

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/329355003>

للحوسبة الاحصائية R لغة البرمجة

Book · December 2018

CITATIONS

0

READS

19,821

1 author:



Nada Jarah

University of Basrah

8 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Simulation of autonomous mobile robot using fuzzy logic [View project](#)



اتجاهات المعلمين نحو استخدام الحاسوب كوسيلة تعليمية في المدارس العراقية [View project](#)

لغة البرمجة
الحوسبة الإحصائية



تأليف

الأستاذ المساعد ندى بدري جراح

ماجستير علوم حاسوبات

لغة البرمجة R للحوسبة الاحصائية

تأليف

الأستاذ المساعد ندى بدوى جراح

ماجستير علوم حاسوبات

لطلبة قسم الإحصاء / كلية الإداره والاقتصاد

الطبعة الاولى

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
نَعْمَلُ مَا نَعْلَمُ وَلَا نَعْلَمُ مَا نَفْعَلُ

(فَوَلَمْ يَأْتِكُمْ مِنْ أَنَّا أَنْذَرْنَاكُمْ
مَا سَرَّتُمُ الْأَعْيُونَ)

عَمَلَكُمْ وَرَأَسْوَاهُ
مَا تَرَكُوا وَمَا حَمَلُوا

صَدَقَ اللَّهُ عَلَيْهِ
مَا يَعْلَمُ

فَلَمْ يَرَهُمْ

شُوَّارٌ
لَا يَنْعَمُونَ (105)

الْقَرْبَاءُ حَمَاسِيَّةٌ

إلى سرح أخي الطاهرة فاتن ...

إلى فلبها الصافي وابتسامتها الاحلة ...

إلى التي لم يعد تجمعنا شيء سوى الذكريات

امنى لولحظة من اوقاتنا الجميلة

امنى لوابسامة من فرحتنا القديمة

امنى لوراثتِ وانا كما كنا



مقدمة

لقد نتج هذا الكتاب من الحاجة الى مرجع عربي لاحد اهم برامج الحوسبة الاحصائية وهو لغة البرمجة R ، حيث تتنسم المؤلفات العربية من البرامج الاحصائية بالندرة الشديدة، ونأمل ان يساهم هذا الكتاب في سد النقص ولو بشيء يسير ، كما وقد حاولنا اغناء موضوع الكتاب بأهم المصادر من كتب اجنبية وموقع الكترونية وخاصة الموقع الرسمي للغة البرمجة R .

فمع زيادة القراءة الحاسوبية تأتي زيادة فرص الحصول على كميات كبيرة من البيانات التي يمكن الحصول عليها بحرية ضمن برنامج لغة R اضافة الى حزم كبيرة من عبارات برمجية او برامج جاهزة لحل اغلب المشاكل الاحصائية ، هذه الحزم التي ساهم بها الكثير من الباحثين من مختلف ارجاء العالم . ولفهم البرنامج المجاني المشهور في مجال الاحصاء وعرض البيانات الذي يتيح لك السيطرة على عملية تحليل البيانات لا بد وان يكون لك دراية ولو بسيطة في استخدام الحاسوب .

علم الاحصاء هو احد العلوم التي تدرس الطرق العلمية لجمع البيانات وتنظيمها وتحليلها للحصول على نتائج موثوقة لدعم اتخاذ القرارات السليمة والصائبة لهذا التحليل .

كما ان اتساع عملية التعامل مع الإحصاء الرياضي وال الحاجة الملحة لإيجاد برامج خاصة تقوم بعملية الحسابات وإيجاد المقاييس وإجراء الاختبارات الإحصائية المهمة، دعت الحاجة المختصين للتعامل مع البرامج الإحصائية للحصول على النتائج، فقد أصبحت البرامج الإحصائية أكثر تداولاً بل تدرس في مختلف المعاهد والجامعات المنتشرة في تطبيقات الأعمال الهامة ، وهذا ما نلمسه واضحاً في تدريس برنامج معالج الجداول Excel والبرنامج الاحصائي SPSS وبرنامج Matlab و Minitab وبرنامج SAS وS .

R هو في العالم الأكثر استخداماً على نطاق واسع كلغة برمجة الإحصاءات. إنها الخيار رقم 1 من علماء البيانات وبدعم من مجتمع نابض بالحياة والموهوبين من المساهمين.

يتضمن هذا الكتاب تسعه فصول تتحدث باختصار عن الحوسبة الاحصائية واهم اللغات في هذا المجال وهي لغة البرمجة R ، ومفردات اللغة من ثوابت ومتغيرات وانواع البيانات وكيفية حفظها واستدعائها ، ثم كيفية الرسم البياني وكتابة العبارات البرمجية ، والتعرف على حزم البيانات والبرامج المضافة وبشكل حر عبر الانترنت واخيرا وليس اخرأ بعض التطبيقات الاحصائية في لغة R .

ان برنامج R يؤهل الطلاب ليصبحوا مختصين في احد برامج الحوسبة الاحصائية لمقابلة الحاجة في سوق العمل حاليا وفي المستقبل القريب .

وارجو المعذرة في التقصير بعدم سرد مختلف العمليات الاحصائية من امثلة تطبيقية وبرامج .. واكتفي بان يكون ما مطروح من مفردات في هذا الكتاب قد ساهم في ازالة الغموض ووضعك على اولى عتبات لغة البرمجة الحرة R والتي تسمح باطلاق متصفح الويب لتصفح الصفحات المساعدة من الارتباطات التشعبية لتدخل الى عالم واسع من التطبيقات والحزم والبيانات لجميع انحاء العالم .



المحتويات

الصفحة	الموضوع	ت
الفصل الاول		
الحوسبة الاحصائية		
9	الحوسبة الاحصائية	1.1
9	الحوسبة الاحصائية في لغة R	2.1
10	مجالات الاستخدام	3.1
11	هدف دراسة الحوسبة الاحصائية	4.1
12	تحليل البيانات والاحصاء	5.1
13	اسئلة الفصل الاول	
الفصل الثاني		
مقدمة لغة R		
17	تعريف لغة R	1.2
18	لماذا تُستخدم R	2.2
20	مميزات لغة R	3.2
21	حدود R	4.2
21	تنصيب لغة R	5.2
22	تشغيل برنامج R	6.2
22	واجهة برنامج R	7.2
26	اوامر الخاصة في لغة R البرمجة	8.2
28	انهاء R	9.2
29	الحصول على المساعدة	10.2
30	اياعات عامة	11.2
31	The فضاء العمل Workspace	12.2

32	الاوامر الاساسية في R	13.2
36	اسئلة الفصل الثاني	
الفصل الثالث		
المتغيرات و الثوابت		
41	مقدمة	1.3
41	الكائن	2.3
42	المتغيرات في R	3.3
42	الثوابت في R	4.3
43	ثوابت اخرى	5.3
45	عمليات المقارنة	6.3
46	العمليات المنطقية	7.3
47	اسبقة التشغيل	8.3
48	اسئلة الفصل الثالث	
الفصل الرابع		
البيانات في R		
53	مقدمة	1.4
53	Vectors	2.4
60	القوائم Lists	3.4
61	المصفوفات والمجموعات Matrices and Arrays	4.4
69	الجداريات Tables	5.4
70	اطر البيانات Data frames	6.4
74	العوامل Factor	7.4
77	دوال للحصول على معلومات عن مجموعة البيانات	8.4
79	اسئلة الفصل الرابع	

الفصل الخامس

حفظ واستدعاء البيانات

85		مقدمة	1.5
86	حفظ واستدعاء كائن		2.5
86	استيراد البيانات من القرص المحلي		3.5
86	استخدام البيانات من R		4.5
89	قراءة البيانات من ملف تنسيقه csv		5.5
90	قراءة البيانات من الملفات النصية txt.		6.5
92	استيراد ملفات Excel إلى R		7.5
93	استيراد البيانات من الانترنت		8.5
93	استيراد ملفات البيانات باستخدام وظيفة المسح الضوئي		9.5
94	تصدير البيانات Exporting data		10.5
95	اسئلة الفصل الخامس		

الفصل السادس

الرسم البياني في R

99		مقدمة	1.6
99	الدالة hist()		2.6
102	الدالة plot(...)		3.6
104	Subplots		4.6
106	دالة المنحني curve		5.6
106	barplot()		6.6
108	Dotchart()		7.6
109	Box plots()		8.6

110	الرسوم المزدوجة pairs()	9.6
111	الدالة layout()	10.6
111	التخطيطات الدائرية Pie charts	11.6
112	رسم الكثافة	12.6
113	رسم qq	13.6
114	الدالة Strip Chart	14.6
115	رسم ثلاثي الابعاد 3D plot	15.6
118	اسئلة الفصل السادس	
الفصل السابع البرمجة في R		
123	مقدمة	1.7
123	عبارات البرمجة في R	2.7
123	عبارات if	3.7
123	عبارة if statement	1.3.7
124	عبارة if...else	2.3.7
127	عبارة if...else المتداخلة	3.3.7
128	الحلقات Loops	4.7
128	عبارة for	1.4.7
131	عبارة while	2.4.7
134	عبارة break	3.4.7
135	عبارة Repeat	4.4.7
136	عبارة next	5.4.7
137	عبارة switch	5.7
138	الدوال functions	6.7
147	دوال اخرى	7.7
149	RStudio	8.7
153	الفرق بين R و RStudio	9.7
154	اسئلة الفصل السابع	

الفصل الثامن الحزم في R

157		مقدمة	1.8
157		كيفية استخدام الحزم	2.8
157		تعريف الحزم في R	3.8
158		تحميل حزم البيانات	4.8
158		تحميل حزم البرامج Packages	5.8
160		قائمة Load package...	6.8
161		Set CRAN mirror...	2.6.8
161		Select repositories...	3.6.8
162		Install package(s)...	4.6.8
162		Update packages...	5.6.8
163		Install package(s) from local zip files...	6.6.8
163		اضافة حزم لبرنامج R على حاسوبك	7.8
165		مستودعات لحزم اضافية	8.8
166		اسئلة الفصل الثامن	

الفصل التاسع تطبيقات احصائية

169		مقدمة	1.9
169		المتوسط Mean والوسيط Median	2.9
171		الانحدار الخطى Linear Regression	3.9
173		دالة التنبؤ predict	4.9
174		الانحدار المتعدد Multiple Regression	5.9
175		التوزيع الطبيعي	6.9

	Normal Distribution
176	dnorm() دالة 1.6.9
176	pnorm() دالة 2.6.9
177	qnorm() دالة 3.6.9
177	rnorm() دالة 4.6.9
178	التوزيع الثنائي (ذي الحدين) 7.9
	Binomial Distribution
178	dbinom() دالة 1.7.9
179	pbinom() دالة 2.7.9
179	qbinom() دالة 3.7.9
180	rbinom() دالة 4.7.9
180	انحدار بواسون 8.9
	Regression
181	Covariance التغير 9.9
182	Correlation الارتباط 10.9
183	تحليل التباين الاحادي 11.9
	one-way Analysis of Variance ANOVA
184	t test t اختبار t 12.9
186	f distribution f توزيع f 13.9
188	اسئلة الفصل التاسع
191	فهرست الاشكال



الفصل الاول

الحوسبة الاحصائية

1.1 الحوسبة الإحصائية

2.1 الحوسبة الاحصائية في لغة البرمجة R

3.1 مجالات الاستخدام

4.1 هدف دراسة الحوسبة الاحصائية

5.1 تحليل البيانات والاحصاء

اسئلة الفصل الاول



1.1 الحوسبة الإحصائية Statistical Computing

تمثل الاستخدام الواسع من الطرق والأساليب والتي تشمل مختلف مهارات علم الإحصاء وعلم الحاسوبات لحل المشاكل وتحليل الظواهر الطبيعية أو الاجتماعية. كما تؤكد على تأثير الحوسبة على الإحصاءات والعكس بالعكس. والعمل في مجموعة متنوعة من المجالات في الإحصاءات، بما في ذلك القياسات الحيوية، الاقتصاد القياسي، وتحليل البيانات، والرسومات، والمحاكاة، والخوارزميات، والنظام القائم على المعرفة، والحوسبة النظرية الافتراضية. إضافة إلى الأساليب الإحصائية المكثفة حسابيا بما في ذلك اختزال طرق، سلسلة ماركوف مونتي كارلو الأساليب والانحدار المحلي ، نواة تقدير الكثافة ، الشبكات العصبية الاصطناعية . ويوجد الكثير من البرامج الإحصائية التي قد تتميز الواحدة عن الأخرى بمعالجة بسيطة أو قد لا تختلف إلا في تسميتها .. ومن البرامج المتخصصة بالحوسبة الإحصائية هي :

Matlab , Minitab , Excel , SPSS , S , R

ويوجد أيضاً مصطلح "الإحصاءات الحاسوبية' computational statistics' الذي يهدف إلى تصميم خوارزمية لتنفيذ الأساليب الإحصائية على أجهزة الكمبيوتر. في الوقت الذي تمثل الحوسبة الإحصائية تطبيق علوم الحاسوب الآلي للاحصاءات.

2.1 الحوسبة الإحصائية في لغة البرمجة R

R هو أقوى لغة برمجة في العالم للحوسبة الإحصائية، والتعلم الآلي والرسومات وكذلك مجتمع عالمي مزدهر من المستخدمين والمطوريين والمساهمين. فقد تطورت R بسرعة لتصبح لغة البرمجة الرئيسية في مجال الإحصاءات والبيانات العلمية وان عدد مستخدمي R ينمو بنسبة 40% سنوياً .

وتعتبر R لهجة من اللغة الإحصائية S وتطبيق حديث لها ، حيث S لغة برمجة احصائية متقدمة واحدى لغات الحوسبة الإحصائية .

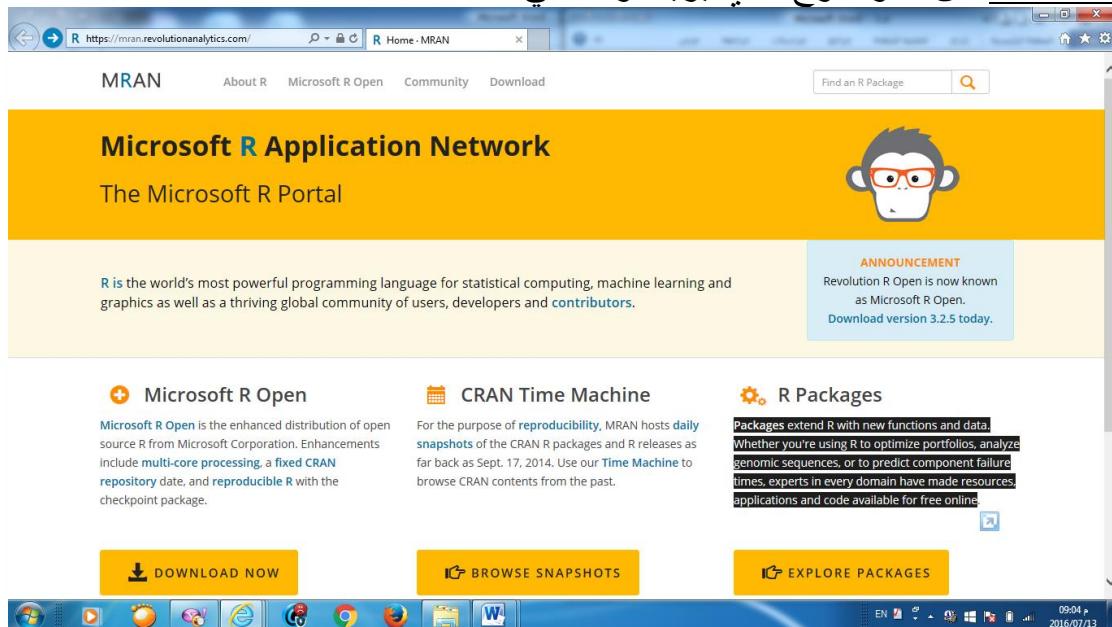
الحوسبة الإحصائية في R هو لتوبيخ كيفية استخدام ما هو متاح بحرية في حزمة البرامج R لتحليل البيانات والبرمجة الإحصائية، والرسومات. حيث وجود الكود والأمثلة من جميع أنحاء النص يتطلب فقط معرفة أساسية من الإحصاءات والحوسبة.

