,9,10,10,11);
    open1(50,51,67,68,12,13,13,14);
    gotoxy(38,10);write('ชื่อตัวแปร ');
    gotoxy(38,13);write('กับตัวแปร ');
    mark(htop);
    head := nil; hvar := nil;
    getlink(nva1,head,10);
    getlink(nva2,hvar,13);
    if head^.vname <> #13 then
        begin repeat
            stpar := stpar + head^.vname + ' ';
            head := head^.vpnt;
        until head^.vpnt = nil;
        end;
    if hvar^.vname <> #13 then
        begin stpar := stpar + 'with ';
            repeat
                stpar := stpar + hvar^.vname + ' ';
                hvar := hvar^.vpnt;
            until hvar^.vpnt = nil;
        end;
    release(htop);
    wind_close(' ',36,8,72,15);end;

```

```
procedure Npva(var stn : string);
var  htop    : ^integer;
     Pvar,hvar : vlink;
procedure getlink;
var  va : string;
begin
    new(Pvar);
    gotoxy(40,11);readchar(va,10);
    pvar^.vname := va;
    pnter(hvar,pvar);
    wind_close(' ',40,11,55,11);
end;
begin
    mark(htop);
    open1(36,37,60,61,8,9,12,13);
    gotoxy(40,9);writeln('ชื่อตัวแปร');
    Hvar := nil;
    open1(38,39,58,59,10,11,11,12);
    repeat
        getlink;
    until pvar^.vname = #13;
    if Hvar^.vname <> #13 then
        begin
            repeat
                stn := stn + Hvar^.vname + ' ';
                Hvar := Hvar^.vpnt;
            until Hvar^.vpnt = nil;
        end;
    release(htop);
    wind_close(' ',36,8,61,13);
end;
```

```

procedure nby(var stn : string);
var htop : ^integer;
    nva,head : vlink;
    va,va1,va2 : string;
    val1,val2 : real;
procedure getlink(var nva : vlink);
var v : string;
begin
    repeat
        new(nva);
        gotoxy(52,10);readchar(v,10);
        nva^.vname := v;
        pnter(head,nva);
        wind_close(' ',52,10,61,10);
    until nva^.vname = #13
end;
begin
    open1(36,37,68,69,8,9,20,21);
    open1(50,51,62,63,9,10,10,11);
    open1(50,51,62,63,12,13,13,14);
    open1(50,51,62,63,15,16,16,17);
    open1(50,51,62,63,18,19,19,20);
    gotoxy(38,10);write('ชื่อตัวแปร ');
    gotoxy(38,13);write('กับตัวแปร ');
    gotoxy(38,16);write('VALUE1 ');
    gotoxy(38,19);write('VALUE2 ');
    mark(htop);
    head := nil;
    getlink(nva);
    repeat
        gotoxy(52,13);readchar(va,10);

```

```

until va <> #13;
gotoxy(52,16);readreal(val1,8,code);
gotoxy(52,19);readreal(val2,8,code);
str(val1:8:2,va1);str(val2:8:2,va2);
if head^.vname <> #13 then begin repeat
    stn := stn + head^.vname + ' ';
    head := head^.vpnt;until head^.vpnt = nil;
    stn := stn + 'by ' + va +'(' + va1 + ',' + va2 + ')';end;
release(htop);wind_close(' ',36,8,72,21);end;
procedure Remo_var(var stn : string);var htop : ^integer;
    Pvar,hvar : vlink;
procedure getlink;
var va : string;
begin
    new(Pvar);gotoxy(40,18);readchar(va,10);
    pvar^.vname := va;pnter(hvar,pvar);
    wind_close(' ',40,18,55,18);end;
begin
    mark(htop);open1(36,37,60,61,15,16,19,20);
    gotoxy(40,16);writeln('ข้อความปร');Hvar := nil;
    open1(38,39,58,59,17,18,18,19);
    repeat getlink;
until pvar^.vname = #13;
if Hvar^.vname <> #13 then
    begin repeat
        stn := stn + Hvar^.vname + ' ';
        Hvar := Hvar^.vpnt;
    until Hvar^.vpnt = nil;end;release(htop);
    wind_close(' ',36,15,61,20);end;
begin
end.

```

โปรแกรม UTI.PAS

```

unit Uti;
interface
uses crt, win, keybord, windo, fi;

type  KeepPtr = ^NameRec;
      NameRec = record
                name : string[10];
                next : KeepPtr;
            end;

var    j : byte;
        LastName, FirstName, wr, disp, Uv, Wv : KeepPtr;

procedure  Uvar(var stlis : string);
procedure  Ulocate(var stlis : string);
procedure  Psave;
procedure  Export_name(var stex : string);
procedure  Export_No(var stex : string);
procedure  Export_Ok(var stex : string);
procedure  Exp_Change(var stex : string);
{procedure  Exp_digit(var stex : string);}
procedure  Sort_case;

implementation

procedure  enterst(n : keepPtr);
begin
    if FirstName = nil then
        FirstName := n
    else
        LastName^.next := n;

```

```

    LastName := n;
    LastName^.next := nil;end;
procedure Uvar(var stlis : string);
var names : string;
    top : ^integer;
begin
    open1(17,18,64,65,12,13,15,16);
    gotoxy(30,12);writeln(' ตัวแปร ');
    gotoxy(19,14);writeln(' รายชื่อตัวแปร ');open1(36,37,61,62,13,14,14,15);
    mark(top);
    FirstName := nil;
    repeat
        begin new(Uv);
            gotoxy(38,14);readchar(names,10);
            Uv^.name := names;
            enterst(Uv);
            wind_close(' ',38,14,59,14);
        end;
    until Uv^.name = #13;
    if FirstName^.name <> #13 then
        begin
            stlis := 'list var = ';
            repeat
                stlis := stlis + FirstName^.name + ' ';
                FirstName := FirstName^.next;
            until FirstName^.next = nil;
        end
    else
        stlis := 'list';
    release(top);
    wind_close(' ',17,12,65,16);

```

```

end;
procedure Ulocate(var stlis : string);
var  top      : ^integer;
     FstLoc,Add  : Integer;
     LstLoc,stAdd,Loc : String;

begin
  open1(17,18,64,65,11,12,20,21);
  gotoxy(30,11);writeln('  ขอบเขตข้อมูล ');
  gotoxy(19,13);writeln('ลำดับเริ่มต้นของชุดข้อมูล');
  open1(44,45,62,63,12,13,13,14);
  gotoxy(19,16);writeln('ลำดับสุดท้ายของชุดข้อมูล');
  open1(44,45,62,63,15,16,16,17);
  gotoxy(19,19);writeln('เพิ่มชุดข้อมูลครั้งละ');
  open1(44,45,62,63,18,19,19,20);
  gotoxy(46,13);readint(FstLoc,6,code);
  gotoxy(46,16);readchar1(LstLoc,3);
  gotoxy(46,19);readInt(Add,6,code);
  str(fstloc,loc);
  str(Add,Stadd);
  stlis := '/case = from ' + Loc + ' to ' + LstLoc + ' by ' + StAdd;
  wind_close(' ',17,11,65,21);

end;
procedure Psave;
var  fname ,st      : string;
     check          : string;
     top            : ^integer;
     f              : text;

procedure Getlink;
var  nkeep : string;

```



```
begin
```

```
  new(Uv);
  gotoxy(52,14);readchar(nkeep,10);
  Uv^.name := nkeep ;
  enterst(Uv);
  wind_close(' ',52,14,64,14);end;
```

```
begin
```

```
  assign(f,'c:\sptool\ex.sps');
  open1(17,18,67,68,08,09,15,16);
  gotoxy(25,08);writeln(' เก็บข้อมูลและผลลัพธ์ลงแฟ้มข้อมูล ');
  gotoxy(19,10);writeln(' ชื่อแฟ้มข้อมูลใหม่ ');
  open1(40,41,63,64,09,10,10,11);
  gotoxy(42,10);readchar(fname,20);
  if fname <> #13 then
    begin
      st := 'save outfile = "' + fname + "' ';
      gotoxy(19,12);write(' เก็บชื่อตัวแปรทั้งหมด [Y/N] [ ] ');
      gotoxy(49,12);readchar(check,1);
      mark(top);
      FirstName := nil;
      if check[1] in['N','n'] then
        begin gotoxy(19,14);write(' ชื่อตัวแปรที่ไม่ต้องการเก็บ ');
          open1(50,51,65,66,13,14,14,15);
          repeat getlink until Uv^.name = #13;
          if FirstName^.name <> #13 then
            begin
              st := st + '/drop = ';
              repeat
                st := st + FirstName^.name + ' ';
                FirstName := FirstName^.next ;
              until FirstName^.next = nil;
```

```

                end;
            end;
            st := st + '.';
            makefi(st,f);
            release(top);
        end;
        wind_close(' ',17,08,68,16);
end;
procedure Export_name(var stex : string);
var Exp_name : string;
    top : ^integer;
begin
    open1(17,18,65,66,11,12,14,15);
    gotoxy(19,13);writeln('ชื่อเพิ่มข้อมูลใหม่');
    open1(40,41,63,64,12,13,13,14);
    gotoxy(42,13);readchar(Exp_Name,20);
    if Exp_Name <> #13 then
        stex := Exp_Name + ' ';
        wind_close(' ',17,11,66,15);
    end;
end;
procedure Export_No(var stex : string);
var names : string;
    top : ^integer;
begin
    stex := '';
    open1(17,18,72,73,11,12,14,15);
    gotoxy(19,13);writeln('รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่');
    open1(58,59,70,71,12,13,13,14);
    FirstName := nil;
    mark(top);
    repeat

```