ويشمل هذا تقديم تحليل وتوزيع التشخيص لعينة واحدة، الانحدار، مشاكل عينتين

اثنين ومقارنة التوزيعات، وتحليل متعدد المتغيرات. ويستخدم مجموعة من الأمثلة لتوضيح كيف يمكن استخدام R لمعالجة المشاكل الإحصائية.

وبسبب كون R مفتوحة المصدر، تم اعتمادها من قبل أقسام الإحصاء في جامعات من مختلف أنحاء العالم، فالطبيعة التوسعية ومجانية اللغة جعلتها تلعب دوراً مهماً في البحوث الأكademية، وخلال فترة قصيرة جداً الباحثون الإحصائيون وعلماء البيانات والتعلم الآلي بنشر الأبحاث العلمية المحتوية على التعليمات البرمجية لـ R لتنفيذ مهام العمل الجديدة بسهولة . كما يمكن لأي شخص أن ينشر حزمة عمل في "شبكة الأرشيف الكامل لـ R" المسماة اختصاراً بـ CRAN، وتصبح متاحةً للجميع.

حتى كتابة هذه السطور، ساهمآلاف من مستخدمو لغة R بأكثر من 6100 حزمة عمل، موسعين قدرات اللغة إلى مجالات متعددة كالاقتصاد وتحليل التجارب السريرية والعلوم الاجتماعية وبيانات الويب. ويمكن لأي شخص أن يقوم بالبحث عن التطبيقات في (MRAN/ Microsoft R Application Network) عن الموضوع الذي يريد وكما في الشكل أدناه .



شكل (1-1)

3.1 مجالات الاستخدام

في مجالات الحياة اليومية نبحث دائماً عن تقدير الأشياء والمتمثلة بالأرقام والاحصائيات لتأكيد صحتها واتخاذ القرارات المناسبة لها.



كما ان استخدام الحاسوب كوسيلة متقدمة لادخال البيانات واجراء الحسابات الاحصائية ..
في الشكل ادناه بعض الامثلة لمجالات عمل الحوسبة الاحصائية



شكل (2-1) مجالات عمل الحوسبة الاحصائية

- (1) التلوث الصناعي للتربة والهواء والماء ودراسة تأثيرها على صحة الانسان بطرق احصائية
- (2) دراسة وتحليل الجينات ومقارنتها وتصنيفها للسيطرة مستقبلا على امراض مسععصية مثل السرطان .
- (3) العمل في شركات العاقير والادوية بالاعتماد على الدراسات الاحصائية لتأثير الدواء وفعاليته.
- (4) اعتماد الدراسة في الاقتصاد والاعمال وتوقعات السوق المالية على التحليل الاحصائي لمؤشرات العرض والطلب والاستهلاك .
- (5) تطبيقات الشبكات والانترنت تأخذ حيزاً كبيراً في مناهج الحوسبة الاحصائية.

4.1 هدف دراسة الحوسبة الاحصائية

يشير مصطلح الحوسبة الاحصائية الى فضاء رحب من الطرق والاساليب التي تشتراك فيها مهارات متعددة من علم الاحصاء وعلم الحواسيب والتعلم الالي لحل المشاكل وتحليل الظواهر الطبيعية والاجتماعية والاقتصادية كما وتساهم الحوسبة

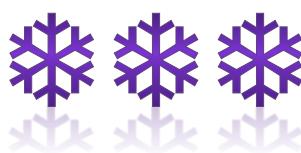
الاحصائية في تلبية حاجة سوق العمل والذي يعتمد على استخدام القياسات الاحصائية وتمثل هذه الدراسة الالمام في المجالات التالية :

- (1) علم الحاسوب والاحصاء
- (2) لغات البرمجة المهيكلة
- (3) نظم المعلومات وتطبيقات النماذج الاحصائية
- (4) تصميم وتحليل الدراسات الاحصائية

5.1 تحليل البيانات والاحصاء

ان تحليل البيانات ومراحله المختلفة يتطلب استعمال ادوات او لغات برمجة ، ومن اللغات الاكثر استعمالا في هذا المجال هي لغة R فهو بيئه البرمجيات الحرة للحوسبة الإحصائية والرسوم البيانية ، بالإضافة إلى ان R تتنفيذ أساسا كل التحليلات الإحصائية القياسية ، حيث يمكن القول بان R لغة مرنة لا مثيل لها من قبل البرامج الاحصائية الاخرى وكما في لغات البرمجة الشيئية تسمح لخلق الوظائف التي تؤدي اجراءات مخصصة وامتناة المهام التي يتم تنفيذها عادةً .

في R تتم التحليلات الإحصائية عادة على شكل سلسلة من الخطوات ، وان النتائج المتوسطة يجري تخزينها في الاشياء ويسمح بسهولة باستخدام نتائج تحليل واحد كمدخل لتحليل آخر. وعلاوة على ذلك R تحتوي على كائنات جميع المعلومات لنموذج ذات الصلة، حزم R لابتكارات جديدة في الحوسبة الإحصائية حيث يميلون أيضاً لتصبح متاحة بسرعة أكثر مما تفعل هذه التطورات في غيرها من حزم البرامج الإحصائية.



اسئلة الفصل الاول

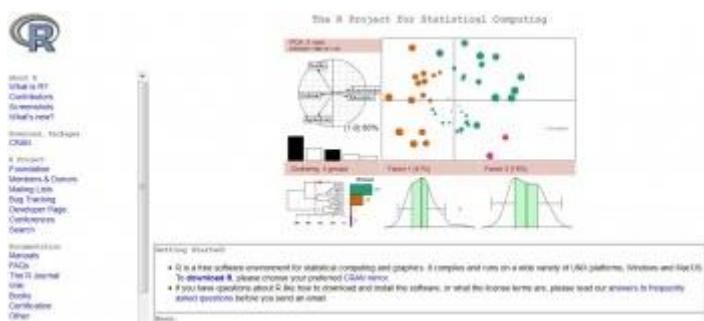
س1: ما هي الحوسبة الاحصائية ؟ وما هي البرامج المتخصصة فيها ؟ وما الفرق عن الاحصاءات الحسابية ؟

س2: تكلم عن الحوسبة الاحصائية في لغة البرمجة R ؟

س3: اذكر امثلة لمجالات عمل الحوسبة الاحصائية ؟

س4: ما الهدف من دراسة الحوسبة الاحصائية ؟

س5: ما اللغة التي تستخدم لتحليل البيانات والاحصاء ؟ ووضح لماذا ؟



الفصل الثاني مقدمة لغة R



- 1.2 تعريف لغة R
- 2.2 لماذا تُستخدم R
- 3.2 مميزات لغة R
- 4.2 حدود R
- 5.2 تنصيب لغة R
- 6.2 تشغيل برنامج R
- 7.2 واجهة برنامج R
- 8.2 الاوامر الخاصة في لغة البرمجة R
- 9.2 انهاء R
- 10.2 الحصول على المساعدة
- 11.2 ایعازات عامة
- 12.2 فضاء العمل The Workspace
- 13.2 الاوامر الاساسية في R

اسئلة الفصل الثاني

1.2 تعريف لغة R

R من البرمجيات الحرة والمصممة للحوسبة الإحصائية المفتوحة المصدر. تُستخدم R بشكل واسع بين الإحصائيين لتطوير البرامج الإحصائية وتحليل وتمثيل البيانات والقيام بالعمليات الحسابية والرسوم البيانية .

وهي متوفرة ومتحدة على شبكة الانترنت من مصادر وكتب الكترونية وبرامج جاهزة تُنفذ مهمة ما كما وتحتوي R على مجموعة حزم مفتوحة المصدر مطورة من طرف اناس ومجموعات ابحاث . وان الاصدار الاول للغة R في تسعينات القرن الماضي وتحديدا عام 1993 ، حيث صُممت هذه اللغة من قبل روس إلهاكا ، وروبرت جينتلمان خبيري الاحصاء بجامعة اوكلاند في نيوزيلندا وقد سُميت بلغة R لأنها ترمز إلى الحرف الاول من اسميهما ، وقد تطور مشروع هذه اللغة ليشمل اكثر من عشرين بين احصائي وعالم كمبيوتر من جميع أنحاء العالم.

استطاعت لغة R على الشبكة العنكبوتية أن تجذب في وقت قصير اهتمام العلماء من جميع أنحاء العالم ، الذين كانوا في حاجة إلى برامج إحصائية ، وعلى استعداد للمشاركة بأفكارهم .

وللغة R ارتباطات بالكثير من لغات البرمجة مثل لغة S والتي تم تصميمها في عام 1980 وقد استخدمت بشكل واسع النطاق في المجتمع الاحصائي .

R هي لغة وبيئة تطوير (Language and development environment) متخصصة في تحليل وتمثيل البيانات والحوسبة الإحصائية . في السنوات العشر الأخيرة، تجاوزت R الدوائر الأكاديمية إلى قطاعات تكنولوجية متعددة حيث تقوم باستعمالها شركات كبرى بما في ذلك Google و Microsoft موقع Facebook (لتحليل البيانات الشخصية . ثم إن شركات أدوية عالمية تعتمد لتحليل البيانات السريرية).

وتعتبر R لغة ديناميكية وسريعة التغير ، لذلك من المهم مواكبة احدث الادوات والتقنيات .

ويتميز الـ R بكونه يتمتع بصفات لغات البرمجة الأخرى حيث يمكن القيام بالبرمجة الكائنية وحتى انشاء صفحات ويب .

ان إحدى نقاط قوة لغة R هي سهولة توسيعها بالاستفادة من مجموعة الإضافات الهائلة المتاحة لها، والتي قام بتطويرها الآلاف من الجامعات والمراکز العلمية وحتى الباحثون المستقلون وطلاب الدراسات العليا، يعينهم في ذلك السهولة

النسبة في آلية بناء مثل تلك المكتبات أو الإضافات الجديدة لهذه اللغة، للاطلاع على لائحة الإضافات الرسمية المنشورة على موقع لغة R من الرابط (وهو مخدماً موجوداً في روسيا الاتحادية لضمان الوصول إليه بسهولة دون أي حجب!):
http://cran.gis-lab.info/web/packages/available_packages_by_name.html
ولتتعرف بالآلية بناء مكتبه أو إضافته الخاصة فعلى المرجع:
<http://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf>
من الإضافات المميزة الإضافية `qqplot2` التي تفتح آفاقاً واسعة لتحسين وإثراء أنواع المخططات البيانية التي يمكن توليدها والحصول عليها.

2.2 لماذا تُستخدم R

R هو برمجيات حرة وهو مشروع جنو الرسمي وتوزيعها وفقاً لمؤسسة البرمجيات الحرة الترخيص العام (GPL/General Public License) كما يعد R مصدراً قوياً لحزمة تحليل البيانات مع العديد من المعايير والدلالات الإحصائية المتطرفة. فهو يستخدم على نطاق واسع في العلوم السياسية، والاحصاءات والاقتصاد القياسي، والعلوم الاكتواريه (علم تخمين المخاطر)، وعلم الاجتماع، والتمويل، الخ
يتميز R بكونه يتمتع بصفات لغات البرمجة الأخرى حيث يمكن القيام بالبرمجة الكائنية وحتى انشاء صفحات ويب. لكن ما يميزه هو انه يحتوي على العديد من الحزم الاحصائية مما جذب إليه العديد من الاحصائيين.
تلقي لغة R اهتماماً كبيراً في مجال المعلوماتية الحيوية لعدة أسباب، من بينها سهولة الاستعمال اذا يمكن لباحث ذو خلفية زهيدة في البرمجة أن يتعلم استعمالها على الأقل في مستوى يلبي حاجياته الأساسية في تحليل البيانات ورسم المخططات.
وسبب آخر هو توفر عدد كبير من حزم تحليل البيانات البيولوجية حيث أن أغلبية الباحثين يقومون بنشر خوازمياتهم بلغة R إلى درجة أنه تم انشاء مستودع حزم [bioconductor](#) موقع يحتوي على الحزم البرمجية) خاص فقط بالحزم ذات العلاقة بمجال البرمجة الحيوية، بالإضافة إلى امكانية اظهار رسومات ذات شكل جذاب بسهولة.

R هي لغة البرمجة، لذلك يمكن بسهولة أن تمتد قدراته من خلال استخدام الدالات المعرفة من قبل المستخدم. ومجموعة كبيرة من الوظائف والحزام المتوفرة مع عائلة (كرا نرجمة CRAN) من موقع الانترنت عبر (<http://CRAN.R-project.org>). وللحصول على فكرة عن ما يمكنك القيام به في R ننظر الى شبكة الارشيف الشامل المسماة (CRAN) وهي كما يلي :

الاستدلال والنظرية الافتراضية	النظرية الافتراضية
الفيزياء الحاسوبية Chemometrics	ChemPhys
السريرية تصميم الابتدائية، رصد، وتحليل	التجارب السريرية
المعادلات التفاضلية	المعادلات التفاضلية
التوزيعات الاحتمالية	التوزيعات
الاقتصاد القياسي	الاقتصاد القياسي
تحليل البيانات الايكولوجية والبيئية	Environmetrics
تصميم التجارب (وزارة الطاقة) وتحليل البيانات التجريبية	تصميم تجاري
المالية التجريبية	تمويل
علم الوراثة الإحصائية	علم الوراثة
عرض الرسوم البيانية والحيوي الرسومات والجرافيك أجهزة والتصور	الرسومات
R عالية الأداء والحوسبة المتوازية مع آلية التعلم والتعلم الإحصائي	الحوسبة عالية الأداء
تحليل الصور الطبية	MedicalImaging
متعدد المتغيرات الاحصائيات	متعدد المتغيرات
معالجة اللغات الطبيعية	NaturalLanguageProcessing
الرياضيات العددية	NumericalMathematics
الاحصائيات الرسمية ومنهجية المسح	OfficialStatistics

تحليل بيانات حركية الدواء	<u>الدوائية</u>
علم الوراثة العرقي، طرق المقارنة خاصة	<u>علم الوراثة العرقي</u>
طرق نماذج النفسية	<u>القياس النفسي</u>
إحصاءات للعلوم الاجتماعية	<u>العلوم الاجتماعية</u>
تحليل البيانات المكانية	<u>مكانى</u>
المناولة وتحليل البيانات المكانية والزمانية	<u>الزمانية المكانية</u>
تحليل البقاء على قيد الحياة	<u>نجاة</u>
تحليل السلسل الزمنية	<u>متسلسلة زمنية</u>
تقنيات الويب والخدمات	<u>Web Technologies</u>

لتثبت هذه الآراء تلقائيا، تحتاج حزمة CTV ليتم تثبيتها، والتي سيتم توضيحها في فصول أخرى.

3.2 مميزات لغة R

- تتميز لغة R ببساطتها اذ أنها لغة سكريبت Scripting Language فلا تحتاج لكتابة دالة دخول كما هو في لغات البرمجة الكلاسيكية.
- حيث يُعتبر اول سطر هو اول امر ، وفيما يلي مميزات اخرى للغة البرمجة R :
- 1- متوفرة في موقع مجاني ومفتوح المصدر لغرض تنصيبها على حاسوبك ، أي متاحة للجميع
 - 2- تعمل مع عدة أنظمة تشغيل مثل نظام لينوكس ويونكس وماك وويندوز .
 - 3- لغة احصائية تعمل على حل جميع المسائل الاحصائية وباسلوب سهل مثلا استعمل الامر sum لحساب المجموع والامر var لحساب التباين والامر mean لحساب المعدلوهكذا
 - 4- انه يحتوي على إجراءات إحصائية متقدمة غير متوفرة في مجموعات أخرى.
 - 5- له قدرات رسومية عالية للمخططات ثلاثية الابعاد فضلاً على امكانية اظهار رسومات جذابة بسهولة
 - 6- R حساسة لحالة الاحرف
 - 7- لها قابلية لبرمجة الحسابات المتوازية parallel computing حيث يمكن حل عمليات الحوسبة المعقدة والنظم الاحيائية

- 8- هي لغة مفسرة Interpreted Language اي التنفيذ مباشرة بدون ترجمة وهذا ما نجده واضحًا في تغيير التعامل مع انواع البيانات اثناء التنفيذ
- 9- تحتوي على العديد من الحزم الاحصائية والتطبيقات الادارية والمالية والاقتصادية والعلمية والتكنولوجية ... الخ مما جذب إليه العديد من الباحثين والاحصائيين
- 10- تمتلك واجهة لغة HTML مما يساعد على تنفيذ الأوامر البرمجية بسهولة.
- 11- تعطي النتائج التي تحتاجها وبشكل مبسط دون تكديس المخرجات خلافاً لبرمجيات احصائية أخرى مثل برنامج SPSS .

4.2 حدود R

لكل لغة برمجة حدود، فمن الواضح أنك لن تستطيع مثلاً عمل لعبة ثلاثية الأبعاد بواسطة R كما أنك لن تستطيع حل كل المشاكل الرياضية وتحليل جميع أنواع البيانات.

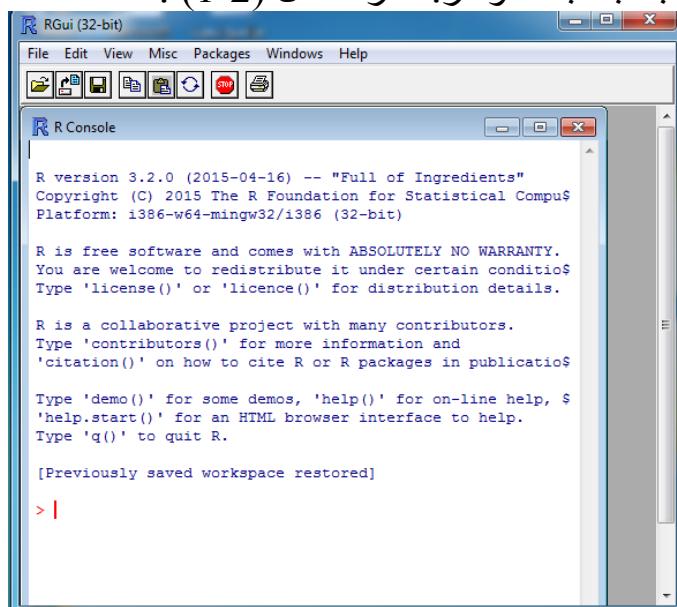
- لتحليل الأطيف والإشارات (Signal processing) ومحاكاة النظم الإلكترونية سيكون من الأفضل استعمال لغات الحوسبة العددية مثل متالاب.
- بما أنها لغة مفسرة، ستكون R أبطأ في بعض البرامج الضخمة وذات البنية معقدة في هذه الحالة، إذا كانت السرعة أمراً حيوياً، يتوجه الكثيرون نحو لغات مصرفية مثل "SAS" أو فورتران "Compiled"
- قد تكون ذاكرة R غير مهيأة لتعامل مع بيانات ذات سعة فائقة لأنها تقوم بتحميل كافة البيانات والمكتبات في الذاكرة الحية) والحلّ المتوفر هو إنشاء قاعدة بيانات خارجية وربطها بها).

5.2 تنصيب لغة R

تستطيع أن تقوم بتحميل لغة R من الموقع الرسمي لها على الرابط <http://www.r-project.org> ، وعملية تنصيب هذه اللغة تخلو من التعقيدات وبنتها يمكن تشغيل سطر الأوامر الخاص بها بغير أيقونة اللغة على سطح المكتب، وكل ما نكتبه تالياً يكون داخل سطر الأوامر هذا.

6.2 تشغيل برنامج R

- (1) النقر المزدوج على أيقونة R على سطح المكتب
- (2) او من start ← All Programs ← وحينها ستظهر الشاشة الافتتاحية بسطر الاوامر الخاصة بلغة R حيث تظهر علامة التنصيص (>) للبدأ بكتابة الاوامر. انظر الشكل (1-2) :

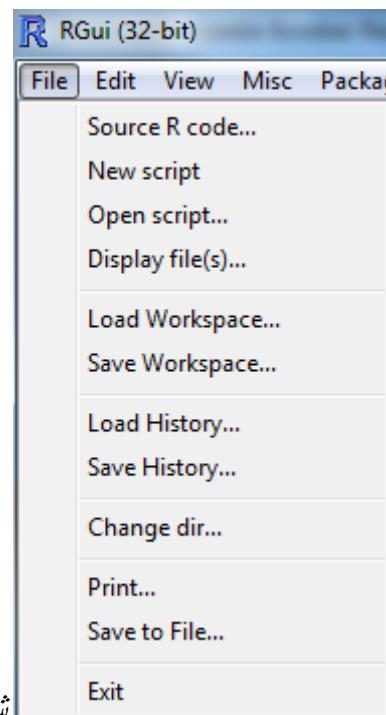


شكل (1-2) الشاشة الافتتاحية في لغة R

7.2 واجهة برنامج R

1.7.2 شريط القوائم

يظهر فيه اسماء القوائم للبرنامج وفيما يلي الاوامر في كل قائمة :
 (1) File وتحتوي الاوامر التالية كما في الشكل (2-2):



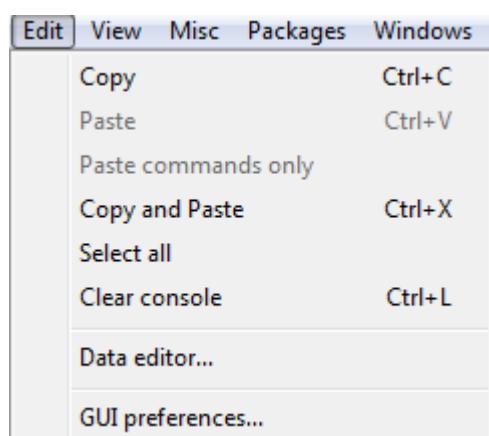
شكل (2-2) قائمة File

- New script : لفتح مكان في البرنامج R لكتابة عبارات برنامج فقط دون تنفيذ :

- من قائمة File نختار New script

- او نضرب المفاتيح ctrl+shift+N معاً

تظهر شاشة " Untitled - R Editor " ، ثم يمكن نقل تلك العبارات الى فضاء العمل



شكل (3-2) قائمة Edit

. بايعازى النسخ واللصق او ctrl+r

(2) Edit : وتحتوي الاوامر التالية :

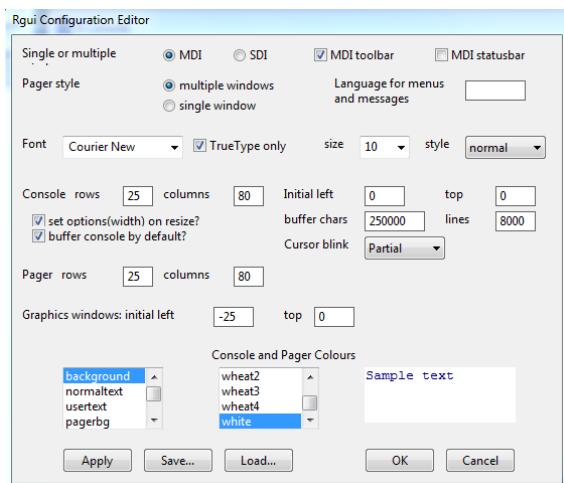
GUI preferences... : لتعديل مواصفات

شاشة العرض في R، من القائمة Edit في

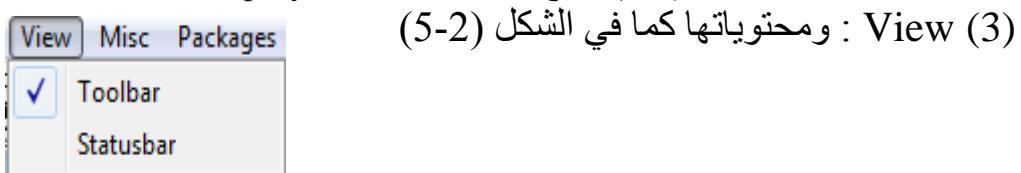
الشكل (3-2) نختار ... GUI preferences...

يظهر مربع حوار " Rgui Configuration "

: (4-2) وكما في الشكل (4-2) "Editor

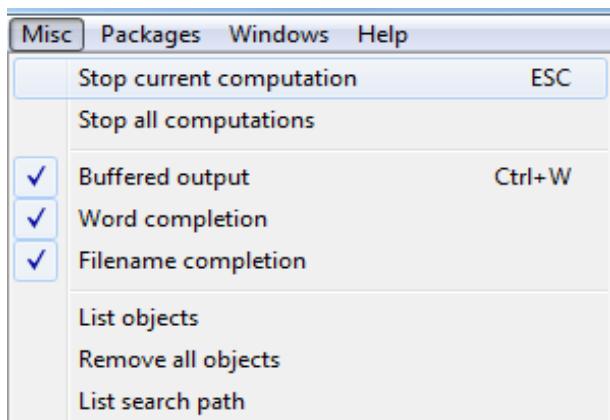


شكل (4-2) مربع حوار *GUI preferences...*



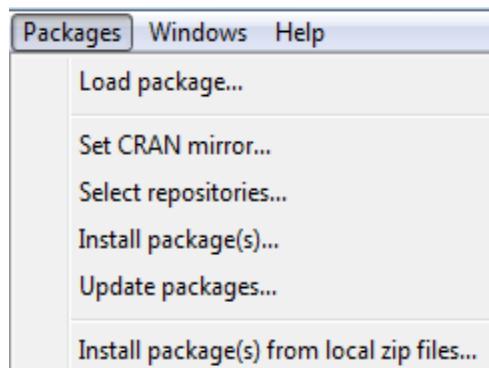
شكل (5-2) قائمة *View*

: وتحتوي على عدة اوامر كما في الشكل (6-2)

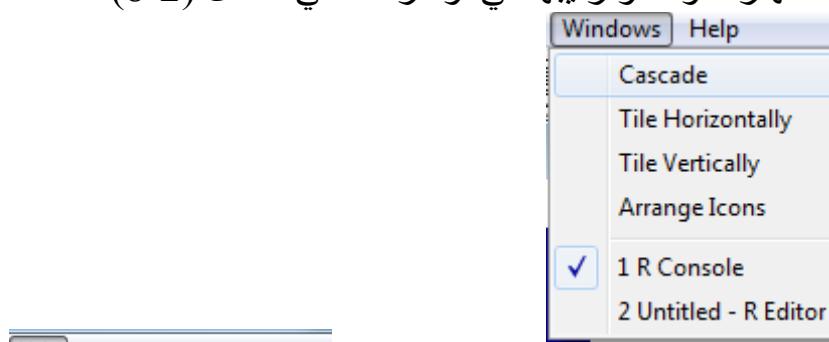


شكل (6-2) قائمة *Misc*

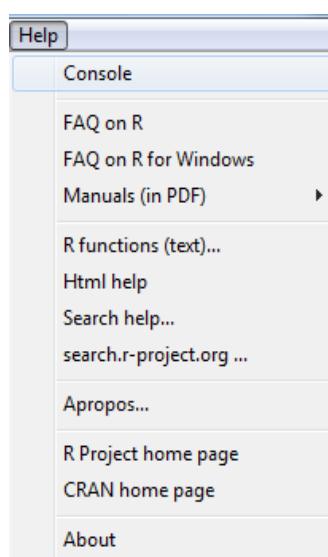
: وتحتوي على اوامر لاستدعاء الحزم وكما في الشكل (7-2)



شكل (7-2) قائمة Packages
لاظهار النوافذ وترتيبها في اوامر كما في الشكل (6) (8-2)



شكل (8-2) قائمة Windows Help (7)
للحصول على التعليمات لبرنامج R
وضمن اوامر عده كما في شكل (9-2)



شكل (9-2) قائمة Help (9-2)
لتحقيق سهولة الحصول على المساعدة لاوامر او ايعازات في برنامج R



2.7.2 شريط الاذوات Toolbar

ويتكون من الايقونات المهمة والاكثر استخدام وهي :

(1) Open script وتعني فتح الملف في برنامج R

(2) Load workspace تحميل فضاء العمل

(3) Save workspace حفظ فضاء العمل

(4) Copy نسخ

(5) Paste لصق

(6) Copy and paste نسخ ولصق

(7) Stop current computation وقف الحساب الحالي

(8) Print طباعة

R version 3.2.0 (2015-04-16)

3.7.2 شريط الحالة statusbar

ويظهر اسفل الشاشة ويُكتب فيه نوع الاصدار لـ R والسنة

8.2 الاوامر الخاصة في لغة البرمجة R

لكل لغة من لغات البرمجة عدد من الرموز الخاصة التي تميزها عن اللغات الأخرى ، وتعتبر قواعد هامة يجب الالتزام باستخدامها الصحيح وفيما يلي اهم واكثر الاوامر الخاصة في لغة البرمجة R :

(1) الرمز (->) : يشار الى عملية الإسناد في لغة R ، وهي الطريقة الأكثر شيوعا مقارنة برمز المساواة = والذي يصح استخدامه ايضاً

(2) الفارزة المنقوطة (;) تستخدم لفصل الاوامر في السطر الواحد ، او يكتب الامر التالي في سطر جديد

(3) العبارة التعريفية : كل شيء بعد الرمز # (هاش) الى نهاية السطر يعتبر عبارة تعريفية او تعليق للمستخدم ويتم تجاهله من قبل لغة R . لكن الغريب أن لغة R تفتقر إلى طريقة لجعل مقطع كامل يعامل معاملة التعليقات (كما هو حال استخدام أسلوب التأطير / * ... */ في العديد من لغات البرمجة الأخرى).

(4) يمكن ان يكون الامر او العبارة في لغة R على عدة اسطر

(5) يمكن كتابة الاوامر على الدوال داخل اقواس مستديرة ، على سبيل المثال :

> (2^(1000)) rnorm()

(6) تُطبع قبل النتيجة العلامة "[1]" وهذا لأن R يعتبر افتراضيا كل شيء بمثابة جدول "vector" والرقم واحد هو مؤشر عن العنصر الأول في الجدول.

e.x.

```
> 1:70
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
[51] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
```

(7) ان ناتج تنفيذ أي أمر أو دالة بلغة R يظهر بعدها مباشرة، وهكذا تكون جلسة العمل الاعتيادية من تنفيذ لنتالي من الأوامر والتعليمات وصولا إلى إنجاز العمل أو التحليل المطلوب.

(8) يوفر R آلية لإعادة تنفيذ الأوامر السابقة وذلك باستخدام مفاتيح الأسهم على لوحة المفاتيح حيث يمكن أن تستخدم للتنقل إلى الأمام والخلف من خلال الأوامر السابقة الاستخدام ، كما يمكن للمؤشر ان ينتقل ضمن الأوامر باستخدام مفاتيح الأسهم الأفقية، والحرف يمكن ازالتها مع مفتاح DEL او اضافتها من مفاتيح أخرى . يمكنك باستخدام أزرار الأسهم إلى الأعلى وإلى الأسفل للتنقل عبر مجموعة الأوامر التي تم تنفيذها .

(9) دالة الربط والتي لها البناء اللغوي التالي: (object1,...,objectn) ، تُستخدم بشكل كبير لوضع قيم ذات نوع واحد في شكل متوجه

e.x: >x<-c(1,2,3,"a")

> x

(للتنفيذ او الانتقال الى سطر جديد نضرب مفتاح الادخال (Enter)

[1] "1" "2" "3" "a"

(10) R حساسة لحالة الاحرف فـ a و A نوعان من كائنات مختلفة وعلى سبيل المثال :

```
> x=1:50
> MEAN(x)
Error: could not find function "MEAN"
> mean(x)
[1] 25.5
```

لكن ممكن تكون الحالة صحيحة في حالة :

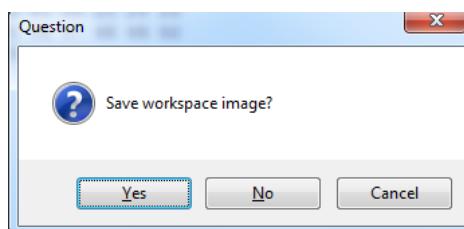
```
> MEAN <- mean
> MEAN(x)
[1] 25.5
```

(11) تعطي R العلامة + في بداية السطر للدلالة على أن المدخل حتى الآن لم يكتمل تعريفه ، كما في المثال التالي:

```
> for (x in 1:3){
+print(x)
+}
```

(12) تقليديا يؤكد " " لا تستخدم في الأسماء. غالبا ما يكون من الأفضل استخدام النقاط ". ". ينبغي للمرء تجنب استخدام تسطير كما الحرف الأول من اسم كائن وكما في المثال التالي :

```
> .a=9
> .a
[1] 9
> -a=9
Error in -a = 9 : could not find function "-<-"
```



R 9.2 انهاء

لانهاء برنامج R نطبع الامر >q() يظهر مربع حوار كما في الشكل(10-2) والذي يطلب منك ما اذا كنت تريد حفظ البيانات من جلسة R الخاصة بك.