```

    new(Wv);
    gotoxy(60,13);readchar(Names,10);
    Wv^.name := Names;
    enterst(Wv);
    wind_close(' ',60,13,70,13);
until Wv^.name = #13;
if FirstName^.Name <> #13 then
    repeat
        stex := stex + FirstName^.name + ' ';
        FirstName := FirstName^.next;
    until FirstName^.next = nil;
release(top);
wind_close(' ',17,11,73,15);
end;
procedure Export_Ok(var stex : string);
var names : string;
    htop : ^integer;
begin
    stex := '';
    open1(17,18,72,73,11,12,14,15);
    gotoxy(19,13);writeln('รายชื่อตัวแปรที่เก็บในแฟ้มข้อมูลใหม่');
    open1(56,57,69,70,12,13,13,14);
    FirstName := nil;mark(htop);
    repeat
        new(dispatch);
        gotoxy(58,13);readchar(Names,10);
        Dispatch^.name := names;
        enterst(Dispatch);
        wind_close(' ',58,13,69,13);
    until dispatch^.name = #13;
    if FirstName^.Name <> #13 then

```

```

repeat
    stex := stex + FirstName^.name + ' ';
    FirstName := FirstName^.next;
until FirstName^.next = nil;
release(htop);
wind_close(' ',17,11,73,15);
end;
procedure Exp_change(var stex : string);
type Explis = ^Explink;
    Explink = record
        Expold : string[10];
        Expnew : string[10];
        next : Explis;
    end;
var Nold,Nnew,st : string;
    top : ^integer;
    first,last,Expo : Explis;
begin
    stex := ''; st := '= ';
    open1(17,18,70,71,11,12,17,18);
    gotoxy(23,11);writeln(' เปลี่ยนรายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่');
    gotoxy(19,13);writeln(' รายชื่อตัวแปรใหม่เพิ่มข้อมูลเก่า');
    open1(57,58,66,67,12,13,13,14);
    gotoxy(19,16);writeln(' รายชื่อตัวแปรใหม่เพิ่มข้อมูลใหม่');
    open1(57,58,67,68,15,16,16,17);
    first := nil;mark(top);
    repeat
        new(Expo);
        gotoxy(59,13);readchar(Nold,10);
        Expo^.Expold := Nold;
        if Nold <> #13 then

```

```

begin
    gotoxy(59,16);readchar(Nnew,10);end
else Nnew := '';
Expo^.Expnew := Nnew;
if first = nil then
    first := Expo
else
    Last^.next := Expo;
Last := Expo;
Last^.next := nil;
wind_close(' ',59,13,66,13);
wind_close(' ',59,16,66,16);
until Expo^.Expold = #13;
if first^.Expold <> #13 then
begin
repeat
    stex := stex + first^.expold + ' ';
    st := st + first^.expnew + ' ';
    first := first^.next;
until first^.next = nil;
stex := stex + st + ')';
end;
release(top);
wind_close(' ',17,11,71,18);
end;{
procedure Exp_digit(var stex : string);
var num : integer;
    stnum : string;
begin
open1(20,21,54,55,11,12,14,15);
gotoxy(23,13);writeln(' จำนวนทศนิยม ');

```

```

open1(42,43,49,50,12,13,13,14);
gotoxy(44,13);readint(num,8,code);
str(num,stnum);
if stnum <> '' then
    stex := ' /digit = '+ stnum;
wind_close(' ',17,11,55,15)
end;

procedure Sort_case;
type Sortlnk = ^Sortlist;
    Sortlist = record Sname    : string[10];
                    Smethod   : string[11];
                    next      : Sortlnk;
                end;

var head,last,so : Sortlnk;
    htop    : ^integer;
    st      : string;
    f       : text;

procedure getlink;
var vname,mth : string;
begin
    new(so);
    gotoxy(42,10);readchar(vName,10);
    if vname <> #13 then
        begin gotoxy(42,13); readchar(mth,1); end
    else mth := '';
    with so^ do
        begin Sname := vname;
            Smethod := mth;
        end;
    if head = nil then
        head := so

```

```

else
    Last^.next := so;
Last := so;
Last^.next := nil;
wind_close(' ',42,10,58,10); wind_close(' ',42,13,58,13);
end;
begin
    assign(f,'c:\sptool\ex.sps');
    st := '';
    open1(17,18,64,65,08,09,16,17);
    gotoxy(30,08);write(' การเรียงลำดับข้อมูล ');
    gotoxy(19,10);write(' ชื่อตัวแปร ');
    open1(40,41,59,60,09,10,10,11);
    gotoxy(19,13);write(' วิธีเรียงลำดับข้อมูล * ');
    open1(40,41,59,60,12,13,13,14);
    gotoxy(19,15);write('* A เรียงจากน้อยไปมาก ');
    gotoxy(19,16);write(' D เรียงจากมากไปน้อย ');
    head := nil;mark(htop);
    repeat getlink until so^.sname = #13;
    if head^.sname <> #13 then
        begin
            st := 'sort case by ';
            repeat
                st := st + head^.sname + '(';
                if head^.smthod <> #13 then
                    st := st + head^.smthod + ')'
                else st := '';
                head := head^.next ;
            until head^.next = nil;
            if st <> '' then
                begin st := st + '.';

```

```
        makefi(st,f); end;
```

```
    end;
```

```
    release(htop);
```

```
    wind_close(' ',17,08,65,17);
```

```
end;
```

```
begin {main}
```

```
end.
```


โปรแกรม SWINO.PAS

```

unit swin0;
interface
uses  Crt,keyboard,Win,SSub,Windo,Trans,proc,uti,display,fi;

const  MaxChoi  = 14;
type   String80  = string[80];
      StructurSmenu= record
                WinS   : array[1..4] of byte;
                ColS   : array[1..MaxChoi] of byte;
                RowS   : array[1..MaxChoi] of byte;
                MsgS   : array[1..MaxChoi] of String80;
                LastChoi  : byte;
            end;
      Strucsub  = array[1..1] of StructurSmenu;
Const  Selec   : Strucsub = (
                (WinS : (22,05,62,10);
                 ColS : (02,02,02,02,00,00,00,00,00,00,00,00,00,00);
                 RowS : (01,02,03,04,00,00,00,00,00,00,00,00,00,00);
                 MsgS : (' จำนวนชุดข้อมูล',
                        ' โดยวิธีการสุ่มตัวอย่าง',
                        ' โดยกำหนดเงื่อนไขทางตรรก',
                        ' โดยกำหนดเงื่อนไขทางตรรก(Procedure)',
                        '','','','','','','','','');
                 LastChoi:4));
      Transf   : Strucsub = (
                (WinS : (22,05,62,11);
                 ColS : (02,02,02,02,02,00,00,00,00,00,00,00,00,00);
                 RowS : (01,02,03,04,05,00,00,00,00,00,00,00,00,00);
                 MsgS : (' เปลี่ยนค่าของตัวแปร',

```

```

' สร้างตัวแปรใหม่โดยการคำนวณ      ',
' สร้างตัวแปรใหม่โดยการกำหนดเงื่อนไข  ',
' สร้างตัวแปรใหม่โดยการนับ          ',
' กำหนดน้ำหนักของตัวแปร            ',
'', '', '', '', '', '', '', '', '', '');

```

```
LastChoi:5));
```

```
Sdef : StrucSub = (
(WinS : (22,04,61,10);
```

```
ColS : (02,02,02,02,02,00,00,00,00,00,00,00,00,00);
```

```
RowS : (01,02,03,04,05,00,00,00,00,00,00,00,00,00);
```

```
MsgS : (
```

```

' เรียกเพิ่มข้อมูล      ',
' บอกรายละเอียดเกี่ยวกับตัวแปร      ',
' บอกค่า ไม่สมบูรณ์ให้กับตัวแปร      ',
' บอกรายละเอียดเกี่ยวกับค่าตัวแปร      ',
' บอกรายละเอียดเกี่ยวกับชื่อตัวแปร      ',
'', '', '', '', '', '', '', '', '', '');

```

```
LastChoi:5));
```

```
Lis : Strucsub = (
```

```
(WinS : (57,05,77,8);
```

```
ColS : (02,02,00,00,00,00,00,00,00,00,00,00,00,00);
```

```
RowS : (01,02,00,00,00,00,00,00,00,00,00,00,00,00);
```

```
MsgS : (' ตัวแปร      ',
```

```
' ขอบเขตของข้อมูล      ',
```

```
{ ' รูปแบบข้อมูล      ', }'',
```

```
'', '', '', '', '', '', '', '', '', '');
```

```
LastChoi:2));
```

```
Export : Strucsub = (
```

```
(WinS : (15,08,59,13);
```

```
ColS : (02,02,02,02,02,00,00,00,00,00,00,00,00,00);
```

```
RowS : (01,02,03,04,05,00,00,00,00,00,00,00,00,00);
```

```

MsgS : (' ชื่อเพิ่มข้อมูลใหม่
        ' รายชื่อตัวแปรที่ไม่เก็บในเพิ่มข้อมูลใหม่ ',
        ' รายชื่อตัวแปรที่เก็บในเพิ่มข้อมูลใหม่ ',
        ' เปลี่ยนรายชื่อตัวแปรในเพิ่มข้อมูลใหม่ ',
        '', '', '', '', '', '', '', '', '', '');
LastChoi:4));

```

```

SubFre : Strucsub = (
(WinS : (34,04,59,12);
ColS : (02,02,02,02,02,02,02,00,00,00,00,00,00,00);
RowS : (01,02,03,04,05,06,07,00,00,00,00,00,00,00);
MsgS : (' ตัวแปร
        ' รวมค่าไม่สมบูรณ์ ',
        ' กราฟแท่ง ',
        ' ฮิสโตแกรม ',
        ' เปอร์เซนต์ไทล์ ',
        ' เอนไทล์ ',
        ' ค่าสถิติ ',
        '', '', '', '', '', '', '', '');
LastChoi:7));

```

```

SubDes : Strucsub = (
(WinS : (30,05,50,09);
ColS : (02,02,02,00,00,00,00,00,00,00,00,00,00,00,00);
RowS : (01,02,03,00,00,00,00,00,00,00,00,00,00,00,00);
MsgS : (' ตัวแปร
        ' รูปแบบการคำนวณ ',
        ' ค่าสถิติ ',
        '',
        '', '', '', '', '', '', '', '', '');
LastChoi:3));

```