شكل (10-2) مربع حوار اما في حالة كتابة q بدون اقواس يظهر خطأ وكما يلي :

```
> q
function (save = "default", status = 0, runLast = TRUE)
.Internal(quit(save, status, runLast))
<bytecode: 0x06960ebc>
<environment: namespace:base>
```

10.2 الحصول على المساعدة :

في عدة حالات تحتاج الى توضيح لامر او ايعاز ما في برنامج R، وهذا ما نجده في الطرق التالية :

- 1) في الغالب توفر R مساعدة في تنسيق HTML للحصول على التعليمات العامة والشاملة لبرنامج R عن طريق تشغيل : `<help.start()>`لينقلني الى موقع التعليمات للبرنامج حيث يسمح متصفح الويب بتصفح المساعدة مع الارتباطات التشعبية ويشير كما يلي :

```
> help.start()
starting httpd help server ... done
If nothing happens, you should open
'http://127.0.0.1:28255/doc/html/index.html' yourself
```

فظهر شاشة برنامج تحليل البيانات الاحصائية R ليحتوي على عدة ارتباطات توضح هذا البرنامج بالتفصيل وكما في الشكل (11-2)



Statistical Data Analysis

Manuals

An Introduction to R	The R Language Definition
Writing R Extensions	R Installation and Administration
R Data Import/Export	R Internals

Reference

Packages	Search Engine & Keywords
--------------------------	--

Miscellaneous Material

About R	Authors	Resources
License	Frequently Asked Questions	Thanks
NEWS	User Manuals	Technical papers

Material specific to the Windows port

CHANGES up to R 2.15.0	Windows FAQ
--	-----------------------------

شكل (11-2) شاشة تحليل البيانات الاحصائية R

- (2) للاستفسار عن ايعاز ما او أي دالة اسمها محدد نستخدم الامر : `>help(...)` حيث نضع بين القوسين الايعاز او الكلمة التي تُريد الاستفسار عنها.
- (3) او نكتب علامة الاستفهام ونذكر بعدها الايعاز او الكلمة التي تُريد الاستفسار عنها. `>?solve` او `e.x.?mean`
- (4) للبحث عن مفهوم معين `help.search("...")` حيث يسمح بالبحث عن المساعدة بطريق مختلفة .

e.x.: help.search("data input")

(5) للبحث في جميع ارشيف R نستخدم ?? قبل الكلمة المراد الاستفسار عنها ،

e.x.: ?? mean

(6) الأمر example هو وسيلة معايدة أخرى متوفرة في لغة R لرؤية الأمثلة.

(7) الأمر example(mean) لذكر اسم الدالة المراد الحصول على أمثلة عملية عنها

```
R Console (64-bit)
File Edit Misc Packages Windows Help
> citation()
To cite R in publications use:
R Development Core Team (2011), R: A language and
environment for statistical computing. R Foundation
for Statistical Computing, Vienna, Austria. ISBN
3-900051-07-0, URL http://www.R-project.org/.

A BibTeX entry for LaTeX users is

@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Development Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2011},
  note = {(ISBN) 3-900051-07-0},
  url = (http://www.R-project.org/),
}

We have invested a lot of time and effort in creating
R, please cite it when using it for data analysis. See
also 'citation("pkgname")' for citing R packages.
> |
```

شكل (12-2) شاشة الایعاز citation

(4) الاختصار ctrl+L : لمسح الشاشة المعروضة بدون مسح اي بيانات في الذاكرة.

اما الدالة rm(x) > فهو لمسح x من الذاكرة ، حيث ان x قد تكون متغير او متوجه او مصفوفة او اطار بيانات

والدالة rm(list=ls()) > لمسح كل ما يوجد الان في ذاكرة العمليات الحالية من بيانات تم تحميلها فيما سبق

(5) الدالة getwd() : لمعرفة دليل العمل working directory

(6) الدالة dir() والدالة list.files() لعرض كل الملفات في دليل العمل .

(7) يمكن تحديد عرض لخط النتائج وكما يلي:

```
> options(width=40)
> 1:50
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
[13] 13 14 15 16 17 18 19 20 21 22 23 24
[25] 25 26 27 28 29 30 31 32 33 34 35 36
[37] 37 38 39 40 41 42 43 44 45 46 47 48
[49] 49 50
> |
```

(8) قد تحتاج الى تحملة لا يعاز او كلمة ما في R ، وبعد كتابة حرف او اكثر نضغط مفتاح Tab تظهر عدة اقتراحات لتحملة الكلمة ومنها اختيار المطلوب ، لاحظ ذلك في المثال التالي :

```
> m
      mad          mahalanobis
maintainer   make.link    make.names
make.packages.html  make.rgb     make.socket
make.unique    makeActiveBinding  makeARIMA
makeClassRepresentation  makeExtends  makeGeneric
makeMethodsList    makepredictcall  makePrototypeFromClassDef
makeRweaveLatexCodeRunner  makeStandardGeneric manova
mantelhaen.test    mapply       margin.table
mat            mat.or.vec   match
match.arg      match.call   match.fun
[...truncated]
```

12.2 فضاء العمل The Workspace

فضاء العمل او مساحة العمل هي البيئة الحالية للعمل في R ويشمل اي كائنات معرفة من قبل المستخدم ، وعند نهاية العمل في بيئة R يمكن حفظ صورة من فضاء العمل الحالي ، والذي يمكن تحميله في المرة القادمة وذلك باستخدام عبارتي الحفظ والتحميل المذكورة سابقاً.

يتم ادخال الاوامر بشكل تفاعلي في R وباستخدام مفاتيح الاسهم صعوداً وهبوطاً يمكن التمرير بين اوامر مختلفة لاعادة استخدامها ، وفيما يلي بعض الاوامر القياسية لادارة العمل الخاص بك :

```
طباعة دليل العمل الحالي #  
getwd()  
كتابة الكائنات في مساحة العمل الحالية #  
ls()  
لتغيير الدليل #  
setwd(mydirectory)  
e.x.: setwd("c:/docs/mydir")
```

1.12.2 العمل مع الاوامر السابقة الخاصة بك

عرض اخر 25 امر #
history()

عرض جميع الاوامر السابقة #
history(max.show=Inf)

2.12.2 حفظ الاوامر السابقة

savehistory(file=" myfile")

3.12.2 استرجاع الاوامر السابقة

loadhistory(file=" myfile")

4.12.2 حفظ فضاء العمل الى الملف

R نهاية 5.12.2

ستطلب حفظ فضاء العمل $q() \ #$

13.2 الاوامر الاساسية في R

في لغة البرمجة R هناك الكثير من الاوامر او الدوال او التعبيرات الاساسية المفيدة والمهمة المتاحة التي تسمح للمستخدم للتعبير عن عمليات معقدة جداً بآيجاز. نتطرق اليها في هذا الفصل بشكل ابتدائي للتعرف على العمل في لغة البرمجة R :

1.13.2 العمليات الحسابية

تُستخدم R كالة حاسبة لتنفيذ العمليات الحسابية مثل الجمع والضرب وفيما يلي جدول بالعمليات الحسابية المتاحة في R.

الوصف	العملية
الجمع	+
الطرح	-
الضرب	*
القسمة	/
الاس	** او ^
المعامل (الباقي من القسمة)	%/%
القسمة الصحيحة	%/%%

```

> 2+7.2-5*3/2+2^4
[1] 17.7
> 31%7
[1] 3
> 16%5
[1] 3

```

<pre> > 5 + 3 [1] 8 > 15.3 * 23.4 [1] 358 > sqrt(16) [1] 4 </pre>	<pre> > v=c(2,5,7) > t=c(8,4,9) > v+t [1] 10 9 16 > v-t [1] -6 1 -2 > v*t [1] 16 20 63 > v/t [1] 0.2500000 1.2500000 0.7777778 > v%%t [1] 2 1 7 > v%/%t [1] 0 1 0 > v^t [1] 256 625 40353607 </pre>
--	--

يمكن حزن القيم في اسماء متغيرات لاعادة استخدامها في وقت لاحق

```
> product = 15.3 * 23.4 # save result
> product # show the result
[1] 358
> product <- 15.3 * 23.4 # <- is assignment operator, same as =
> product
[1] 358
> 15.3 * 23.4 -> newproduct # -> assigns to the right
> newproduct
[1] 358
```

2.13.2 الدوال الرياضية

- $\text{abs}(x)$ والتي تعيد القيمة المطلقة

- $\text{sqrt}(x)$ التي تحسب الجذر التربيعي او $x^{0.5}$.

- الدوال المثلثية المختلفة مثل $\sin(x)$ و $\cos(x)$ وغيرها.

```
e.x.>x=1:5
>y=sin(x)+cos(x)
```

- دوال التقريب المختلفة مثل $\text{floor}(2.718)$ والتي ستعيد القيمة 2 كأكبر عدد صحيح أصغر من القيمة المعطاة.

- الدالة $\text{ceiling}(3.142)$ والتي ستعيد القيمة 4 كأصغر عدد صحيح أكبر من القيمة

- دوال التحويل مثل $\log(x)$ التي تحسب اللوغاريتم الطبيعي للمقدار x .

- $\log_{10}(x)$ اللوغاريتم العشري لذات المقدار x .

- $\log(x,n)$ والتي تحسب اللوغاريتم لأي أساس يحدده المبرمج من خلال المقدار n ، فمثلا يمكنك حساب اللوغاريتم الثنائي للمقدار x باستخدام التعليمية $\log(x,2)$.

- $\exp(0)$

وفيما يلي امثلة للدوال الرياضية :

```
> sqrt(2)
[1]1.414214
> cos(pi)
[1] -1
> sin(20)^2+cos(20)^2
```

```
[1] 1
> log(1)
[1] 0
> log10(10)
[1] 1
> exp(0)
[1] 1
```

3.13.2 الدوال الاحصائية في R

الكثير من الناس تستخدم لغة البرمجة R باعتبارها نظام الاحصاءات ، وذلك لتنفيذ الاساليب الاحصائية الحديثة اضافة الى الدوال الكلاسيكية الاخرى المستخدمة في الاحصاء وفيما يلي بعض منها مع الامثلة:

```
> a=3;b=7;c=9
> x=sum(a,b,c)
> x
[1] 19
> min(x)
[1] 19
> a=3;b=7;c=9
> sum(a,b,c)
[1] 19
> x=c(3,7,1)
> max(x)
[1] 7
> min(x)
[1] 1
> mean(x)
[1] 3.666667
> median(x)
[1] 3
> var(x)
[1] 9.333333
> sd(x)
[1] 3.05505
```

- دالة الجمع $\sum (x_i - \bar{x})^2$
- $\max(x)$ تعيد القيمة العظمى ضمن العمود X
- قشى - $\min(x)$ تعيد القيمة الصغرى ضمن العمود X
- $\text{mean}(x)$ تعيد المتوسط الحسابي للقيم الواردة في X
- $\text{median}(x)$ تعيد قيمة الوسيط (الوسيط هو القيمة التي تقع في المنتصف عند ترتيب القيم تصاعديا، وعادة ما يستخدم الوسيط للدلالة على مركز المجموعة).
- $\text{var}(x)$ تحسب مقدار التباين من العلاقة
- $\text{sd}(x)$ تعيد قيمة الانحراف المعياري

- تمثل مدى البيانات Range
- لترتيب البيانات Sort

> x=c(1,2,3,4,5)	Rank	-
> range(x)	Order	-
[1] 1 5	Cumsum	-
> rank(x)	Cumprod	-
[1] 1 2 3 4 5		
> order(x)		
[1] 1 2 3 4 5		
> cumsum(x)		
[1] 1 3 6 10 15		
> cumprod(x)		
[1] 1 2 6 24 120		
> sort(x)		
[1] 1 2 3 4 5		

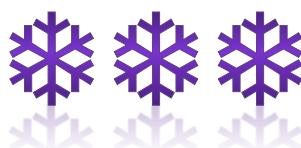
4.13.2 الاحصاءات الموجزة

في الاحصاء الوصفي تُستخدم

احصاءات موجزة لتلخيص مجموعة من الملاحظات ، من اجل ايصال اكبر قدر من المعلومات ببساطة ممكنة مثل الوسط الحسابي والانحراف المعياري ومعامل الارتباط وغيرها .

الدالة **summary(mpg)** فهي عامة الاستخدام ويختلف سلوكها وخرجها بحسب الكائن المُمرر إليها ، ففي حالة تمرير شاعع من القيم العددية فسيكون ناتج تنفيذها هو ملخص لتلك القيم والذي يشمل كل من المتوسط والوسيط إضافة إلى القيمتين العظمى والصغرى والربعين الأول والثالث (ويعرفان بشكل مشابه للوسيط ، إذ يشير الرابع الأول إلى القيمة التي نقل عنها ربع القراءاتك بعد ترتيبها تصاعديا ، فيما الرابع الثالث كما هو واضح من اسمه فهو القيمة التي نقل عنها ثلاثة أرباع قيم mpg المرتبة تصاعديا ، وهما قيمتان تساعدان في فهم كيفية توزع بياناتك). وفيما يلي مثال يوضح ذلك :

```
> x=c(1,2,3,4,5)
> summary(x)
   Min. 1st Qu. Median      Mean 3rd Qu.      Max.
1          2       3       3       4       5
```



اسئلة الفصل الثاني

- س1: ما هي لغة البرمجة R وما اهمية هذه اللغة ولماذا تُستخدم ؟
- س2: اذكر عشر مواضيع من شبكة الارشيف الشامل (كرا ترجمة CRAN) يمكن القيام بها في R ؟
- س3: ما مميزات لغة البرمجة R ؟
- س4: كيف يتم تشغيل وانهاء برنامج R ؟
- س5: عدد مكونات برنامج R ؟
- س6: اذكر استخدامات كل من الاوامر التالية :
- (1) GUI preferences ... (2) New script ...
- س7: عدد طرق الحصول على المساعدة في برنامج R ؟
- س8: اذكر اسم الدالة لكل مما يلي :
- 1) لعرض الطريقة الرسمية للإشارة الى لغة R كمرجع ضمن لائحة المراجع المستخدمة في اي ورقة علمية؟
 - 2) لاستعراض اخر 15 امر؟
 - 3) لمشاهدة كل الكائنات في الجلسة الجارية والمخزونة في R؟
 - 4) لمسح الشاشة المعروضة بدون مسح اي بيانات في الذاكرة؟
 - 5) لمسح x من الذاكرة ، حيث ان x قد تكون متغير او متوجه او مصفوفة او اطار بيانات
 - 6) لمسح كل العمليات الحالية من الذاكرة
 - 7) لمعرفة دليل العمل working directory
 - 8) لعرض كل الملفات في دليل العمل .
 - 9) لتحديد عرض خط النتائج
- س9: وضح استخدام كل من الرموز التالية ، مع ذكر مثال ؟

(1) #	(2) >	(3) <-
(4) ;	(5) [1]	(6) + السطر في بداية
(7) .	(8) اول اسم كائن_	(9) ^ او **
(10) *	(11) %/%	(12)%/%

س10: اذكر خطوات تنفيذ كل مما يلي:

1) تغيير مواصفات شاشة العرض في R

2) فتح مكان في البرنامج R لكتابه عبارات برنامج فقط دون تنفيذ

3) تكملة ايعاز او كلمة ما في R

س11: ما اوامر كل مما يلي :

(7) للاستفسار عن ايعاز ما	(4) المتوسط الحسابي	(1) انهاء برنامج R
(8) مسح workspace	(5) الوسيط	(2) القيمة العظمى
(9) الانحراف المعياري	(6) مقدار التباين	(3) القيمة الصغرى

س12: وضح استخدام الاوامر التالية في R ،مع ذكر مثال ؟

(1) median	(5) sqrt(x)	(9) abs(x)
(2) sd	(6) sin(x)	(10) cos(x)
(3) summary	(7) ceiling	(11) log10(x)
(4) exp(x)	(8) floor	(12) log(x,n)
(13) sum	(14) max(x)	(15) min(x)
(16) mean(x)	(17) var(x)	(18) sd(x)

س13: ما ناتج كل مما يلي :

1) >-2^2	2) 44% %5	3)>a=c(x,y,z); mean(a);sd(a); median(a);var(a);range(a);sort(a)
4)>sqrt(9)	3)26%/%5	6) > sin(20)^2+cos(20)^2
7)>cos(pi)	8)>log10(10)	9)> x=2;y=6;z=8;sum(x,y,z); min(x,y,z);max(x,y,z)
10)>sqrt(2)	11)> exp(0)	12)> log(1)
13)>"eee"		



الفصل الثالث

المتغيرات والثوابت

- 1.3 مقدمة
- 2.3 الكائن
- 3.3 المتغيرات في R
- 4.3 الثوابت في R
- 5.3 ثوابت اخرى
- 6.3 عمليات المقارنة
- 7.3 العمليات المنطقية
- 8.3 اسбقة التشغيل
- اسئلة الفصل الثالث



1.3 مقدمة

في كل جهاز كمبيوتر توفر متغيرات اللغة وسيلة للوصول إلى البيانات المخزنة في الذاكرة.