```

SubMean : Strucsub = (
(WinS : (34,06,71,11);

```

```
ColS : (02,02,02,02,00,00,00,00,00,00,00,00,00,00);
```

```
RowS : (01,02,03,04,00,00,00,00,00,00,00,00,00,00);
```

```
MsgS : (
```

```
    ' ตัวแปรที่คำนวณค่าสถิติเบื้องต้น ',
```

```
    ' ตัวแปรแบ่งกลุ่ม ',
```

```
    ' รูปแบบ ',
```

```
    ' ค่าสถิติ ',
```

```
    ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ');
```

```
LastChoi:4));
```

```
SubCross : Strucsub = (
```

```
(WinS : (34,07,50,11);
```

```
ColS : (02,02,02,00,00,00,00,00,00,00,00,00,00,00);
```

```
RowS : (01,02,03,00,00,00,00,00,00,00,00,00,00,00);
```

```
MsgS : (' ตัวแปร ',
```

```
        ' รูปแบบ ',
```

```
        ' ค่าสถิติ ',
```

```
    ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ');
```

```
LastChoi:3));
```

```
SubT_test: Strucsub = (
```

```
(WinS : (34,08,72,12);
```

```
ColS : (02,02,02,00,00,00,00,00,00,00,00,00,00,00);
```

```
RowS : (01,02,03,00,00,00,00,00,00,00,00,00,00,00);
```

```
MsgS : (
```

```
    ' กลุ่มตัวอย่างเป็นอิสระต่อกัน ',
```

```
    ' กลุ่มตัวอย่างไม่เป็นอิสระต่อกัน ',
```

```
    ' รูปแบบผลลัพธ์ ',
```

```
    ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ');
```

```
LastChoi:3));
```

```
SubOneway : Strucsub = (
```

```
(WinS : (29,09,75,17);
```

```
ColS : (02,02,02,02,02,02,02,00,00,00,00,00,00,00);
```

RowS : (01,02,03,04,05,06,07,00,00,00,00,00,00,00);

MsgS :(

' ตัวแปรตาม ' ,
 ' ตัวแปรแบ่งกลุ่ม ' ,
 ' กำหนดการแบ่งกลุ่มของตัวแปรต้นแปร ' ,
 ' เปรียบเทียบ, ทดสอบค่าเฉลี่ยในรูปเวกเตอร์ ' ,
 ' กำหนดตัวสถิติเพื่อทดสอบค่าเฉลี่ย ' ,
 ' รูปแบบ ' ,
 ' ค่าสถิติ ' ,
 ' ',' ',' ',' ',' ',' ',' ');

LastChoi:7));

Sanova : Strucsub = (

(WinS : (34,09,73,15);

ColS : (02,02,02,02,02,00,00,00,00,00,00,00,00,00);

RowS : (01,02,03,04,05,00,00,00,00,00,00,00,00,00);

MsgS : (' ตัวแปรที่วิเคราะห์ความแปรปรวน ' ,
 ' ตัวแปรที่ให้เป็นตัวแปรแบ่งกลุ่ม ' ,
 ' ชื่อตัวแปรที่เป็นตัวแปรร่วม ' ,
 ' รูปแบบการวิเคราะห์ความแปรปรวน ' ,
 ' ค่าสถิติ ' ,
 ' ',' ',' ',' ',' ',' ',' ',' ');

LastChoi:5));

SubCorr : Strucsub = (

(WinS : (30,10,55,14);

ColS : (02,02,02,00,00,00,00,00,00,00,00,00,00,00);

RowS : (01,02,03,00,00,00,00,00,00,00,00,00,00,00);

MsgS : (' ตัวแปร ' ,
 ' รายละเอียดเพิ่มเติม ' ,
 ' ค่าสถิติ ' ,
 ' ',' ',' ',' ',' ',' ',' ',' ');

LastChoi:3));

```

SubRegr : StrucSub = (
    (WinS : (34, 11, 61, 14));
    ColS : (02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);
    RowS : (01, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);
    MsgS : (
        ' ตัวแปร ',
        ' วิธีการคัดเลือกตัวแปร ',
        '', '', '', '', '', '', '', '', '', '', '', '');
    LastChoi:2));

```

```

SubPlot : Strucsub = (
    (WinS : (34, 9, 69, 14));
    ColS : (02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);
    RowS : (01, 02, 03, 04, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);
    MsgS : (
        ' กำหนดข้อความและสเกลในแนวนอน ',
        ' กำหนดข้อความและสเกลในแนวตั้ง ',
        ' กำหนดชื่อของกราฟ ',
        ' เขียนกราฟ ',
        '', '', '', '', '', '', '', '', '', '');
    LastChoi:4));

```

```

SubNpar : Strucsub = (
    (WinS : (34, 5, 61, 20));
    ColS : (02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02);
    RowS : (01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14);
    MsgS : (
        ' ตัวแปรที่ต้องการทดสอบ ',
        ' Binomial ',
        ' Mcnemar ',
        ' Wilcoxon ',
        ' Cochran Q tests ',
        ' Frideman ');

```

```

' Kendall      ',
' Median      ',
' Mann - Whitney U ',
' Kolmogorov - Smirnov ',
' Wald - Wolfowitz ',
' Kruskal - Wallis  ',
' รูปแบบ      ',
' ค่าสถิติ      ');

```

```
LastChoi:14));
```

```
procedure Initsub (SubM:StrucSub;StatusMenu ,CurChoice:byte);
```

```
procedure DisplayFuncKey;
```

```
implementation
```

```
Const ChoiSmenu : array[1..1] of 1..MaxChoi = (1);
```

```
StatusChoi : 1..MaxChoi = 1;
```

```
var KeyS : char;
```

```
SubMenu : strucSub;
```

```
SSMenu : StrucSsub;
```

```
Succ : Boolean;
```

```
Current, ShowStatus, CurSub, j : byte;
```

```
s,s1 : string;
```

```
procedure DisplayFuncKey;
```

```
var I : byte;
```

```
begin
```

```
window(1,1,80,25);
```

```
gotoxy(3,24);
```

```
SetAttr(HighDisplay);write('F1 ');
```

```
SetAttr(LowDisplay);write('Help');
```

```
gotoxy(13,24);
```

```
SetAttr(HighDisplay);write('F10 ');
```

```
SetAttr(LowDisplay);write('Exit');gotoxy(78,24);
```

```
end;( DisplayFuncKey)
```

```

procedure  Disp_SubFKey;
begin
    window(1,1,80,25);
    gotoxy(3,24);SetAttr(HighDisplay);
    write('F1  ');SetAttr(LowDisplay);
    write('Help ');gotoxy(13,24);
    setAttr(HighDisplay);write('Esc ');
    SetAttr(LowDisplay);write('Menu');gotoxy(78,24);
end;{ DisplayFuncKey}
procedure  ChoiceActiv(Old,New : byte; SubMenu : strucSub);
begin
    with  SubMenu[StatusChoi]  do
        begin
            Window(WinS[1]+1,WinS[2]+1,WinS[3],WinS[4]);
            { Display old choice with Normal Attribute }
            Setattr(LowDisplay);
            Gotoxy(ColS[old],RowS[old]);
            write(MsgS[old]);
            { Display new Choice with Reverse Attribute }
            SetAttr(ReverseLow);
            gotoxy(ColS[New],RowS[New]);
            write(MsgS[New]);
            Window(1,1,80,25);gotoxy(78,24);
        end;
    end;
end;
procedure  SmenuActive(New : byte ; SubMenu : strucSub);
var i : byte;
begin
    window(1,1,80,25);
    setattr(reverselow);
    with  SubMenu[New]  do

```



```

begin
    setwinattr(lowdisplay);
    setboxattr(lowdisplay);
    setcharattr(lowdisplay);
    SetBoxStyle(single);
    SetWinHeader('');
    WindowOpen(WinS[1],WinS[2],WinS[3],WinS[4]);
    { Display Choices of Menu }
    for i := 1 to LastChoi do
    begin
        gotoxy(ColS[i],RowS[i]);
        write(MsgS[i]);
    end;
end;

ChoiceActiv(1,ChoiSmenu[New],SubMenu);
end;

procedure MoveUpSmenu;
var CurrentChoi : byte;
begin
    CurrentChoi := ChoiSMenu[StatusChoi];
    if CurrentChoi = 1 then
        ChoiSMenu[StatusChoi] := SubMenu[StatusChoi].LastChoi
    else
        ChoiSMenu[StatusChoi] := CurrentChoi - 1;
    ChoiceActiv(CurrentChoi,ChoiSMenu[StatusChoi],SubMenu);
end;

procedure MoveDnSmenu;
var CurrentChoi : byte;
begin
    CurrentChoi := ChoiSMenu[StatusChoi];
    if CurrentChoi = SubMenu[StatusChoi].LastChoi then

```

```

        Choismenu[StatusChoi] := 1
    else
        Choismenu[StatusChoi] := CurrentChoi + 1;
        ChoiceActiv(CurrentChoi, Choismenu[StatusChoi], SubMenu);
    end;

procedure MovetoFirstChoi;
begin
    ChoiceActiv(Choismenu[StatusChoi], 1, SubMenu);
    Choismenu[StatusChoi] := 1;
end;

procedure MoveToLastchoi;
begin
    with SubMenu[StatusChoi] do
        begin
            ChoiceActiv(Choismenu[StatusChoi], LastChoi, SubMenu);
            Choismenu[StatusChoi] := LastChoi;
        end;
    end;
end;

procedure Do_SubFre3(var st : string);
var fix, sting : string;
begin
    close_win;
    open1(38, 39, 70, 71, 9, 10, 10, 11);
    gotoxy(40, 10); write('ระบุส่วนขยายหรือไม่ [Y/N] [ ]');
    gotoxy(68, 10); readchar(fix, 2);
    wind_close(' ', 38, 9, 71, 11);
    if fix[1] in ['Y', 'y'] then
        begin
            st := '/bar = ';
            Fregraph(sting);
        end;
    end;
end;

```

```

    if sting <> '' then
        st := st + sting
    else st := '';
end
else if fix[1] in['N','n'] then
    st := '/bar ';
SmenuActive(1,SubMenu);
end;

procedure Do_SubFre4(var st :string);
var fix,sting : string;
begin
    close_win;
    open1(38,39,70,71,9,10,10,11);
    gotoxy(40,10);write('ระบุส่วนขยายหรือไม่ [Y/N] [ ]');
    gotoxy(68,10);readchar(fix,1);
    wind_close(' ',38,9,71,11);
    if fix[1] in['Y','y'] then
        begin
            st := '/his = ';
            Fregraph(sting);
            if sting <> '' then
                st := st + sting
            else st := '';
        end
    else if fix[1] in['N','n'] then
        st := '/his ';
        SmenuActive(1,SubMenu);
    end;

procedure Do_SubFre5(var st : string);
var sting :string;
begin

```

```

close_win;
st := '/perc ';
Fpercent(sting);
if sting <> '' then
    st := st+sting
else
    st := '';
SmenuActive(1,SubMenu);
end;

procedure Do_SubFre6;
var num : integer;
    stnum : string;
begin
    close_win;
    open1(36,37,76,77,8,9,9,10);
    gotoxy(38,9);write('ระบุตัวเลขที่ต้องการแบ่งส่วน [      ]');
    gotoxy(68,9);readint(num,10,code);
    if num <> 0 then
        begin str(num,stnum);
            s1 := s1 + '/nti = ' + stnum + ' ';
        end;
    wind_close(' ',36,8,77,10);
    SmenuActive(1,SubMenu);
end;

procedure Do_SubOneway3;
var poly : integer;
    polst : string;
    j : integer;
begin
    j := 1;
    close_win;

```

```

open1(36,37,78,79,9,10,10,11);
gotoxy(38,10);write('กำหนดการแบ่งกลุ่มของแหล่งต้นแปร [   ]');
gotoxy(73,10);readint(poly,5,code);
if (poly <> 0) and (j = 1 ) then
  begin str(poly,polst);
        s1 := s1 + '/poly = '+ polst + ' ';
  end;
wind_close(' ',38,9,79,11);
SmenuActive(1,SubMenu);
j := j+1;
end;
{-----}
procedure Do_11(var sting : string);
var str : string;
begin
  case CurSub of
    1 : begin close_win;
          def_file(sting,str);
          if (str <> #13) and (sting <> #13) then
            if str[1] in['a','A'] then
              sting := 'dat lis fil = "' + sting + "' '
            else if str[1] in['s','S'] then
              begin
                sting := 'get fil = "' + sting + '".';
                erase(f);
                makefi(sting,f);
                sting := '';
              end;
            SmenuActive(1,SubMenu);
          end;
    3 : begin close_win;

```

```

        DefMiss;
        SmenuActive(1,SubMenu);
    end;
4 : begin close_win;
        DefValue;
        SmenuActive(1,SubMenu);
    end;
5 : begin close_win;
        Defnme_var;
        SmenuActive(1,SubMenu);
    end;
2 : begin SSmenu := Def_var;
        Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
    end;
end; { case }
end;
procedure Do_12;
begin
    case CurSub of
        1 : begin close_win;
                TrChange;
            end;
        2 : begin close_win;
                Trcompute;
            end;
        3 : begin close_win;
                Trif;
            end;
        4 : begin close_win;
                Trcount;
            end;
    end;
end;

```

```

        5 : begin close_win;
            Trweight;
        end;
    end; {case}
    SmenuActive(1,SubMenu);
end;

procedure Do_13;
var sting : string;
begin
    case CurSub of
        1 : begin close_win;
            SelectN;
        end;
        2 : begin close_win;
            SelecSamp;
        end;
        3 : begin close_win;
            sting := 'sel if (';
            SelecLogi(sting);
        end;
        4 : begin close_win;
            sting := 'pro if (';
            SelecLogi(sting);
        end;
    end; {case}
    SmenuActive(1,SubMenu);
end;

procedure Do_21;
var sting : string;
begin
    case CurSub of

```

```

1 : begin close_win;
      Frevar(sting,j);
      s:= s+sting;
      j :=j+1;
      SmenuActive(1,SubMenu);
      end;
2 : s := s + '/miss = include ';
3 : begin Do_SubFre3(st); s1 := s1 + st; end;
4 : begin Do_SubFre4(st); s1 := s1 + st; end;
5 : begin Do_SubFre5(st); s1 := s1 + st; end;
6 : Do_SubFre6;
7 : begin SSmenu := Fre_Stat;
      Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
      s1 := s1 + sting;
      end;
end; {case}
end;
procedure Do_22;
var sting : string;
begin
  case CurSub of
    1 : begin close_win;
          Descript_var(sting,j);
          s := s+sting;
          j := j+1;
          SmenuActive(1,SubMenu);
          end;
    2 : begin SSmenu := DesCal;
          Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
          s1 := s1+sting;

```



```

        end;
    3 : begin SMenu := Des_Stat;
        Init_Sub(SMenu, ShowStatus, Current, CurSub, sting);
        s1 := s1+sting;
        end;
    end; { case }
end;
procedure Do_23;
var sting : string;
begin
    case CurSub of
        1 : begin close_win;
            Meanvar(sting, j);
            s := s+sting;
            j := j+1;
            SmenuActive(1, SubMenu);
            end;
        2 : begin close_win;
            MeanGr(sting);
            s := s+sting;
            SmenuActive(1, SubMenu);
            end;
        3 : begin SMenu := Mean_Form;
            Init_Sub(SMenu, ShowStatus, Current, CurSub, sting);
            s1 := s1+sting;
            end;
        4 : begin SMenu := Mean_Stat;
            Init_Sub(SMenu, ShowStatus, Current, CurSub, sting);
            s1 := s1+sting;
            end;
    end; { case }
end;

```

```

        end;
    3 : begin SMenu := Des_Stat;
        Init_Sub(SMenu, ShowStatus, Current, CurSub, sting);
        s1 := s1+sting;
        end;
    end; { case }
end;
procedure Do_23;
var sting : string;
begin
    case CurSub of
        1 : begin close_win;
            Meanvar(sting, j);
            s := s+sting;
            j := j+1;
            SmenuActive(1, SubMenu);
            end;
        2 : begin close_win;
            MeanGr(sting);
            s := s+sting;
            SmenuActive(1, SubMenu);
            end;
        3 : begin SMenu := Mean_Form;
            Init_Sub(SMenu, ShowStatus, Current, CurSub, sting);
            s1 := s1+sting;
            end;
        4 : begin SMenu := Mean_Stat;
            Init_Sub(SMenu, ShowStatus, Current, CurSub, sting);
            s1 := s1+sting;
            end;
    end; { case }
end;

```

end;

procedure Do_24;

var sting : string;

begin

case CurSub of

1 : begin close_win;

Crossvar(sting,j);

s := s+sting;

j := j+1;

SmenuActive(1,SubMenu);

end;

2 : begin SSmenu := Cross_Form;

Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);

s1 := s1+sting;

end;

3 : begin SSmenu := Cross_Stat;

Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);

s1 := s1+sting;

end;

end; { case }

end;

procedure Do_25;

var sting : string;

begin

case CurSub of

1 : begin SSmenu := Ttest_depen;

Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);

s := s+sting;

end;

2 : begin close_win;

Tdep_va(s);

```

        s := s+sting;
        SmenuActive(1,SubMenu);
    end;
3 : begin SSmenu := Ttest_opt;
        Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
        s1 := s1+sting;
    end;
end; { case }
end;
procedure Do_26;
var sting : string;
begin
    case CurSub of
        1 : begin close_win;
                Onewayvar(sting);
                s := s+sting;
                SmenuActive(1,SubMenu);
            end;
        2 : begin close_win;
                OnewayGr(sting);
                s := s+sting;
                SmenuActive(1,SubMenu);
            end;
        3 : Do_SubOneway3;
        4 : begin close_win;
                OnewayVec(sting);
                s1 := s1 + sting;
                SmenuActive(1,SubMenu);
            end;
        5 : begin SSmenu := Oneway_Stat;
                Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
    end;
end;

```

```

        s1 := s1 + sting; end;
    6 : begin  SSmenu := Oneway_Form;
        Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
        s1 := s1 + sting; end;
    7 : begin  SSmenu := Oneway_Statval;
        Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
        s1 := s1 + sting; end;

end;

end;

procedure Do_27;
var sting : string;
begin
    case CurSub of
        1 : begin  close_win;
                anovar(sting);
                s := s + sting;
                SmenuActive(1, SubMenu);
            end;
        2 : begin  close_win;
                devi(sting);
                s := s + sting;
                SmenuActive(1, SubMenu);
            end;
        3 : begin  close_win;
                Covar(sting);
                s1 := s1 + sting;
                SmenuActive(1, SubMenu);
            end;
        4 : begin  SSmenu := Ano_Form;
                Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
                s1 := s1 + sting;

```

```

        end;
    5 : begin  SSmenu := Stat_ValAno;
          Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
          s1 := s1 + sting;
        end;
    end; {case}
end;

procedure Do_28;
var sting : string;
begin
    case CurSub of
        1 : begin  close_win;
              Corrvar(sting);
              s := s+sting;
              SmenuActive(1, SubMenu);
            end;
        2 : begin  SSmenu := Corr_form;
              Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
              s1 := s1+sting;
            end;
        3 : begin  SSmenu := Corr_stat;
              Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
              s1 := s1+sting;
            end;
    end; {case}
end;

procedure Do_29;
var sting : string;
begin
    case CurSub of
        1 : begin  close_win;

```

```

        Regression_var(sting);
        s := s+sting;
        SmenuActive(1,SubMenu);
    end;
2 : begin SSmenu := ReGr_Selvar;
        Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
        s1 := s1+sting;
    end;
end; { case }
end;
procedure Do_210;
var sting : string;
begin
    if j = 1 then
        s := 'plot ';
    case CurSub of
        1 : begin SSmenu := plot_scale;
                Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
                if (j = 1) or (s = 'plot ') then
                    s := s + sting
                else s := s + '/' + sting;
            end;
        2 : begin SSmenu := plot_scale;
                Init_Sub(SSmenu,ShowStatus,Current,CurSub,sting);
                if (j = 1) or (s = 'plot ') then
                    s := s + sting
                else s := s + '/' + sting;
            end;
        3 : begin close_win;
                open1(36,37,77,78,9,10,10,11);
                gotoxy(45,9);write(' กำหนดชื่อของกราฟ ');
            end;
    end;
end;

```