مثلاً نفترض أننا نود تخزين نتيجة الحساب $30^{0.0025}$.
 في المستقبل ويكون ذلك بحجز مكان في الذاكرة تحت اسم معين ولتكن a ليكتب بالصيغة: $>a<-1.0025^{30}$ او يستخدم علامة المساواة وتكون بالصيغة: $>a=1.0025^{30}$

2.3 الكائن Object

R لا يوفر الوصول المباشر إلى ذاكرة الكمبيوتر بل يوفر عدداً من هيئات البيانات المتخصصة وتنشير إليها بالكائنات أو الأشياء. ويشير إليها من خلال رموز أو متغيرات وهذا يختلف عن الكثير من اللغات الأخرى.

وقد تطرقنا في الفصل السابق صيغة دالة الربط والتي لها البناء اللغوي التالي:

(object1,...,c)، تُستخدم بشكل كبير لوضع قيم ذات نوع واحد في شكل متوجه مثل

```
>x<-c(1,2,3,"a")
```

```
> x
```

للتنفيذ أو الانتقال إلى سطر جديد نضرب مفتاح الادخال (Enter)

```
[1] "1" "2" "3" "a"
```

ولما كان الكائن يخزن معلومات كما في المثال أعلاه وهو (c(1,2,3,"a"))، لذلك غالباً ما نحتاج إلى تسميته بحيث نستطيع الإشارة إليه لاحقاً، وهذا ما يسمى بالمعرف والذى يتم اختياره حسب شروط معينة وهي كما يلى:
 1. يمكن ان تكون المعرفات مزيجاً من الحروف والأرقام وكذلك النقطة (.) و (_).
 ويفضل ان يكون واصف لطبيعة الكائن.

2. ان لا يبدأ بعد او نقطة

3. الحروف والكلمات المحجوزة في R لا يمكن أن تستخدم معرفات وهي كما يلى :

T , I , F , D , C , q , c

[FALSE](#), [if](#) , [TRUE](#) , [else](#) , [repeat](#) , [while](#) , [function](#) , [for](#) , [next](#) , [break](#),

NULL, Inf, NaN, NA, NA_integer, NA_real, NA_complex, NA_character.

ومن الامثلة على المعرفات الصالحة في R هي :
total, Sum, .fine.with.dot, this_is_acceptable, Number5

والامثلة على المعرفات غير الصالحة في R هي :
tot@l, 5um, _fine, TRUE, .0ne

3.3 المتغيرات في R

هي البيانات التي تتغير قيمتها حسب الحاجة ، وتسخدم المتغيرات لتخزين البيانات وتكون اسمائها فريدة (للدوال او الاشياء) ، ويكون اسم المتغير من الحروف والارقام والنقطة وان لا يبدأ برقم ولها الانواع التالية:

- 1 numeric
- 2 عددية معقدة complex
- 3 نصية character
- 4 منطقية logical
- 5 خاصة special

ويعتبر R بشكل افتراضي كل الاعداد التي نقوم بادخالها ثنائية double ، حتى نقوم نحن بتحديد ما إذا كنا نريدها صحيحة ولإنشاء متغير مثلاً له قيمة معينة يُخزن في الذاكرة ، حيث يمكن استدعائه عند الحاجة يكون بالشكل التالي : > x=2 او >y<-7 (المساواة لها نفس المعنى للرمز -<)

```
> a <- 1  
> b <- 2.3  
> c <- "Hellow, world"
```

4.3 الثوابت في R

وكما يوحى اسمها، هي الكيانات التي لا يمكن تغييرها. الانواع الأساسية من الثوابت هي الثوابت الرقمية والثوابت النصية.

1- الثوابت الرقمية : وتشمل كل الارقام ، ويمكن ان تكون من نوع عدد صحيح او معقد ، والدالة `typeof()` لمعرفة نوع الثابت ويعتبر الثابت الرقمي الذي يليه الحرف `L` عدد صحيح ومعقد تلك الذي يليه الحرف `i`

امثلة :

```
> typeof(5)
[1] "double"
> typeof(5L)
[1] "integer"
> typeof(5i)
[1] "complex"
```

2- الثوابت النصية : وتمثل بتحديد لها باستخدام علامات الاقتباس المفردة `(')` أو علامات اقتباس مزدوجة `(")`.

امثلة :

```
[1] "example"
> typeof("5")
[1] "character"
```

الدالة `mode` هي نمط او صيغة وتبين نوع الشيء كما في الامثلة التالية :

```
> x=1
> a="ali"
> c=TRUE
> b=2i
> mode(x)
[1] "numeric"
> mode(a)
[1] "character"
> mode(c)
[1] "logical"
> mode(b)
[1] "complex"
```

5.3 ثوابت اخرى

R لديها عدد قليل من الثوابت المضمنة ، حيث توفر الثوابت التالية :

(1) LETTERS : وهي الـ 26 حرف الانكليزي الكبير اي العلوية في مفاتيح لوحة المفاتيح ، وكما في المثال :

> LETTERS

[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N"
 "O" "P" "Q" "R" "S"
 [20] "T" "U" "V" "W" "X" "Y" "Z"

: وهي الـ 26 حرف الانكليزي الصغيرة وكما في المثال :

> letters

[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
 "p" "q" "r" "s"
 [20] "t" "u" "v" "w" "x" "y" "z"

: وهي اسماء اشهر السنة بالانكليزي ، وكما في المثال:

> month.name

[1] "January" "February" "March" "April" "May"
 "June"
 [7] "July" "August" "September" "October" "November"
 "December"

: اختصارات ثلاثة احرف لاسماء الاشهر الانكليزية ، وكما في المثال:

> month.abb

[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep"
 "Oct" "Nov" "Dec"

(5) للتعرف على التعبير العادي Regular Expressions يمكن الرجوع الى صفحة المساعدة الرئيسية حول هذا الموضوع بكتابة الابعاد ?regexp

> month.name[grep("A", month.name)]

[1] "April" "August"

: نسبة محيط الدائرة الى قطرها اي النسبة الثابتة ، كما في المثال :

> pi

[1] 3.141593

pi في هذه الحالة تمثل قيمة عددية مفروضة وكما في المثال :

> pi

[1] 56

6.3 عمليات المقارنة Comparison Operations

تستخدم للمقارنة بين القيم. وفيما يلي جدول بالعمليات المتاحة في R.

الوصف	العملية
اقل من	<
اكبر من	>
اقل من او يساوي	<=
اكبر من او يساوي	>=
يساوي	==
لا يساوي	!=

امثلة :

```

> x <- 5
> y <- 16
> x<y
[1] TRUE
> x>y
[1] FALSE
> x<=5
[1] TRUE
> y>=20
[1] FALSE
> y == 16
[1] TRUE
> x != 5
[1] FALSE
> m<-1:5; m==4
[1] FALSE FALSE FALSE TRUE FALSE

```

```
> v=c(2,5,7)
> t=c(8,4,9)
> v>t
[1] FALSE TRUE FALSE
> v<t
[1] TRUE FALSE TRUE
> v==t
[1] FALSE FALSE FALSE
> v>=t
[1] FALSE TRUE FALSE
> v<=t
[1] TRUE FALSE TRUE
> v!=t
[1] TRUE TRUE TRUE
```

7.3 العمليات المنطقية Logical Operations

في الجدول أدناه المعاملات المنطقية ووصف لكل منها :

```
> x <- c(TRUE,FALSE,0,6)
> y <- c(FALSE,TRUE,FALSE,TRUE)
> !x
[1] FALSE TRUE TRUE FALSE
> x&y
[1] FALSE FALSE FALSE TRUE
> x&&y
[1] FALSE
> x|y
[1] TRUE TRUE FALSE TRUE
> x||y
[1] TRUE
> s=c("male","Female","Female","male","male")
> s[c(FALSE,TRUE,TRUE,FALSE,FALSE)]
[1] "Female" "Female"
```

الوصف	العملية
لا	!
و	&
و للعنصر الاول	&&
او	
او للعنصر الاول	

8.3 أسبقية التشغيل

ان تنفيذ العمليات الحسابية او الدوال في التعبير الواحد، يتم حسب تسلسل العملية التي ستجرى. وتحدد الأسبقية في ترتيب التنفيذ من قبل لغة R ، وفي حالة وجود تعبيران او اكثر بنفس الاسبقية ف يتم حساب التعبير بين الاقواس اولاً ان وجدت وبعدها نأخذ بنظر الاعتبار الترتيب في التنفيذ مثلاً من جهة اليسار الى يمين التعبير والجدول التالي يوضح اولويات التنفيذ للعمليات:

العملية	الوصف
\wedge	الأس
$\% \%$	باقي القسمة
$*, /$	الضرب، القسمة
$+, -$	الجمع، الطرح
$<, >, <=, >=, ==, !=$	مقارنات
!	المنطقى NOT
$\&, \&&$	منطقية AND
$, $	منطقية OR
$->, ->>$	احالة نحو اليمين
$<- , <<- , =$	احالة نحو اليسار

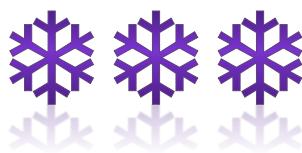
e.x.: $> 3 / 4 / 5$

[1] 0.15

$> 3 / (4 / 5)$

[1] 3.75

في المثال الاول، يكون الناتج حسب اسبقية التنفيذ من اليسار الى اليمين لنفس عملية القسمة اما في المثال الثاني فالناتج يختلف بسبب تنفيذ ناتج الاقواس اولاً.



اسئلة الفصل الثالث

س 1 : مادا يطبع كل مما يلي :

- 1) >x<-c(1,2,3,"a") ; >x
- 2) > typeof(5)
- 3) > typeof(5L)
- 4) > typeof(5i)
- 5) >'hellow'
- 6) > typeof("7.4")
- 7) a=TRUE;x=1 ;mode(a);mode(x)
- 8)> b=c(TRUE,FALSE,5) ; b ; mode(b)
- 9) >x<-c(2,TRUE,"hello") ; x ; mode(x)
- 10) >letters
- 11) >LETTERS
- 12) > month.name
- 13) > month.abb
- 14) > pi
- 15) >x<-2 ;y<-6;x<y;x>y;x!=5
- 16) >m<-1:5; m==4
- 17) >x=4:9 ; x+c(1,2)
- 18) >s=c("aa","bb","cc","dd")
 >s[c(FALSE,TRUE,FALSE,TRUE)]
- 19) > x <- c(TRUE,FALSE,0,6)
 > y <- c(FALSE,TRUE,FALSE,TRUE)
 > !x ; x&y ; x&&y ; x|y ; x||y ;
- 20) > s=c("male","Female","Female","male","male")
 > s[c(FALSE,TRUE,TRUE,FALSE,FALSE)]
- 21) >12/(6/2)

س2: ما هي شروط اختيار المعرف ؟

س3: ما المتغيرات في R ؟ اذكر انواعها ؟

س4: حدد نوع كل من المتغيرات التالية :

>n<-20

>m<-7.9

>k<-"ALI"

س5: ما الثوابت في R ؟ عددها مع التوضيح ؟

س6: في لغة البرمجة R ، اكمل ما يلي:

- (1) انواع المتغيرات هي:
- (2) قواعد كتابة المعرفات هي :
- (3) من الكلمات المحجوزة في R هي:
- (4) من الامثلة على المعرفات الصالحة :
- (5) من الحروف المحجوزة في R :

س7: وضح استخدام كل من الرموز التالية ، مع ذكر مثال ؟

- | | | |
|------------|---------|---------|
| (1) c | (5) i | (9) ! |
| (2) " او " | (6) L | (10) && |
| (3) > | (7) >= | (11) & |
| (4) < | (8) != | (12) |
| (13) <= | (14) == | (15) |

س8: وضح استخدام الاوامر typeof و mode في R ،مع ذكر مثال ؟

س9: ما اولويات تنفيذ العمليات التالية في لغة R :

<, >, <=,>=, ==, != ,^ ,% % ,+,-,!,->,<-,= , *,/ ,->>,<<-,|,&,||,&&



الفصل الرابع البيانات في R

- 1.4 مقدمة
- 2.4 المتجهات Vectors
- 3.4 القوائم Lists
- 4.4 المصفوفات والمجموعات Matrices and Arrays
- 5.4 الجداول Tables
- 6.4 العوامل Factor
- 7.4 دول للحصول على معلومات عن مجموعة البيانات
- اسئلة الفصل الرابع



1.4 مقدمة

في كل لغة برمجة يتم التعامل مع انواع مختلفة من البيانات والتي تكتب بصيغ خاصة بذلك اللغة وفيما يلي كائنات البيانات الموجودة في R والصيغ التي تتعامل فيها .

2.4 المتجهات Vectors

المتجهات هي بنية البيانات الأساسية في R. حيث يمكن انشاء متغيرات اكثر تعقيداً وبشكل جدول بصف واحد باستعمال الدالة c والمتجهات تحتوي على عناصر من نفس النوع. انواع البيانات المختلفة المتاحة في R هي منطقية، صحيح، حرف وعقد. امثلة :

a<- c(1,2,5,3,6,-2,4) # متجه رقمي

b<- c("one","two","three") # متجه حRFي

c<-c(TRUE,TRUE,FALSE,TRUE,FALSE) # متجه منطقي

الدالة typeof() : لمعرفة نوع المتجه

الدالة length() : لمعرفة عدد العناصر في المتجه.

الدالة c() : تستخدم لوضع الاشياء معاً في المتجه

امثلة :

>a[c(2,4)] # العنصر الثاني والرابع في المتجه

[1] 2 3

> c(0, 7, 8)

[1] 0 7 8

> x <- c(7.2, 3, 9)

> x

[1] 7.2 3.0 9.0

الرمز لانشاء تسلسل تصاعدي او تنازلي للقيم في مثال كما يلي :

> numbers5to20 <- 5:20

> numbers5to20

[1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

ممكن ربط المتجهات معاً :

> $y=c(0,7,8)$

> $c(numbers5to20, y)$

[1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 0 7 8

ول يكن :

> $some.numbers <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,$

+ 43, 47, 59, 67, 71, 73, 79, 83, 89, 97, 103, 107, 109, 113, 119)

يمكن الحق المتجه "some numbers" في نهاية المتجه "numbers5to20" ثم الحق تسلسل التناقص من 4 الى 1 وكما يلي :

> $a.mess <- c(some.numbers, numbers5to20, 4:1)$

> $a.mess$

[1] 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 59

[17] 67 71 73 79 83 89 97 103 107 109 113 119 5 6 7 8

[33] 9 10 11 12 13 14 15 16 17 18 19 20 4 3 2 1

لاستخراج فقط عنصر معين نكتب بالصيغة التالية:

> $a.mess[22]$

[1] 89

يمكن استخراج اكثـر من عنصر واحد في وقت واحد:

> $some.numbers[c(3, 6, 7)]$

[1] 5 13 17

للحصول على العناصر من 3 الى 7:

> $numbers5to20[3:7]$

[1] 7 8 9 10 11

لاختيار العناصر جميعاً ما عدا العنصر الثاني كما في المتجه التالي :

> $x <- c(0, 7, 8)$

> $x[-2]$

[1] 0 8

تجنب اختيار الارقام من 3 الى 11 كما في المتجه التالي :

> $some.numbers[-(3:11)]$

```
[1] 2 3 37 41 43 47 59 67 71 73 79 83 89 97 103 107
[17] 109 113 119
```

الدالة diff(): لمعرفة الفرق بين العناصر :

e.x.:

```
x <- c(1, 4, 6, 9, 23)
diff(x)
[1] 3 2 3 14
```

الدالة setdiff() : لمعرفة الفرق بين عناصر متوجهين :

e.x.:

```
x <- c(1, 4, 6, 9, 23)
y <- c(12, 4, 6, 78, 44)
setdiff(x,y)
[1] 1 9 23
```

(1) العمليات الحسابية على المتجهات Vector arithmetic

يمكن ان تتم العمليات الحسابية على المتجهات وعلى سبيل المثال ضرب جميع عناصر المتجه x في 3 :

```
> x * 3
[1] 0 21 24
> x <- c(0, 7, 8)
> y <- x - 5
> y
[1] -5 2 3
> x^3
[1] 0 343 512
```

يمكن حساب $y_1^{x_1}, y_2^{x_2}, y_3^{x_3} : y_i^{x_i}$ for $i=1, 2, 3$, i.e.

```
> y^x
[1] 1 128 6561
```

Recycling rule

يُستخدم تدوير او تكرار المتوجه الاصغر في حالة جمع متوجهين لهما طول مختلف:

e.x.:

```
> s=c(1,2)
> l=c(0,0,0,0,0,0)
> l+s
[1] 1 2 1 2 1 2
```

(2) عمليات اخرى على المتوجهات :

> rep(4,7) # تكرار القيمة 4 ، 7 مرات

[1] 4 4 4 4 4 4 4

تكرار 1 ثلث مرات و 4 مرتين # ((1, 4), c(3, 2))

[1] 1 1 1 4 4

تكرار كل قيمة 3 مرات # rep(c(1, 4), each=3)

[1] 1 1 1 4 4 4

تكرار كل قيمة مرتين # rep(seq(2, 20, 2), rep(2, 10))

[1] 2 2 4 4 6 6 8 8 10 10 12 12 14 14 16 16 18 18 20 20

> rep(seq(from=2,to=7,by=2),time=3)

[1] 2 4 6 2 4 6 2 4 6

(3) القيم المفقودة وقيم خاصة اخرى**Missing values and other special values**

رمز القيمة المفقودة هو NA (Not Available)، وهي كثيرة ما تنشأ في مسائل البيانات الحقيقية او تنشأ بسبب الطريقة التي يتم بها تنفيذ العمليات الحسابية .

> some.evens <- NULL # ينشأ متوجه بدون عناصر

> some.evens[seq(2, 20, 2)] <- seq(2, 20, 2)

> some.evens

[1] NA 2 NA 4 NA 6 NA 8 NA 10 NA 12 NA 14 NA 16 NA 18
NA 20

ما حدث هنا هو تعين قيم لعناصر 2, 4, ..., 20 لكن لم يُعين أي شيء للعناصر 3, ..., 19 لذلك R يستخدم NA للإشارة الى القيم الغير معروفة.

>a=c(1,2,3,NA,6) : ليكن :

>mean(a)

[1]NA

>mean(a,na.rm=TRUE) ولحذف القيم المفقودة يكون التعبير بالشكل :

[1]3 حيث rm تعني remove

ليكن x متوجه يحتوي على القيم (0,7,8) ، نجد ان :

> x / x

[1] NaN 1 1

تُستخدم الرمز NAN عندما تكون العملية الحسابية لا معنى لها ، وفي حالات اخرى، قد تظهر القيم الخاصة او قد تحصل على رسالة خطأ او تحذير:

> 1 / x

[1] Inf 0.1428571 0.1250000

(4) المتجهات الحرفية Character vectors

يمكن ان تحتوي المتجهات كلمات او قيم حرفية ، عند ادخال هذه القيم ، يجب ان تكون مزدوج (double) او مفرد (single) مزدوج double : ينشأ متجهات مزدوجة الدقة من طول محدد . عناصر المتجه كلها تساوي صفر ، وهي مطابقة لارقام .

> x = c("Bob","Carol","Ted","Alice")

> x

[1] "Bob" "Carol" "Ted" "Alice"

يمكن كتابة متجهين في متوجه واحد :

> y = c("John","Joy","Fred","Frances")

> z = c(x, y)

> z

[1] "Bob" "Carol" "Ted" "Alice" "John" "Joy"

[7] "Fred" "Frances"

> z = c("x", "y")

> z

> Joy طباعة موضوع #joy

خطأ: الكائن "joy" لم يتم العثور عليه

```
> "Joy"          # اطبع "Joy".
[1] "Joy"
> y[2]          طباعة ثاني قيمة في متوجه "joy".
[1] "Joy"
> Joy = 6       انشاء كائن جديد اسمه "joy"
> Joy
[1] 6
```

(5) تسمية المتوجه named vector
فيما يلي بيانات عن الجزر :

```
> islands
```

	Africa	Antarctica	Asia	Australia
11506	11506	5500	16988	2968
Axel Heiberg	Axel Heiberg	Baffin	Banks	Borneo
16	16	184	23	280
Britain	Britain	Celebes	Celon	Cuba
84	84	73	25	43
Devon	Devon	Ellesmere	Europe	Greenland
21	21	82	3745	840
Hainan	Hainan	Hispaniola	Hokkaido	Honshu
13	13	30	30	89
Iceland	Iceland	Ireland	Java	Kyushu
40	40	33	49	14
Luzon	Luzon	Madagascar	Melville	Mindanao
42	42	227	16	36
Moluccas	Moluccas	New Britain	New Guinea	New Zealand (N)
29	29	15	306	44
New Zealand (S)	New Zealand (S)	Newfoundland	North America	Novaya Zemlya
58	58	43	9390	32
Prince of Wales	Prince of Wales	Sakhalin	South America	Southampton
13	13	29	6795	16
Spitsbergen	Spitsbergen	Sumatra	Taiwan	Tasmania
15	15	183	14	26
Tierra del Fuego	Tierra del Fuego	Timor	Vancouver	Victoria
19	19	13	12	82

هذا يُسمى تسمية متوجه ، هنا هو كيفية انشاء واحد.

```
> x = c("Robert Culp","Natalie Wood","Elliott Gould","Dyan Cannon")
```

```
> x          لم تتم تسمية القيم حتى الان #
```

```
[1] "Robert Culp"   "Natalie Wood"  "Elliott Gould" "Dyan
Cannon"
> names(x) = c("Bob","Carol","Ted","Alice")
> x
      Bob       Carol       Ted       Alice
"Robert Culp"  "Natalie Wood" "Elliott Gould"  "Dyan
Cannon"
> x[Alice]          # وهذا غير صحيح ! لما لا ؟
Error: object "Alice" not found
> x["Alice"]
      Alice
"Dyan Cannon"
> Alice = 4
> x[Alice]          # نفس الشيء كما [4]
      Alice
"Dyan Cannon"
في "الجزر" المتوجه : قيم البيانات هي حجم كتلة اليابسة في الاموال المربعة
> islands["Cuba"]
Cuba
43
```

الدالة sort()

```
> x=c(2,6,1,9,4,78,67,90)
> sort(x)
[1] 1 2 4 6 9 67 78 90
> sort(c(5,0,7,1,3))
[1] 0 1 3 5 7
```

الدالة objects() : لسرد الكائنات في مساحة العمل

3.4 القوائم Lists

القائمة هي متجه عناصره مختلفة النوع ، كما في المثال التالي :

```
> x=1:10    # a vector
> y=matrix(1:12,nrow=3)    # a matrix
> z="Bill"        # a character variable
> my.list=list(x,y,z)    # creat the list
> my.list          #view the list
[[1]]
[1] 1 2 3 4 5 6 7 8 9 10

[[2]]
 [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

[[3]]
[1] "Bill"

> my.list[[3]]      # my.list[3]. العنصر الثالث في القائمة
[1] "Bill"
```

- لتسمية العناصر في القائمة

```

> names(my.list) = c("my.vector","my.matrix","my.name")
> my.list
Error: unexpected symbol in "my.list"
> my.list
$my.vector
[1] 1 2 3 4 5 6 7 8 9 10

$my.matrix
 [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

$my.name
[1] "Bill"

```

في R ، يتم استخدام \$ لقائمة الفهرسة. وهذا يعني أنه يسمح لك لسحب العناصر من القوائم بالاسم.
اكتب أولاً اسم القائمة، تليها \$، يليه اسم العنصر في القائمة.

e.x.:

```

> my.list$my.name
[1] "Bill"

```

```

> ls()
[1] "my.list" "x"      "y"      "z"

```

4.4 المصفوفات والمجموعات Matrices and Arrays
المجموعة مثل المصفوفة، إلا أنه يمكن أن يكون أكثر من بعدين. وبعبارة أخرى، المصفوفة ثنائية الأبعاد ، وان جميع الاعمدة في المصفوفة يجب ان تكون بنفس النوع (رقمي ، او حرفي ، الخ) وكذلك بنفس الطول ، والشكل العام للمصفوفة هو :

`mymatrix -> matrix(vector , nrow= r , ncol= c , byrow= FALSE)`

يشير `byrow=TRUE` الى ملئ الصفوف اولا اما `byrow=FALSE` فان المصفوفة تملأ من قبل الاعمدة (الافتراضي).

مثال :

```
> y<-matrix(1:20, nrow=5,ncol=4)
> y
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
```

(1) انشاء وتنمية المصفوفة :

لانشاء مصفوفة في R يُستخدم الامر `matrix` ، وغالباً ما تحتوي على قيم عدديه ، كما يمكن تحديد عدد الاعمدة باستخدام المعامل `nrow` وعدد الاسطع عن طريق المعامل `ncol` ، في المثال التالي مصفوفة بثلاثة اعمدة وسطرين :

```
> m=matrix(c(2,7,9,5,6,9),nrow=2,ncol=3)
> m
     [,1] [,2] [,3]
[1,]    2    9    6
[2,]    7    5    9
```

(2) معرفة قيمة العنصر في مصفوفة :

وبتحديد رقم الصف والعمود لمعرفة قيمة اي عنصر في المصفوفة لاحظ المثال التالي :

(3) كتابة اسماء الاعمدة او الصفوف

المصفوفة تُرتّب اولا الاعمدة ، رغم وجود خيار اخر من شأنه تغيير هذا السلوك ، يكتب لكل مصفوفة اسماء للاعمدة والصفوف كما يلي:

العنوان `[1]` يعني الصف الاول لكل الاعمدة

جميع القيم في الصف الاول #

```
> y[1,]
```

```
[1] 1 6 11 16
```

> y[,3] # جميع القيم في العمود الثالث

```
[1] 11 12 13 14 15
```

>y[-1,] # جميع قيم المصفوفة ما عدا الصف الأول

	[,1]	[,2]	[,3]	[,4]
[1,]	2	7	12	17
[2,]	3	8	13	18
[3,]	4	9	14	19
[4,]	5	10	15	20

>y[2:4,1:3] # للاعتمدة 1,2,3 للعمدة 2,3,4

	[,1]	[,2]	[,3]
[1,]	2	7	12
[2,]	3	8	13
[3,]	4	9	14

يمكن أيضاً اعطاء قيم للمصفوفة على شكل متوجه أو مجموعة متوجهات كما في المثال التالي :

مثال : لدينا قيم تعبير ثلاث جينات في ثلاثة تجارب ونريد وضعها في مصفوفة . ولجعل المصفوفة سهلة القراءة يمكننا استعمال الدالة rownames لتسمية السطور والدالة colnames لتسمية الاعمدة ، ويستعمل التعبير byrow=TRUE ليتم كتابة صفوف المصفوفة اولاً وليس الطريقة الشائعة في كتابة الاعمدة اولاً كما في الامثلة اعلاه

```
> g1<- c(10,3,8)
```

```
> g2<- c(5,2,7)
```

```
> g3<- c(3,10,1)
```

تُخزن في مصفوفة وتتمثل الصيغة

```
> exp<- matrix(c(g1,g2,g3),nrow=3,ncol=3,byrow=TRUE)
```

```
> exp
```

	[,1]	[,2]	[,3]
[1,]	10	3	8
[2,]	5	2	7
[3,]	3	10	1

(4) تسمية الصفوف والاعمدة

```

> cells <- c(1,26,24,68)
> rnames <- c("R1", "R2")
> cnames <- c("C1", "C2")
> mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE,
+ dimnames=list(rnames, cnames))
> mymatrix
   C1 C2
R1  1 26
R2 24 68
> rownames(exp) <- c("gene1","gene2","gene3")
> colnames(exp) <-c("experiment1","experiment1","experiment1")
> exp
      experiment1 experiment1 experiment1
gene1          10          5          3
gene2          3          2         10
gene3          8          7          1

```

(5) ابعاد المصفوفة :

نستخدم عبارة `dim` ثم اسم المصفوفة وكما في المثال المجاور:

(5) العمليات على المصفوفات :

-1 جمع او طرح المصفوفات : لجمع مصفوفتين يُكتب الايغاز : `m+n` وكما في المثال التالي :

```

> m=matrix(c(2,4,6,3,5,0),nrow=2,ncol=3)
> n=matrix(c(1,5,3,7,0,9),nrow=2,ncol=3)
> m;n
      [,1] [,2] [,3]
[1,]    2    6    5
[2,]    4    3    0
      [,1] [,2] [,3]
[1,]    1    3    0
[2,]    5    7    9
> m+n
      [,1] [,2] [,3]
[1,]    3    9    5
[2,]    9   10    9

```

ولطرح المصفوفات يُكتب الايغاز `m-n` وكما في المثال اعلاه

المثال اعلاه

2- ضرب المصفوفات :

(1) ضرب عنصر بعنصر :

```
> mat*mat
 [,1] [,2]
[1,] 100 169
[2,] 121 196
[3,] 144 225
```

(2) الضرب المتجهي :

```
> mat%*%t(mat)
 [,1] [,2] [,3]
[1,] 269 292 315
[2,] 292 317 342
[3,] 315 342 369
```

(3) فئة المصفوفة :

تستخدم الدالة class كتصنيف مفصل عن نوع الشيء وهي توضح استخدام بعض الدوال وكيفية التعامل مع الاشياء فمثلاً الاشياء من فئة مصفوفة تُعرض على الشاشة بطريقة خاصة وهذا لبقية انواع البيانات ، وفي حالة المصفوفة تكون الفئة matrix ونلاحظ استخدام بيانات حقيقية في مجموعة وهي ببساطة وضع البيانات في متجهات وفي مثالنا التالي البيانات من 1:16 وكما موضح ادناه :

```
> y = matrix(1:16, nrow=4) # صفوف من 4
> class(y) # هو كائن من فئة "مصفوفة"
[1] "matrix"
> y
 [,1] [,2] [,3] [,4]
[1,] 1 5 9 13
[2,] 2 6 10 14
[3,] 3 7 11 15
[4,] 4 8 12 16
> y[3,2]
[1] 7
```

دائماً وضع مؤشر الصف او لاً تليها مؤشر العمود، ودائماً وضع فهارس داخل أقواس مستطيلة.