```

        gotoxy(38,10);readchar(sting,40);
        if (sting <> #13) then
            if j = 1 then
                s1 := 'tit = ' + sting + ' '
            else s1 := s1 + '/tit = "' + sting + '" ' ;
        wind_close(' ',36,9,78,11);
        SmenuActive(1,SubMenu);

    end;
4 : begin close_win;
        plot_var(sting);
        if sting <> #13 then
            if j = 1 then
                s1 := sting
            else
                s1 := s1 + '/' + sting;
        SmenuActive(1,SubMenu);

    end;

end;

if (sting = '') and ((s = 'plot ') and (s1 = '')) then
    s := '';
j := j +1;

end;

procedure Do_211;
var sting : string;
procedure npa1(sign : string);
begin
    close_win;
    npar_with(sting);
    if sting <> '' then
        if j = 1 then
            s1 := s1 + sign + sting

```



```

        else s1 := s1 + '/' + sign + sting;
    SmenuActive(1,SubMenu);
end;
procedure npa2(sign : string);
begin
    close_win;
    npva(sting);
    if sting <> '' then
        if j = 1 then
            s1 := s1 + sign + sting
        else s1 := s1 + '/' + sign + sting;
        SmenuActive(1,SubMenu);
end;
procedure npa3(sign : string);
begin
    close_win;
    nby(sting);
    if sting <> '' then
        if j = 1 then
            s1 := s1 + sign + sting
        else s1 := s1 + '/' + sign + sting;
        SmenuActive(1,SubMenu);
end;
begin
    case CurSub of
        1 : npa2('chi = ');
        2 : npa2('bin = ');
        3 : npa1('mcne = ');
        4 : npa1('wil = ');
        5 : npa2('coc = ');
        6 : npa2('fri = ');

```

```

    7 : npa2('ken = ');
    8 : npa3('med = ');
    9 : npa3('m-w = ');
   10 : npa3('k-d = ');
   11 : npa3('w-w = ');
   12 : npa3('k-w = ');
   13 : begin  SSmenu := Npar_Form;
           Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
           s1 := s1 + sting;
           end;
   14 : begin  SSmenu := Npar_Stat;
           Init_Sub(SSmenu, ShowStatus, Current, CurSub, sting);
           s1 := s1 + sting;
           end;
       end;
   j := j + 1;
end;
{-----}
procedure Do_Ssub1(var sting : string);
begin
    case Current of
        1 : Do_11(sting);
        2 : Do_12;
        3 : Do_13;
    end; {case Current}
end;
procedure Do_Ssub2;
begin
    case Current of
        1 : Do_21;
        2 : Do_22;
    end;
end;

```

```

3 : Do_23;
4 : Do_24;
5 : Do_25;
6 : Do_26;
7 : Do_27;
8 : Do_28;
9 : Do_29;
10 : Do_210;
11 : begin Do_211;
        if s1 <> '' then
            s := 'npar tes';
        end;
    end; {case Current}
end;
procedure Do_Ssub3;
var stlis,stex : string;
begin
    case Current of
        1 : begin case CurSub of
                1 : begin close_win;
                        Uvar(stlis);
                        s := s+stlis;
                        SmenuActive(1,SubMenu);
                    end;
                2 : begin close_win;
                        Ulocate(s1);
                        SmenuActive(1,SubMenu);
                    end;
            end; { case }
        end;
    3,4 : begin case CurSub of

```

```

end; { case }
    end;
end; {case Current}
end;
procedure DoSMenu(var s1 : string);
begin
    window(1,1,80,25);
    gotoxy(3,24);SetAttr(HighDisplay);write('Esc ');
    SetAttr(LowDisplay);write('Menu ');
    wind_close(' ',13,24,23,24);
    CurSub := ChoiSMenu[StatusChoi];
    case ShowStatus of
        1 : Do_Ssub1(s1);
        2 : Do_Ssub2;
        3 : Do_Ssub3;
    end; { case }
    Disp_SubFKey;
end;
procedure Do_help1;
begin
    case Current of
        1 : Read_txt('c:\sptool\help\defi.txt');
        2 : begin Read_txt('c:\sptool\help\recode.txt');readln;
            Read_txt('c:\sptool\help\compute.txt');
            end;
        3 : Read_txt('c:\sptool\help\sel.txt');
    end;
    readln; gotoxy(78,24);
end;
procedure Do_help2;
begin

```

```

case Current of
  1 : Read_txt('c:\sptool\help\freq.txt');
  2 : Read_txt('c:\sptool\help\des.txt');
  3 : Read_txt('c:\sptool\help\mean.txt');
  4 : Read_txt('c:\sptool\help\cross.txt');
  5 : Read_txt('c:\sptool\help\t_test.txt');
  6 : Read_txt('c:\sptool\help\one.txt');
  7 : Read_txt('c:\sptool\help\anova.txt');
  8 : Read_txt('c:\sptool\help\corr.txt');
  9 : Read_txt('c:\sptool\help\regress.txt');
 10 : Read_txt('c:\sptool\help\plot.txt');
 11 : Read_txt('c:\sptool\help\ncpar.txt');
end;
readln; gotoxy(78,24);
end;
procedure DoF1;
begin
  save_scr;
  case ShowStatus of
    1 : Do_help1;
    2 : Do_help2;
  end; { case }
  get_oldscr;
end;
procedure QuitS;
begin
  windowclose;
  DisplayFuncKey;
  Succ := true;
end;
procedure TestKeyS(KeyS: char; var s1: string);

```

```

begin
  if KeyS = Esc_Key then Quits;
  if FuncKey then
    case (KeyS) of
      Up_Key      : MoveUpSmenu;
      Dn_Key      : MoveDnSmenu;
      PgUp_Key    : MoveToFirstchoi;
      PgDn_Key    : MoveToLastchoi;
      F1_Key      : DoF1;
    end { of Case }
  else
    if KeyS = Return_Key Then DoSMenu(s1);
  end;
procedure  Initsub(SubM : StrucSub;StatusMenu,CurChoice : byte);
var  stin : string;
     word,cnt : byte;
begin
  Disp_SubFKey;
  assign(f,'c:\sptool\ex.sps');
  s := ''; s1 := '';
  Succ := false;
  ShowStatus := StatusMenu;
  Current := CurChoice;
  SubMenu := SubM;
  ChoiSmenu[1] := 1;
  SmenuActive(1,SubMenu);
  j := 1;
  repeat
    ReadFuncKey(KeyS);
    TestkeyS(keyS,s1);
  until Succ;

```

```

if (s <> '') and (s1 = '') then
  begin
    s := s+'.';
    makefi(s,f);
  end
else if (s <> '') and (s1 <> '') then
  begin makefi(s,f);
        checkword(s1,word,cnt);
        if word > 5 then
          repeat
            stin := copy(s1,1,cnt);
            makefi(stin,f);
            delete(s1,1,cnt+1);
            checkword(s1,word,cnt);
          until word <= 5 ;
        s1 := s1 + '.';
        makefi(s1,f);
      end;
end;
end;
begin
end.

```



```

' Uncertainty coefficient          '
' kendall's tau-b                  '
' kendall's tau-c                  '
' Gamma                            '
' Somer'd                          '
' Eta                              '
' Pearson's r                      '
'', '', '', '' );

```

```
LastChoi:11));
```

```
DesCal : StrucSub = (
```

```
(WinS : (40, 6, 74, 12);
```

```
ColS : (02, 02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
```

```
RowS : (01, 02, 03, 04, 05, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
```

```
MsgS : (
```

```
' รวมค่า ไม่สมบูรณ์          ',
```

```
' ไม่พิมพ์ค่าขยายตัวแปร      ',
```

```
' ไม่รวมค่า ไม่สมบูรณ์          ',
```

```
' พิมพ์ค่าสถิติได้ชื่อตัวแปร  ',
```

```
' ไม่พิมพ์ชื่อตัวแปร          ',
```

```
'', '', '' ,
```

```
'', '', '', '', '', '', '' );
```

```
LastChoi:5));
```

```
Des_Stat : StrucSub = (
```

```
(WinS : (43, 5, 71, 17);
```

```
ColS : (02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 00, 00, 00, 00
```

```
RowS : (01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 00, 00, 00, 00
```

```
MsgS : (
```

```
' ค่าเฉลี่ย          ',
```

```
' ความคลาดเคลื่อนมาตรฐาน ',
```

```
' ส่วนเบี่ยงเบนมาตรฐาน  ',
```

```
' ความแปรปรวน          ',
```

```

' ความโค้ง          ',
' ความเบี้ย          ',
' พิสัย            ',
' ค่าต่ำสุด        ',
' ค่าสูงสุด        ',
' ผลรวม            ',
' แสดงค่าสถิติทั้งหมด ',
'', '', '', '' );

```

```
LastChoi:11));
```

```
Fre_Stat : StrucSsub = (
```

```
(WinS : (35, 4, 74, 20);
```

```
ColS : (02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02,
```

```
RowS : (01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15
```

```
MsgS : (
```

```

' ค่าเฉลี่ย          ',
' ค่าคลาดเคลื่อนมาตรฐานของค่าเฉลี่ย ',
' มัชยฐาน          ',
' ฐานนิยม          ',
' ส่วนเบี่ยงเบนมาตรฐาน ',
' ความแปรปรวน      ',
' ความโค้ง          ',
' ค่าคลาดเคลื่อนมาตรฐานของความโค้ง ',
' ความเบี้ย          ',
' ค่าคลาดเคลื่อนมาตรฐานของความเบี้ย ',
' พิสัย            ',
' ค่าต่ำสุด        ',
' ค่าสูงสุด        ',
' ผลรวม            ',
' แสดงค่าสถิติทั้งหมด ',

```

```
);
```

```
LastChoi:15));
```

```

Mean_Form : StrucSsub = (
    (WinS : (28,07,77,17));
    ColS : (02,02,02,02,02,02,02,02,02,00,00,00,00,00,00);
    RowS : (01,02,03,04,05,06,07,08,09,00,00,00,00,00,00);
    MsgS : (
        ' รวมค่า ไม่สมบูรณ์ ',
        ' ตัดค่า ไม่สมบูรณ์ของตัวอย่าง ',
        ' พิมพ์ผลรวมของกลุ่มย่อย ',
        ' ไม่พิมพ์ค่าส่วนเบี่ยงเบนมาตรฐานของกลุ่มย่อย ',
        ' ไม่พิมพ์ส่วนขยายของค่าตัวแปร ',
        ' ไม่พิมพ์ชื่อตัวแปรแบ่งกลุ่ม ',
        ' ไม่พิมพ์ค่าตัวแปรแบ่งกลุ่ม ',
        ' ไม่พิมพ์ค่าของค่าเฉลี่ยในแต่ละกลุ่มย่อย ',
        ' พิมพ์ค่าความแปรปรวนในแต่ละกลุ่มย่อย ',
        '','','',' ',' ',' ');
    LastChoi:9));

Mean_Stat : StrucSsub = (
    (WinS : (35,10,73,13));
    ColS : (02,02,00,00,00,00,00,00,00,00,00,00,00,00,00);
    RowS : (01,02,00,00,00,00,00,00,00,00,00,00,00,00,00);
    MsgS : (
        ' พิมพ์ตารางความแปรปรวนแบบทางเดียว ',
        ' พิมพ์การทดสอบเชิงเส้น ',
        '','','',' ',' ',' ',' ',' ',' ',' ');
    LastChoi:2));

ReGr_Selvar : StrucSsub = (
    (WinS : (60,9,74,16));
    ColS : (02,02,02,02,02,02,00,00,00,00,00,00,00,00,00);
    RowS : (01,02,03,04,05,06,00,00,00,00,00,00,00,00,00);
    MsgS : (
        ' Stepwise ',

```

```

' Forward ',
' Backward ',
' Enter ',
' Remove ',
' Test ',
'', '', '', '', '',
'', '', '', '');

```

```
LastChoi:6));
```

```
Oneway_stat : StrucSsub = (
```

```
(WinS : (60, 13, 74, 21));
```

```
ColS : (02, 02, 02, 02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
RowS : (01, 02, 03, 04, 05, 06, 07, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
MsgS : (
```

```
' LSD ',
```

```
' DUNCAN ',
```

```
' SNK ',
```

```
' TUKEY ',
```

```
' BTUKEY ',
```

```
' MODLSD ',
```

```
' SCHEFFE ',
```

```
'', '', '',
```

```
'', '', '', '', '');
```

```
LastChoi:7));
```

```
Oneway_Form : StrucSsub = (
```

```
(WinS : (23, 15, 68, 20));
```

```
ColS : (02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
RowS : (01, 02, 03, 04, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
MsgS : (
```

```
' รวมค่าไม่สมบูรณ์ ',
```

```
' ตัดข้อมูลที่มีค่าไม่สมบูรณ์ ',
```

```

' ไม่พิมพ์ส่วนขยายและตัวแปร      ',
' พิมพ์ค่าสถิติเบื้องต้นในรูปแบบเตอร์ ',
'', '', '', '', '', '',
'', '', '', '', '');

```

```
LastChoi:4));
```

```
Oneway_Statval: StrucSsub = (
```

```
(WinS : (20, 15, 70, 19);
```

```
ColS : (02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
RowS : (01, 02, 03, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
MsgS : (
```

```
' พิมพ์ค่าสถิติเบื้องต้น      ',
```

```
' พิมพ์รูปแบบการวิเคราะห์ความแปรปรวน      ',
```

```
' พิมพ์ค่าสถิติที่ทดสอบความเท่ากันในแต่ละกลุ่ม      ',
```

```
'', '', '', '', '', '', '', '', '', '', '', '', '');
```

```
LastChoi:3));
```

```
Npar_Form : StrucSsub = (
```

```
(WinS : (18, 16, 69, 21);
```

```
ColS : (02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
RowS : (01, 02, 03, 04, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
MsgS : (
```

```
' รวมค่าไม่สมบูรณ์
```

```
' ไม่ตัดข้อมูลชุดที่มีค่าตัวแปรหนึ่งในคำสั่งช่วย      ',
```

```
' แสดง chi-square residuals
```

```
' สุ่มตัวอย่างแบบ Random
```

```
'', '', '', '', '', '', '', '', '', '', '', '', '');
```

```
LastChoi:4));
```

```
Npar_Stat : StrucSsub = (
```

```
(WinS : (18, 16, 58, 20);
```

```
ColS : (01, 01, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
RowS : (01, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
```

```
MsgS : (
```

```

' พิมพ์ Quatiles และ count      ',
' พิมพ์ค่าเฉลี่ย,ค่าสูงสุด,ค่าต่ำสุด, ส่วนเบี่ยงเบนมาตรฐาน
'', '', '', '', '', '', '', '', '', '', '', '', '', '');

```

```

LastChoi:2));

```

```

Plot_scale : StrucSsub = (

```

```

(WinS : (18, 15, 60, 21));

```

```

ColS : (02, 02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);

```

```

RowS : (01, 02, 03, 04, 05, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);

```

```

MsgS : (

```

```

' กำหนดข้อความเพื่อแสดงในแกน      ',

```

```

' ใช้ค่าคะแนนมาตรฐานในการเขียนกราฟ  ',

```

```

' แสดงค่าที่ระบุในส่วนของ VALUES      ',

```

```

' ระบุค่าต่ำสุด      ',

```

```

' ระบุค่าสูงสุด      ',

```

```

'', '', '', '', '', '', '', '', '', '', '', '');

```

```

LastChoi:5));

```

```

Ttest_depen : StrucSsub = (

```

```

(WinS : (20, 12, 54, 15));

```

```

ColS : (02, 02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);

```

```

RowS : (01, 02, 03, 04, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);

```

```

MsgS : (

```

```

' ตัวแปร      ',

```

```

' แบ่งกลุ่มข้อมูลเป็น 2 กลุ่ม  ',

```

```

'', '',

```

```

'', '', '', '', '', '', '', '', '', '', '', '');

```

```

LastChoi:2));

```

```

Ttest_opt : StrucSsub = (

```

```

(WinS : (20, 12, 59, 16));

```

```

ColS : (02, 02, 02, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);

```

```

RowS : (01, 02, 03, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00);

```

```

MsgS : (' รวมค่าไม่สมบูรณ์      ',

```



```

' ตัดข้อมูลที่ตัวแปรมีค่าไม่สมบูรณ์ ',
' ไม่พิมพ์คำอธิบายชื่อตัวแปร ',
'', '', '', '', '', '', '', '', '', '', '', '', '');

```

```

LastChoi:3));

```

```

procedure Init_Sub (SS_Menu:StrucSsub;

```

```

                Status,SCurr,CurSub: byte;var s : string);

```

```

procedure close_win;

```

```

implementation

```

```

Const ChoiSsub : array[1..1] of 1..MaxChoi = (1);

```

```

    Status_Ssub = 1;

```

```

    Status_Schoi : 1..MaxChoi = 1;

```

```

    Ok          : Boolean = true;

```

```

var Ksub       : char;

```

```

    j           : byte;

```

```

    Ssub_Menu   : StrucSsub;

```

```

    ShowStatus,Current,SubCur : byte;

```

```

    st1,st2    : string;

```

```

procedure ChoiActiv(Old,New : byte; {var} Ssub_Menu : StrucSsub);

```

```

begin

```

```

    with Ssub_Menu[Status_Ssub] do

```

```

        begin

```

```

            Window(WinS[1]+1,WinS[2]+1,WinS[3],WinS[4]);

```

```

            { Display old choice with Normal Attribute }

```

```

            Setattr(LowDisplay);

```

```

            Gotoxy(ColS[old],RowS[old]);

```

```

            write(MsgS[old]);

```

```

            { Display new Choice with Reverse Attribute }

```

```

    SetAttr(ReverseLow);
    gotoxy(ColS[New],RowS[New]);
    write(MsgS[New]);
    SetAttr(Nodisplay);
    gotoxy(WinS[3],RowS[New]);write(' ');
    Window(1,1,80,24);gotoxy(77,24);write(' ');
end;
end;

procedure SsubActive(New : byte ; SS_Menu :StrucSsub);
var i : byte;
begin
    setattr(reverselow);
    with Ssub_Menu[New] do
    begin
        setwinattr(lowdisplay);
        setboxattr(lowdisplay);
        setcharattr(lowdisplay);
        SetBoxStyle(single);
        SetWinHeader('');
        WindowOpen(WinS[1],WinS[2],WinS[3],WinS[4]);
        { Display Choices of Menu }
        for i := 1 to LastChoi do
        begin
            gotoxy(ColS[i],RowS[i]);
            write(MsgS[i]);
        end;
    end;
end;

ChoiActiv(1,ChoiSsub[New],Ssub_Menu);
end;

```

```
procedure MoveUpSsub;
```

```
var CurrentChoi : byte;
```

```
begin
```

```
    CurrentChoi := ChoiSsub[status_Ssub];
```

```
    if CurrentChoi = 1 then
```

```
        ChoiSsub[Status_Ssub] := Ssub_Menu[Status_Ssub].LastChoi
```

```
    else
```

```
        ChoiSsub[Status_Ssub] := CurrentChoi - 1;
```

```
        ChoiActiv(CurrentChoi, ChoiSsub[Status_Ssub], Ssub_Menu);
```

```
end;
```

```
procedure MoveDnSsub;
```

```
var CurrentChoi : byte;
```

```
begin
```

```
    CurrentChoi := ChoiSsub[Status_Ssub];
```

```
    if CurrentChoi = Ssub_Menu[Status_Ssub].LastChoi then
```

```
        ChoiSsub[Status_Ssub] := 1
```

```
    else
```

```
        ChoiSsub[Status_Ssub] := CurrentChoi + 1;
```

```
        ChoiActiv(CurrentChoi, ChoiSsub[Status_Ssub], Ssub_Menu);
```

```
end;
```

```
procedure MoveToFirstSchoi;
```

```
begin
```

```
    ChoiActiv(ChoiSsub[Status_Ssub], 1, Ssub_Menu);
```

```
    ChoiSsub[Status_Ssub] := 1;
```

```
end;
```

```
procedure MoveToLastSchoi;
```

```
begin
```

```
    with Ssub_Menu[Status_Ssub] do
```

```

begin
    ChoiActiv(ChoiSsub[Status_Ssub], LastChoi, Ssub_Menu);
    ChoiSsub[Status_Ssub] := LastChoi;
end;
end;

procedure close_win;
begin
    windowclose;
    window(1,1,80,25);
    SetAttr(lowdisplay);
end;
{-----}
procedure Perform_Anov4(var st : string);
var i : string;
begin
    if j = 1 then
        begin
            st := '/opt ';
            case ChoiSsub[Status_Ssub] of
                1 : st := st + '1 ';
                2 : st := st + '3 ';
                3..8 : begin str(ChoiSsub[Status_Ssub]+1, i);
                        st := st + i + ' ';
                    end;
            end;
        end;
    end;
end
else
    case ChoiSsub[Status_Ssub] of
        1 : st := st + '1 ';
        2 : st := st + '3 ';
    end;
end;

```