(7) دور المصفوفة :



هو المصفوفة الناتجة عن تبديل الاعمدة بالاسطر للمصفوفة $A_{m \times n}$ فتصبح $A_{n \times m}^T$ ويرمز لها بالرمز

```
> mat<-matrix(10:15,nrow=3,ncol=2)
> mat
     [,1] [,2]
[1,]   10   13
[2,]   11   14
[3,]   12   15
> t(mat)
     [,1] [,2] [,3]
[1,]   10   11   12
[2,]   13   14   15
```

وفي مثال اخر :

```
> a=1:20
> dim(a)=c(4,5)
> a
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

(8) المصفوفة الاحادية :

```
> m=diag(3)
> m
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

(9) المصفوفة الصفرية :

لإنشاء مصفوفة باربعة اعمدة وثلاث صفوف ، تحتوي على قيمة 0 ، نكتب الايماز : > matrix(0,nrow=3,ncol=4)
او بصورة اخرى يُكتب كالتالي :

```
> matrix(0,3,4)
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```



(10) الدالة array()

توجد اكثراً من طريقة أخرى لتسمية الصفوف والأعمدة في مصفوفة حيث تستخدم الدالة array() لكتابة مصفوفة ببعدين ، كما في المثال التالي :

```
> y = array(1:8, dim=c(4,2))
> y
[,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```

و غالباً ما تُستخدم الدالة array() لكتابة مصفوفة بأكثر من بعدين حين ان الخيار "dim" يعطي عدد الصفوف والأعمدة والطبقات وكما في المثال التالي :

```
> y = array(1:16, dim=c(4,2,2))
```

```
> y
,, 1
[,1] [,2]
[1,] 1 5
[2,] 2 6
[3,] 3 7
[4,] 4 8
,, 2
[,1] [,2]
[1,] 9 13
[2,] 10 14
[3,] 15 11
[4,] 12 16
```

مثال

```
>a<- array(c('green','yellow'),dim=c(3,2,2))
>a
```



```

, , 1

[,1]      [,2]
[1,] "green"  "yellow"
[2,] "yellow" "green"
[3,] "green"  "yellow"

, , 2

[,1]      [,2]
[1,] "green"  "yellow"
[2,] "yellow" "green"
[3,] "green"  "yellow"

> x = c(1.26, 3.89, 4.20, 0.76, 2.22, 6.01, 5.29, 1.93, 3.27)
> y = matrix(x, nrow=3)

 خطأ: لا توجد دالة "matrix" اي خطأ املائي :
> y = matrix(x, nrow=3)
> y
[,1] [,2] [,3]
[1,] 1.26 0.76 5.29
[2,] 3.89 2.22 1.93
[3,] 4.20 6.01 3.27

```

(11) المعادلات الخطية Linear equation
 الحل لنظام المعادلات الخطية في R فقط بدلالة واحدة (solve) ، والمثال التالي يوضح ذلك :

$$\begin{array}{l}
 3x_1 + 2x_2 - x_3 = 1 \\
 2x_1 - 2x_2 + 4x_3 = -2 \\
 -x_1 + 0.5x_2 - x_3 = 0
 \end{array}$$

$$\left(\begin{array}{ccc|c}
 3 & 2 & -1 & 1 \\
 2 & -2 & 4 & -2 \\
 -1 & 0.5 & -1 & 0
 \end{array} \right) , A^*x=b , A^{-1}*b=x$$

```

> A=array(c(3,2,-1,2,-2,0.5,-1,4,-1),c(3,3))
> b=c(1,-2,0)

```

```
>solve(A,b)
[1] 1 -2 -2
```

5.4 الجداول Tables

دالة `table()` تُستخدم لإنشاء جداول تكرارية او جداول عرضية من البيانات الواردة في المتغيرات او اطار البيانات ، والنتيجة هي شيء يبدو في كثير من الحالات مشابه الى المصفوفة او المجموعة (array) وفيما يلي انشاء بعض البيانات:

```
> y = rnorm(100, mean=100, sd=15) # 100
> y = round(y, 0) # التقريب لعدد صحيح
```

نستخدمها لانشاء بعض البيانات لوضعها في الجدول ، حيث ان القيم في المتغير `y` هي عشوائية ،في كل المرة النتائج تكون مختلفة . لعرض جدول للأرقام وتكراراتها :

```
> table(y)
y
64 69 73 74 77 79 80 81 82 84 85 86 87 88 89 90 91
92 93
1 1 1 1 4 4 2 1 1 2 1 1 1 3 1 1 1 2 1
94 95 96 97 98 99 100 101 102 103 104 105 106 107 109
110 111 112 113
4 4 3 3 5 2 6 3 1 5 4 2 2 2 1 2 1 4 3
114 116 117 118 119 120 123 125 129
2 2 1 1 2 1 1 2 1
```

الصف العلوي من الأرقام يحتوي على قيم البيانات، التي يمكن أن نراها بمدى من 129-64 ، والصف السفلي من أرقام يعطي الترددات. قيمة البيانات (أي قيمة `y` هي مجموع التكرارات = 100) . الجداول بطبيعة الحال، تماماً مثل كل شيء آخر في R، يمكن تخزينها واستخدامها بعد ذلك لمزيد من التحليل :

```
> table(y) -> mytable      # خزنها
> barplot(mytable)
> ls()
[1] "myTable" "y"
```

```
> rm(myTable, y)
```

5.18.5 اطر البيانات Data frames

يتم استخدام إطار بيانات لتخزين جداول البيانات. ومن قائمة متوجهات متساوية الطول. فهي اعم من المصفوفة لكن الاعمدة تكون مختلفة بين رقم ، حرف ، عامل ، متوجه وما الى ذلك ، على سبيل المثال ، المتغير التالي df إطار بيانات تحتوي على ثلاثة متوجهات : n, s, b

```
> n = c(2, 3, 5)
> s = c("red", "green", "white")
> b = c(TRUE, FALSE, TRUE)
> df = data.frame(n, s, b)      # df هو اطار بيانات
> fix(df)
```

بيانات المتغيرات n,s,b مغلقة في إطار البيانات df ، ونستخدم الدالة fix لترتيب اطر البيانات في شاشة R Data Editor بشكل جدول حيث تمثل المتوجهات n ، s ، b عناوين الاعمدة وكما يلي :

	n	s	b	var4
1	2	aa	TRUE	
2	3	bb	FALSE	
3	5	cc	TRUE	
4				

يمكن استخدام الدالة التالية لتسمية المتغيرات وكما يلي :

```
> names(df)<-c("ID","color","passed") # اسماء المتغيرات
```

```
> df
  ID color passed
1  2   red    TRUE
2  3 green   FALSE
3  5 white   TRUE
```

وهناك مجموعة متنوعة من الطرق لتحديد عناصر إطار البيانات:

طباعة الاعمدة 2 و 3 من اطر البيانات اعلاه :

```
> df[2:3]
  color passed
1   red   TRUE
2 green FALSE
3 white TRUE
```

او الاعمدة ID و color من اطار البيانات : df

```
> df[c("ID","color")]
  ID color
1  2   red
2  3 green
3  5 white
```

ولطباعة احد الاعمدة من اطار البيانات نستخدم الابعاد التالي :
`> df$ID`

مثال : انشاء جدول يحتوي اسماء الجينات ونسبة التعبير :

```
> results<- data.frame(geneName = c("gene1","gene2","gene3"),
  expression= c(1,4,0.3) )
```

```
> results
  geneName expression
```

1	gene1	1.0
2	gene2	4.0
3	gene3	0.3

اما اذا اردنا الحصول على اسماء الجينات فقط ، فيكتب اسم المتغير + \$ + اسم العمود ، مثلاً:

```
> results$geneName
[1] gene1 gene2 gene3
Levels: gene1 gene2 gene3
```

مثال :

```

> x=c(1,2,3)
> y=sin(x)
> ss=data.fram(x,y)
Error: could not find function "data.fram"
> ss=data.frame(x,y)
> rm(x,y)
> x
Error: object 'x' not found
> y
Error: object 'y' not found
> fix(ss)
> ss$x
[1] 1 2 3
> ss$y
[1] 0.8414710 0.9092974 0.1411200
```

```

|    | x | y         | var3 |
|----|---|-----------|------|
| 1  | 1 | 0.841471  |      |
| 2  | 2 | 0.9092974 |      |
| 3  | 3 | 0.14112   |      |
| 4  |   |           |      |
| 5  |   |           |      |
| 6  |   |           |      |
| 7  |   |           |      |
| 8  |   |           |      |
| 9  |   |           |      |
| 10 |   |           |      |
| 11 |   |           |      |
| 12 |   |           |      |

يأتي محوظاً مع لغة R بشكل افتراضي وهو كما يلي:

> mtcars

|               | mpg  | cyl | disp | hp  | drat | wt   | ... |
|---------------|------|-----|------|-----|------|------|-----|
| Mazda RX4     | 21.0 | 6   | 160  | 110 | 3.90 | 2.62 | ... |
| Mazda RX4 Wag | 21.0 | 6   | 160  | 110 | 3.90 | 2.88 | ... |
| Datsun 710    | 22.8 | 4   | 108  | 93  | 3.85 | 2.32 | ... |

.....

السطر العلوي من الجدول، يدعى الرأس، يحتوي على أسماء الأعمدة. كل خط أفقى يدل على التوالى بعد ذلك البيانات، التي تبدأ مع اسم الصف، ثم تليها البيانات الفعلية. هنا هو قيمة الخلية من الصف الأولى، العمود الثاني من .mtcars

> mtcars[1, 2]

[1] 6

وعلاوة على ذلك، يمكننا استخدام أسماء الصفوف والأعمدة بدلاً من إحداثيات رقمية.

> mtcars["Mazda RX4", "cyl"]

[1] 6

```
> nrow(mtcars) # عدد بيانات الصف
[1] 23
> ncol(mtcars) # number of columns
[1] 11
```

```
> help(mtcars)
```

- بدلاً من طباعة إطار البيانات بالكامل، غالباً ما يكون من المرغوب فيه معاينة البداية مع دالة `head`

```
> head(mtcars)
```

|                                        |
|----------------------------------------|
| mpg cyl disp hp drat wt ...            |
| Mazda RX4 21.0 6 160 110 3.90 2.62 ... |

.....  
- دالة `expand.grid()` : تنشأ إطار بيانات من كافة تركيبات المتجهات او العوامل وكما موضح في المثال التالي :

```
> aa=c("green","red","yellow")
> hh=c("3.2","2.5","6.1")
> ss=c("M","F","F")
> expand.grid(aa,hh,ss)
 Var1 Var2 Var3
1 green 3.2 M
2 red 3.2 M
3 yellow 3.2 M
4 green 2.5 M
5 red 2.5 M
6 yellow 2.5 M
7 green 6.1 M
8 red 6.1 M
9 yellow 6.1 M
10 green 3.2 F
11 red 3.2 F
12 yellow 3.2 F
13 green 2.5 F
```

## 6.4 العوامل Factor

العامل هو بنية البيانات المستخدمة في الحقول التي تأخذ فقط قيم معرفة مسبقا ، ولعدد محدود من القيم أي بيانات متكررة او مصنفة ، مثلا ذكر او انثى او مثل الحالة الاجتماعية قد تحتوي على قيم وحيدة من : اعزب ، متزوج ، مطلق ، أو الأرامل . في مثل هذه الحالة ، ونحن نعلم القيم الممكنة مسبقا وتسمى هذه القيم متميزة المستويات .

وفيما يلي امثلة على العامل في R .

```
> x=c("Yes","No","No","Yes","Yes")
```

طباعة القيم في x

```
[1] "Yes" "No" "No" "Yes" "Yes"
```

طباعة القيم في عامل(x)

```
[1] Yes No No Yes Yes
```

طبع المستويات

-لانشاء عامل نستخدم الدالة factor()

```
> x <factor(c("single","married","married", "single"))
```

x

```
[1] single married married single
```

Levels: married single

```
> x <- factor(c("single","married","married", "single"),
levels=c("single","married","divorced")); x
```

```
[1] single married married single
```

Levels: single married divorced

المستويات قد تكون محددة مسبقاً حتى اذا لم يتم استخدامها .

العوامل ترتبط ارتباطاً وثيقاً مع المتجهات ، في الواقع ، يتم تخزين العوامل مثل المتجهات الصحيحة . ويُرى هذا بوضوح من هيكلها .

```
> x <- factor(c("single","married","married", "single"))
```

> str(x)

Factor w/ 2 levels "married","single": 2 1 1 2

المستويات يتم تخزينها في متجهات حرفية

-العوامل أيضا يتم انشاؤها عندما نقرأ الأعمدة غير العددية في إطار البيانات .

افتراضيا دالة `data.frame()` تحول المتوجه الحرفى الى عامل .  
 - الدالة `nlevels()` تعطى عدد المستويات للعوامل وكما في المثال :

```
> ff=c('green','red','red')
> fff=factor(ff)
> fff
[1] green red red
Levels: green red
> nlevels(fff)
[1] 2
```

- الوصول الى مكونات العامل : وهو يشبه الى حد كبير المتوجه

```
> x
[1] single married married single
Levels: married single
```

الوصول للعنصر الثالث #

```
[1] married
Levels: married single
```

الوصول للعنصر الثاني والرابع #

```
[1] married single
Levels: married single
```

الوصول للكل ماعدا العنصر الاول #

```
[1] married married single
Levels: married single
```

باستخدام متوجه منطقي #

```
[1] single single
Levels: married single
```

```
>s=c("aa","bb","cc","dd")
>s[c(FALSE,TRUE,FALSE,TRUE)]
```

[1]"bb" "dd"

- تعديل العامل :

مكونات العامل يمكن تعديلها باستخدام مهام بسيطة. ومع ذلك، فإننا لا نستطيع اختيار القيم خارج مستوياته المحددة مسبقا.

> x

[1] single married married single  
Levels: single married divorced

> x[2] <- "divorced" # تعديل العنصر الثاني x

[1] single divorced married single

Levels: single married divorced

لايمكن تعريف قيم المستويات الخارجية # ارمل

- رسالة تحذير :

In `[-.factor`(`\*tmp\*`, 3, value = "widowed") :

مستوى عامل غير صالح يولد NA

> x

[1] single divorced <NA> single

Levels: single married divorced

والحل لهذه المشكلة هو إضافة قيمة إلى المستوى الأول

> levels(x) <- c(levels(x), "widowed")

اضافة مستوى جديد #

> x[3] <- "widowed"

> x

[1] single divorced widowed single

Levels: single married divorced widowed

مثال : في هذا المثال نلاحظ أنه لدينا مستويين (نوعين) من القيم f و m :

```
> gender <- factor(c("m", "m", "f", "m", "f", "f", "f"))
```

```
> gender
```

```
[1] m m f m f f f
```

```
Levels: f m
```

مثال : متغير الجنس مع 20 ادخالات "ذكور" و 30 ادخالات "إناث"

```
> gender <- c(rep("male",20), rep("female", 30))
> gender <- factor(gender)
> summary(gender)
female male
 30 20
> gender
[1] male male male male male male male male male male
[11] male male male male male male male male male male
[21] female
[31] female
[41] female
Levels: female male
```

**7.4 دول للحصول على معلومات عن مجموعة البيانات :**  
 فيما يلي ملخص لعدد من الدول المستخدمة في سرد محتويات الكائن او البيانات :

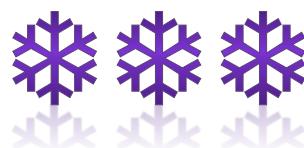
- (1) قائمة الكائنات في بيئة العمل : ls()
- (2) قائمة المتغيرات في بياناتك : names(mydata)
- (3) قائمة لهيكل البيانات : str(mydata)
- (4) ابعاد الكائن : dim(object)
- (5) فئة الكائن (رقمي ، مصفوفة ، اطار بيانات، الخ) class(object)
- (6) طباعة البيانات : mydata
- (7) طباعة الـ 10 صفوف الاولى من البيانات : head(mydata, n=10)

مثال : لتطبيق الدول اعلاه على بيانات mtcars نحصل على الاتي :

```

> ls()
[1] "data" "data1" "data2" "m" "mm"
> names(mtcars)
[1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear"
[11] "carb"
> str(mtcars)
'data.frame': 32 obs. of 11 variables:
 $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num 160 160 108 258 360 ...
 $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num 16.5 17 18.6 19.4 17 ...
 $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
 $ am : num 1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num 4 4 1 1 2 1 4 2 2 4 ...
> dim(mtcars)
[1] 32 11
> class(mtcars)
[1] "data.frame"
> head(mtcars,n=4)
 mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4
Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4
Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1
Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1

```



اسئلة الفصل الرابع

س 1 : مَاذَا يَطْبَعُ كُلُّ مَمَأْلِي :