```

    3..8 : begin  str(ChoiSsub[Status_Ssub]+1,i);
           st := st + i + ' ';
           end;
    end; {case}
    j := j+1;
end;

```

```

procedure Perform_Anov5(var st : string);
begin
    if j = 1 then
        begin st := '/stat ';
            case ChoiSsub[Status_Ssub] of
                1 : st := st + '1 ';
                2 : st := st + '2 ';
                3 : st := st + '3 ';
            end; {case}
        end
    else
        case ChoiSsub[Status_Ssub] of
            1 : st := st + '1 ';
            2 : st := st + '2 ';
            3 : st := st + '3 ';
        end; {case}
        j := j + 1;
    end;
end;

```

```

procedure Perform_Corr2(var stcorr : string);
var i : string;
begin
    if j = 1 then
        begin

```

```

stcorr := '/opt ';
case ChoiSsub[Status_Ssub] of
  1..3 : begin str(ChoiSsub[Status_Ssub],i);
           stcorr := stcorr + i + ' ';
        end;
  4 : stcorr := stcorr + '5 ';
end; {case}
end
else
begin
  case ChoiSsub[Status_Ssub] of
    1..3 : begin str(ChoiSsub[Status_Ssub],i);
               stcorr := stcorr + i + ' ';
          end;
    4 : stcorr := stcorr + '5 ';
  end; {case}
end;
j := j+1;
end;

```

```

procedure Perform_Corr3(var stcor3 : string);
begin
  if j = 1 then
  begin
    stcor3 := '/stat ';
    case ChoiSsub[Status_Ssub] of
      1 : stcor3 := stcor3 + '1 ';
      2 : stcor3 := stcor3 + '2 ';
    end; {case}
  end
else

```

```
begin
  case ChoiSsub[Status_Ssub] of
    1 : stcor3 := stcor3 + '1 ';
    2 : stcor3 := stcor3 + '2 ';
  end; {case}
end;
j := j+1;
end;

procedure Perform_Cros2(var stcros : string);
begin
  if j = 1 then
    begin
      stcros := '/opt ';
      case ChoiSsub[Status_Ssub] of
        1 : stcros := stcros + '8 ';
        2 : stcros := stcros + '9 ';
        3 : stcros := stcros + '15 ';
        4 : stcros := stcros + '16 ';
        5 : stcros := stcros + '17 ';
      end; {case}
    end
  else
    begin
      case ChoiSsub[Status_Ssub] of
        1 : stcros := stcros + '8 ';
        2 : stcros := stcros + '9 ';
        3 : stcros := stcros + '15 ';
        4 : stcros := stcros + '16 ';
        5 : stcros := stcros + '17 ';
      end; {case}
    end
  end;
end;
```

```

    end;
    j := j+1;
end;

procedure Perform_Cros3(var steros : string);
var i : string;
begin
    if j = 1 then
        begin
            steros := '/stat ';
            case ChoiSsub[Status_Ssub] of
                1..11 : begin str(ChoiSsub[Status_Ssub],i);
                           steros := steros + i + ' ';
                           end;
            end; {case}
        end
    else
        case ChoiSsub[Status_Ssub] of
            1..11 : begin str(ChoiSsub[Status_Ssub],i);
                       steros := steros + i + ' ';
                       end;
            end; {case}
        j := j+1;
    end;
end;

```

```

procedure Perform_Des2(var stdes:string);
begin
    if j = 1 then
        begin
            stdes := '/opt ';
            case ChoiSsub[Status_Ssub] of

```



```

    1 : stdes := stdes + '1 ';
    2 : stdes := stdes + '2 ';
    3 : stdes := stdes + '5 ';
    4 : stdes := stdes + '6 ';
    5 : stdes := stdes + '8 ';
  end; {case}
end
else
  begin
    case ChoiSsub[Status_Ssub] of
      1 : stdes := stdes + '1 ';
      2 : stdes := stdes + '2 ';
      3 : stdes := stdes + '5 ';
      4 : stdes := stdes + '6 ';
      5 : stdes := stdes + '8 ';
    end; {case}
  end;
  j := j+1;
end;

procedure Perform_Des3(var stdes : string);
var i : string;
begin
  if j = 1 then
    begin
      stdes := '/stat ';
      case ChoiSsub[Status_Ssub] of
        1 : stdes := stdes + '1 ';
        2 : stdes := stdes + '2 ';
        3..11 : begin str(ChoiSsub[Status_Ssub]+2,i);
                    stdes := stdes + i + ' ';
                end;
      end;
    end;
  end;
end;

```

```

                end;
    end; {case}
end
else
    case ChoiSsub[Status_Ssub] of
        1 : stdes := stdes + '1 ';
        2 : stdes := stdes + '2 ';
        3..11 : begin str(ChoiSsub[Status_Ssub]+2,i);
                    stdes := stdes + i + ' ';
                end;
    end; {case}
    j := j+1;
end;

```

```

procedure Perform_Mean3(var stmean1 : string);
var i :string;
begin
    if j = 1 then
        begin
            stmean1 := '/opt ';
            case ChoiSsub[Status_Ssub] of
                1 : stmean1 := stmean1 + '1 ';
                2 : stmean1 := stmean1 + '2 ';
                3..9 : begin str(ChoiSsub[Status_Ssub]+3,i);
                        stmean1 := stmean1 + i + ' ';
                    end;
            end;
        end; {case}
    end
else
    case ChoiSsub[Status_Ssub] of
        1 : stmean1 := stmean1 + '1 ';

```

```

    2 : stmean1 := stmean1 + '2 ' ;
    3..9 : begin str(ChoiSsub[Status_Ssub]+3,i);
           stmean1 := stmean1 + i + ' ' ;
        end;
    end; {case}
    j := j+1;
end;

```

```

procedure Perform_Mean4(var stmean2 : string);
begin
    if j = 1 then
        begin
            stmean2 := '/stat ' ;
            case ChoiSsub[Status_Ssub] of
                1 : stmean2 := stmean2 + '1 ' ;
                2 : stmean2 := stmean2 + '2 ' ;
            end; {case}
        end
    else
        case ChoiSsub[Status_Ssub] of
            1 : stmean2 := stmean2 + '1 ' ;
            2 : stmean2 := stmean2 + '2 ' ;
        end; {case}
        j := j+1;
    end;
end;

```

```

procedure Perform_Oneway5(var stone : string);
begin
    stone := '/rang = ' ;
    case ChoiSsub[Status_Ssub] of
        1 : stone := stone + 'lsd ( ) ' ;
    end;
end;

```

```

2 : stone := stone + 'duncan () ';
3 : stone := stone + 'snk ';
4 : stone := stone + 'tukey ';
5 : stone := stone + 'btukey ';
6 : stone := stone + 'modlsd () ';
7 : stone := stone + 'scheffe () ';

```

```
end;
```

```
end;
```

```
procedure Perform_Oneway6(var stopt : string);
```

```
var i : string;
```

```
begin
```

```
  if j = 1 then
```

```
    begin
```

```
      stopt := '/opt = ';
```

```
      case ChoiSsub[Status_Ssub] of
```

```
        1..4 : begin str(ChoiSsub[Status_Ssub],i);
```

```
                stopt := stopt + i + ' ';
```

```
            end;
```

```
      end; {case}
```

```
    end
```

```
  else
```

```
    case ChoiSsub[Status_Ssub] of
```

```
      1..4 : begin str(ChoiSsub[Status_Ssub],i);
```

```
                stopt := stopt + i + ' ';
```

```
    end;
```

```
    end; {case}
```

```
  j := j + 1;
```

```
end;
```

```
procedure Perform_Oneway7(var stone : string);
```

```

begin
  if j = 1 then
    begin
      stone := '/stat = ';
      case ChoiSsub[Status_Ssub] of
        1 : stone := stone + '1 ';
        2 : stone := stone + '2 ';
        3 : stone := stone + '3 ';
      end; {case}
    end
  else
    case ChoiSsub[Status_Ssub] of
      1 : stone := stone + '1 ';
      2 : stone := stone + '2 ';
      3 : stone := stone + '3 ';
    end; {case}
  j := j + 1;
end;

procedure Perform_Plot1(var stplot : string);
var sti, valst : string;
    valu      : real;
begin
  if j <> 1 then
    stplot := '/' + stplot;
  case ChoiSsub[Status_Ssub] of
    1 : begin close_win;
          open1(20,21,70,71,17,18,18,19);
          gotoxy(22,18);write('กำหนดข้อความเพื่อแสดงในแกน ');
          gotoxy(50,18);readchar(sti,30);
          if sti <> #13 then

```

```

        stplot := stplot + ""+sti + ""';
wind_close(' ',20,17,71,19);
SsubActive(1,Ssub_menu);

end;
2 : stplot := stplot + 'stand';
3 : begin close_win;
    open1(20,21,67,68,17,18,18,19);
    gotoxy(22,18);write('แสดงค่าที่ระบุในส่วนของ VALUES');
    gotoxy(55,18);readreal(valu,8,code);
    if valu <> 0 then
        begin
            str(valu:5:2,valst);
            stplot := stplot + 'ref('+ valst + ')';
        end;
    wind_close(' ',20,17,68,19);
    SsubActive(1,Ssub_menu);

end;
4 : begin close_win;
    open1(20,21,67,68,17,18,18,19);
    gotoxy(22,18);write('ระบุค่าต่ำสุดที่ใช้เขียนกราฟ');
    gotoxy(55,18);readreal(valu,8,code);
    if valu <> 0 then
        begin str(valu:7:2,valst);
            stplot := stplot + 'min('+ valst + ')';
        end;
    wind_close(' ',20,17,68,19);
    SsubActive(1,Ssub_menu);

end;
5 : begin close_win;
    open1(20,21,67,68,17,18,18,19);
    gotoxy(22,18);write('ระบุค่าสูงสุดที่ใช้เขียนกราฟ');

```

```

        gotoxy(55,18);readreal(valu,8,code);
        if valu <> 0 then
            begin str(valu:7:2,valst);
                stplot := stplot + 'max('+ valst + ')' ';
            end;
        wind_close(' ',20,17,68,19);
        SsubActive(1,Ssub_menu);
    end;
end; {case}
j := j + 1;
end;

procedure Perform_Npar1(var stNpa : string);
var i : string;
begin
    if j = 1 then
        begin
            stNpa := '/opt = ';
            case ChoiSsub[Status_Ssub] of
                1..4 : begin str(ChoiSsub[Status_Ssub],i);
                    stNpa := stNpa + i + ' ';
                end;
            end; {case}
        end
    else
        case ChoiSsub[Status_Ssub] of
            1..4 : begin str(ChoiSsub[Status_Ssub],i);
                stNpa := stNpa + i + ' ';
            end;
        end; {case}
    j := j + 1;
end;

```

```

end;

procedure Perform_Npar2(var stNpa : string);
begin
  if j = 1 then
    begin stNpa := '/stat = ';
      case ChoiSsub[Status_Ssub] of
        1 : stNpa := stNpa + '2';
        2 : stNpa := stNpa + '1';
      end; {case}
    end
  else
    case ChoiSsub[Status_Ssub] of
      1 : stNpa := stNpa + '2';
      2 : stNpa := stNpa + '1';
    end; {case}
  j := j + 1;
end;

```

```

procedure Perform_21(var stf :string);
begin
  if j = 1 then
    begin
      stf := '/stat ';
      case ChoiSsub[Status_Ssub] of
        1 : stf := stf + 'mean ';
        2 : stf := stf + 'sem ';
        3 : stf := stf + 'med ';
        4 : stf := stf + 'mod ';
        5 : stf := stf + 'std ';
      end;
    end
  end;

```



```
6 : stf := stf + 'vari ';
7 : stf := stf + 'skew ';
8 : stf := stf + 'sesk ';
9 : stf := stf + 'kur ';
10 : stf := stf + 'seku ';
11 : stf := stf + 'rang ';
12 : stf := stf + 'min ';
13 : stf := stf + 'max ';
14 : stf := stf + 'sum ';
15 : stf := stf + 'all ';
end;
end
else
  case ChoiSsub[Status_Ssub] of
    1 : stf := stf + 'mean ';
    2 : stf := stf + 'sem ';
    3 : stf := stf + 'med ';
    4 : stf := stf + 'mod ';
    5 : stf := stf + 'std ';
    6 : stf := stf + 'vari ';
    7 : stf := stf + 'skew ';
    8 : stf := stf + 'sesk ';
    9 : stf := stf + 'kur ';
    10 : stf := stf + 'seku ';
    11 : stf := stf + 'rang ';
    12 : stf := stf + 'min ';
    13 : stf := stf + 'max ';
    14 : stf := stf + 'sum ';
    15 : stf := stf + 'all ';
  end;
j := j+1;
```

end;

```
procedure Perform_optest(var sting : string);
```

```
var i : string;
```

```
begin
```

```
  if j = 1 then
```

```
    begin
```

```
      sting := '/opt ';
```

```
      case ChoiSsub[Status_Ssub] of
```

```
        1..3 : begin str(ChoiSsub[Status_Ssub],i);
```

```
                sting := sting + i + ' ';
```

```
            end;
```

```
      end; {case}
```

```
    end
```

```
  else
```

```
    case ChoiSsub[Status_Ssub] of
```

```
      1..3 : begin str(ChoiSsub[Status_Ssub],i);
```

```
                sting := sting + i + ' ';
```

```
          end;
```

```
    end; {case}
```

```
    j := j+1;
```

```
end;
```

```
procedure Perform_29(var str1 : string);
```

```
var st1 : string;
```

```
begin
```

```
  str1 := '/method = ';
```

```
  case ChoiSsub[Status_Ssub] of
```

```
    1 : str1 := str1 + 'stepwise';
```

```
    2 : str1 := str1 + 'forward';
```

```
    3 : str1 := str1 + 'backward';
```

```

    4 : str1 := str1 + 'enter';
    5 : begin st1 := 'rem ';
          close_win;
          Remo_var(st1);
          str1 := str1 + st1;
          SsubActive(1,Ssub_Menu);end;
    6 : begin st1 := 'test ';
          close_win;
          Remo_var(st1);
          str1 := str1 + st1;
          SsubActive(1,Ssub_Menu);end;
end;
end;

procedure Perform_LastSub1(var s:string);
begin
    case ChoiSsub[Status_Ssub] of
        1 : begin close_win;
              defix(s);
            end;
        2 : begin close_win;
              defree(s);
            end;
    end;
    SsubActive(1,Ssub_Menu);
end;

procedure Perform_LastSub2(var sting : string);
var str1,str2 : string;
begin
    case Current of

```

```
1 : begin Perform_21(str1); sting := sting+str1; end;
2 : begin case SubCur of
      2 : begin Perform_Des2(str1);
            st1 := st1+str1;
            end;
      3 : begin Perform_Des3(str2);
            st2 := st2+str2;
            end;
      end; {case}
      sting := st1+st2;
    end;
3 : begin case SubCur of
      3 : begin Perform_Mean3(str1);
            st1 := st1+str1;
            end;
      4 : begin Perform_Mean4(str2);
            st2 := st2+str2;
            end;
      end;
      sting := st1+st2;
    end;
4 : begin case SubCur of
      2 : begin Perform_Cros2(str1);
            st1 := st1+str1;
            end;
      3 : begin Perform_Cros3(str2);
            st2 := st2+str2;
            end;
      end;
      sting := st1 + st2;
    end;
```

```

5 : begin
  if SubCur = 1 then
    begin
      case ChoiSsub[Status_Ssub] of
        1 : begin close_win; Tind_va(str1);
              st1 := st1+str1;
              SsubActive(1,Ssub_menu);end;
        2 : begin close_win; T2gr(str2);
              st2 := st2+str2;
              SsubActive(1,Ssub_menu);end;
      end; {case}
      sting := st2+st1;
    end
  else if SubCur=3 then
    perform_optest(sting);
  end;
6 : begin case SubCur of
          5 : Perform_Oneway5(str1);
          6 : Perform_Oneway6(st1);
          7 : Perform_Oneway7(str2);
        end; {case}
      sting := str1+st1+str2;
    end;
7 : begin case SubCur of
          4 : begin Perform_Anov4(str1);
                st1 := st1 + str1;
              end;
          5 : begin Perform_Anov5(str2);
                st2 := st2 + str2;
              end;
        end; {case}

```

```

        sting := st1 + st2;
    end;
8 : begin case SubCur of
        2 : begin Perform_Corr2(str1);
            st1 := st1+str1;
            end;
        3 : begin Perform_Corr3(str2);
            st2 := st2+str2;
            end;
        end; {case}
        sting := st1+st2;
    end;
9 : Perform_29(sting);
10 : begin case SubCur of
        1 : begin str1 := 'hor = ';
            Perform_Plot1(str1); st1 := st1 + str1; end;
        2 : begin str2 := 'ver = ';
            Perform_Plot1(str2); st2 := st2 + str2; end;
        end;
        sting := st1+st2;
    end;
11 : begin case SubCur of
        13 : begin str1 := '';
            Perform_Npar1(str1); st1 := st1 + str1; end;
        14 : begin str2 := '';
            Perform_Npar2(str2); st2 := st2 + str2; end;
        end;
        sting := st1+st2;
    end;
end; {case}
end;
```

```

procedure DoSsub(var s:string);
begin
  case ShowStatus of
    1 : Perform_LastSub1(s);
    2 : Perform_LastSub2(s);
  end; { case } end;
procedure TestKsub(Ksub: char;var s:string);
begin
  if Ksub = Esc_Key then
    begin windowclose;Ok := true; end;
  if FuncKey then
    case (Ksub) of
      Up_Key      : MoveUpSsub;
      Dn_Key      : MoveDnSsub;
      PgUp_Key    : MoveToFirstSchoi;
      PgDn_Key    : MoveToLastSchoi;
    end { of Case }
  else
    if Ksub = Return_Key Then DoSsub(S);
  end;
procedure Init_Sub(SS_Menu : StrucSsub;
                  Status,SCurr,CurSub : byte;var s:string);
begin
  Ok := false;SubCur := CurSub;Current := SCurr;
  ShowStatus := Status;ChoiSsub[1] := 1;
  Ssub_Menu := SS_Menu;SsubActive(1,Ssub_Menu);
  st1 := ''; st2 := '';j := 1;
  repeat
    ReadFuncKey(Ksub);TestKsub(Ksub,s);until Ok;end;
begin
end.

```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\FREQ.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\DES.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\cross.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\t_test.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\MEAN.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\CORR.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\ONE.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\anova.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\regress.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\npar.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\pj\help\count.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\recode.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\if.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\sel.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\weight.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\hcom.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\proc.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\compute.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\defi.TXT');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\mdat.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```



```
Var ScrFile : File of ScreenArray;
```

```
Begin
```

```
    Assign(ScrFile, 'c:\sptool\help\uses.txt');
```

```
    Rewrite(ScrFile);
```

```
    write(ScrFile, ScreenData);
```

```
End.
```