- ```
1) >c(1,4,9)
2) > x <- c(0, 7, 8)
   > y <- x - 5 ; y;x^2;y^x
3) >a=c(1:5);b=c(1:10);a+b
4) > rep(3,8)
5)> rep(c(3, 4), c(4, 3))
6)> rep(c(6, 9), each=4)
7)> rep(seq(2, 20, 2))
8)> rep(seq(from=3,to=15,by=3),time=3)
9)> s=NULL;s[seq(1,10,2)]=seq(1,10,2);s
10)> x=c(3,6,8,NA,5,2);mean(x,na.rm=TRUE)
11)> x=c(3,6,8,NA,5,2);mean(x)
12)> a=c(3,0,-2);a/a
13)> b=c(4,0,-6);12/b
14)> a=c("2","g");a
15)> x <- factor(c("single","married","married","single"))
   ,                                         levels=c("single","married","divorced"));
x;str(x);nlevels(x)
```

س2: عدد انواع البيانات في برنامج R ؟ واذكر اوجه الاختلاف بين هذه الانواع ؟

س3: هل يمكن ربط المتجهات معاً؟ اعط مثال؟

س4: عرف ما يلي :

المتجهات (1) القيم المفقودة (2) المتجهات الحرفية (3)

س5: ما الفرق بين المصفوفات والمجموعات ? اذكر
اسم الدالة لكل منهما ؟

س6: وضح استخدام كل من الرموز التالية ، مع ذكر مثال ؟

(1) NA	(2) rm	(3) NaN
(4) Inf	(5) \$	

س7: وضح استخدام الاوامر التالية في R ، مع ذكر مثال ؟

(1) typeof()	(5) setdiff()	(9) table()
(2) length()	(6) sort()	(10) fix()
(3) c()	(7) objects()	(11) ncol()
(4) diff()	(8) class()	(12) nrow()
(13) head()	(14) str()	(15) names()

$$a = \begin{pmatrix} 3 & 6 & 2 \\ 1 & 4 & -8 \\ -2 & 7 & 9 \end{pmatrix}$$

اجب عن ما يلي في س8: ليكن لديك المصفوفة التالية:

لغة البرمجة R:

- (1) انشاء مصفوفة
- (2) قيمة العنصر في الصف الثاني والعمود الثالث
- (3) جميع القيم في الصف الثاني
- (4) جميع القيم في العمود الاول
- (5) جميع قيم المصفوفة ماعدا الصف الثالث
- (6) الصف الثاني والثالث للاعمدة الاول والثاني
- (7) اعط الاسماء (a1,a2,a3) للاعمدة والاسماء (b1,b2,b3) للصفوف

$$m = \begin{pmatrix} 2 & 9 & 6 \\ 7 & 5 & 9 \end{pmatrix}$$

س9: ليكن لديك المصفوفة التالية: اطبع في R ما يلي :

(1) m (2) m*m (3) دور المصفوفة (4) ابعاد المصفوفة

س10: اكتب دالة كل مما يلي :

- (1) مصفوفة احادية ثلاثة الابعاد
- (2) مصفوفة صفرية بثلاث اعمدة واربع صفوف

س11: حل المعادلات التفاضلية التالية باستخدام R ؟

$$1) 3x_1 + 2x_2 - x_3 = 1$$

$$2x_1 - 2x_2 + 4x_3 = -2$$

$$-x_1 + 0.5x_2 - x_3 = 0$$

$$2) 2a - 3b + c = 1$$

$$a + b - 2 = 3$$

$$3a - b + 2c = 5$$

س12: ليكن a اطار بيانات معرف كالاتي :

> a=factor(c("male","female","female","male"))

اكتب العبارات في R والتي تمثل ما يلي :

(1) اطار البيانات a (2) هيكل اطار البيانات (3) العنصر الثاني

(4) العنصر الاول والرابع (5) جميع العناصر ماعدا الثالث



الفصل الخامس

حفظ واستدعاء البيانات

1.5 مقدمة

2.5 حفظ واستدعاء كائن

3.5 استيراد البيانات من القرص المحلي

4.5 استخدام البيانات من R

5.5 قراءة البيانات من ملف تنسيق csv

6.5 قراءة البيانات من الملفات النصية txt

7.5 استيراد ملفات Excel إلى R

8.5 استيراد البيانات من الانترنت

9.5 استيراد ملفات البيانات باستخدام وظيفة المسح الضوئي

10.5 تصدير البيانات Exporting data

اسئلة الفصل الخامس



مقدمة 1.5

قد نجد في تحميل البيانات إلى R بعض القيود الواجب العمل بها لكل نوع من ملفات البيانات التي تحتاجها ، وهذا ما يتطلب دالة خاصة لكل نوع لاستيراد البيانات بسهولة وسرعة إلى R .

ولاستيراد البيانات إلى R، يجب أولاً الحصول على بيانات والتي يمكن ان تكون مخزونة في ملف على جهاز الكمبيوتر الخاص بك (على سبيل المثال في Excel، SPSS، أو أي نوع آخر من الملفات)، ولكن يمكن أيضاً أن يوجد على شبكة الإنترنت أو يمكن الحصول عليها من خلال مصادر أخرى. وقبل ان نمضي قدماً في كيفية تحميل البيانات الى R قد يكون من المفيد التعرف الى قائمة الاختيارات التالية والتي من شأنها أن تجعل من السهل على استيراد البيانات بشكل صحيح إلى R:

(1) إذا كنت تعمل مع جداول البيانات، الصف الأول عادة محفوظة للرأس، في حين

(2) تجنب الأسماء أو القيم أو الحقول والتي تحتوي مسافات فارغة، وإلا سيتتم تفسير كل كلمة كمتغير مستقل، مما أدى إلى الأخطاء التي ترتبط إلى عدد من العناصر في كل سطر في مجموعة البيانات.

(3) ادخال الكلمات بدلاً من الفراغات بين سلسلة كلمات .

(4) تفضل الاسماء القصيرة على الاسماء الاطول .

(5) حاول تجنب استخدام الأسماء التي تحتوي على رموز مثل :

? , \$, % , ^ , & , * , (,) , - , # , ? , , , < , > , / , | , \ , [,] , { , }

(6) حذف أي تعليقات قمت بها في ملف Excel لتجنب أعمدة إضافية أو NA تضاف إلى الملف.

(7) تأكُّد من أن يتم الإشارة إلى أية قيم مفقودة في مجموعة البيانات بذكر NA . والخطوة التالية في استيراد البيانات إلى R هي اعداد مساحة العمل R (R Workspace) فقد يكون لديك بيئة التي لا تزال مليئة بالبيانات والقيم حيث يمكن الاطلاع على محتوياتها باستخدام الدالة (ls) ، ثم يمكنك حذف جميع الكائنات من بيئة معينة باستخدام السطر التالي من التعليمات البرمجية: (rm(list=ls())) كما في المثال التالي:

```
> ls()
[1] "data1" "data2" "m"      "mm"
> rm(list=ls())
> ls()
character(0)
> |
```

وباستخدام الدالة `getwd()` يمكن معرفة المسار التي تحصل فيه على النتائج وربما الى المجلد التي قمت ب تخزين بياناتك فيه.

2.5 حفظ واستدعاء كائن :

في R يمكن تخزين كل كائن واستعادته من ملف مع الأوامر حفظ `save` وتحميل `load`، وكما في المثال التالي:

```
x <- 1:4
save(x,file="x.Rdata")      لخزن x
rm(x)                      لمسح x من الذاكرة
                           x
Error :object 'x' not found      يظهر الخطأ : الكائن "x" لا يوجد
load("x.Rdata")               لتحميل x
x [1] 1 2 3 4
```

3.5 استيراد البيانات من القرص المحلي :

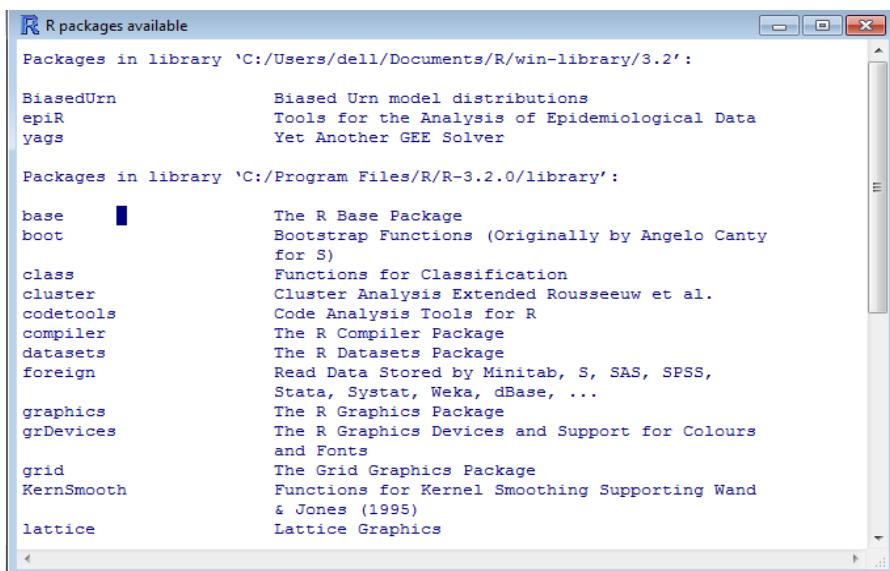
```
> getwd()
[1] "C:/Users/amb/Desktop/R python course"
> h2 <- read.table("C:/Users/amb/Desktop/R python course/ hsb2.csv")
Or > h2 <- file.choose()
- الدالة File.choose() : لفتح نافذة بالملفات واختيار الملف من المجلد او القرص.
```

4.5 استخدام البيانات من R

كل الحزم والبيانات هي مناسبة للاستخدام في عدة امثلة تأتي مع R ، وان استخدام البيانات من اي حزمة ، قد تحتاج الى تحميلها الى R من قبل الدالة:

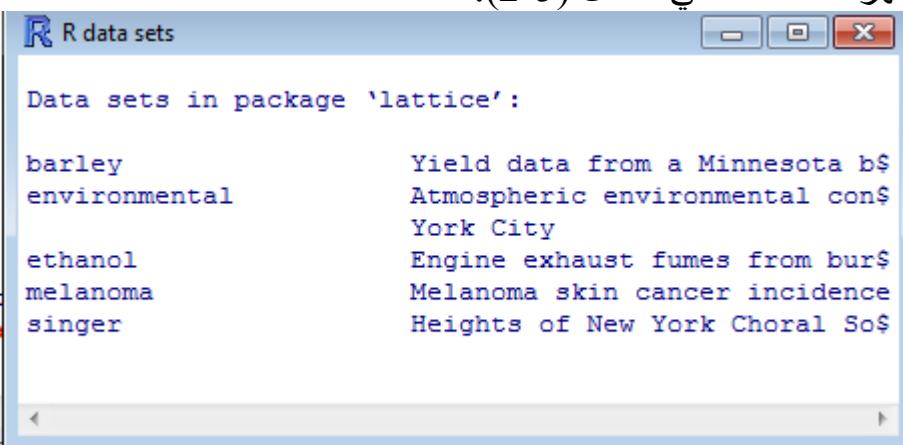
```
>library()
```

: (1-5) والتي تعرض نافذة بحزم البيانات المتوفرة في R وكما في الشكل



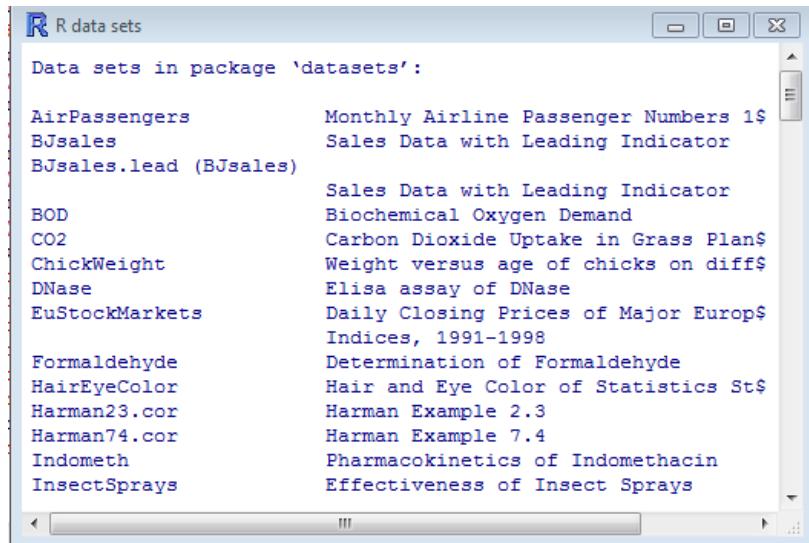
شكل (1-5) نافذة حزم البيانات في R

لاستخدام البيانات لا ي حزمة وهذا اسم الحزمة
عرض مجموعة البيانات في الحزمة
حيث تظهر النافذة كما في الشكل (2-5):



شكل (2-5) نافذة الحزمة lattice

اما الدالة data() فهي لعرض كل مجموعات البيانات في حزمة 'مجموعة البيانات'
وكما في النافذة في الشكل (3-5):



شكل (3-5) نافذة مجموعة البيانات

لتحميل واستخدام البيانات وهي في مثالنا
`> data(CO2) # CO2`
 لتحميل البيانات لست صفوف في المقدمة
`> head(CO2) #`

	Plant	Type	Treatment	conc	uptake
1	Qn1	Quebec	nonchilled	95	16.0
2	Qn1	Quebec	nonchilled	175	30.4
3	Qn1	Quebec	nonchilled	250	34.8
4	Qn1	Quebec	nonchilled	350	37.2
5	Qn1	Quebec	nonchilled	500	35.3
6	Qn1	Quebec	nonchilled	675	39.2

لتحميل البيانات لست صفوف في النهاية
`>tail(CO2) #`
 لتحميل البيانات لعشر صفوف في النهاية
`>tail(CO2,10) #`
 بمجرد استيراد بياناتك يمكنك الوصول إلى القيم الموجودة في أي من أعمدة جدولك باستخدام الصيغة `CO2$conc` على سبيل المثال حيث `conc` يشير إلى اسم العمود، أما إن أردت أسلوباً أكثر سهولة واختصاراً يقتصر على ذكر اسم العمود فقط دون الحاجة إلى ذكر اسم إطار البيانات المأخوذ منه في كل مرة، فعليك بدأياً استخدام الأمر `(attach(CO2))` عقب استيرادك للبيانات، وحينها يكفي ذكر الاسم `conc` للدلالة على ذات العمود من البيانات.

- الدالة attach() : هي لاستخدام اسماء المتغيرات في العمل مع البيانات :

```
> attach(CO2)
> names(CO2)
[1] "Plant"      "Type"       "Treatment"  "conc"       "uptake"
```

بعد الانتهاء من البيانات تُحفظ نسخة جديدة #

5.5 قراءة البيانات من ملف تنسيق csv
او الملف الذي يفصل بين قيمه ، او ; وكما

في المثال :

col1	col2	col3
1	2	3
4	5	6
7	8	9
a	b	c

الفاصل ، او تستخدم الدالة read.csv . وبذلك تكون قراءة الملف من نوع csv.
بالطريقتين :

```
df <- read.table("<FileName>.csv",header=FALSE,sep=",")
df <- read.csv("<FileName>.csv",header=FALSE)
```

محتويات لملف csv .

```
data <- read.csv("d:/mydir/myfile.csv", header=TRUE, sep=";")
```

التعبير header يعني ان السطر الاول من البيانات في الملف يحتوي على تسميات او لا .

فإذا header=TRUE معناها ان الصف الاول يحوي بيانات لاسماء المتغيرات .
والخيار sep يعين كيفية فصل وحدات البيانات ولها الاشكال التالية :

- Commas : sep="," الفارزة
- Tab : seq="\t" استخدام المفتاح
- Space : sep=" " الفراغ

```
> data<-read.csv("e:/nn.csv",header=TRUE)
> data
  a.b.c.d.e
1 q;w;e;r;t
2 1;2;3;4;5
3 z;x;c;v;b
```

6.5 قراءة البيانات من الملفات النصية .txt

وهي النوع الثاني الاكثر استعمالا في قراءة البيانات من الملفات النصية txt. حيث توجد الفواصل بين الحقول مثل الفراغ او النقطة الفاصلة ، وهنا توفر R دالة (read.table) التي تسمح بقراءة الملفات النصية وتعطيك تحكما تاما في كيفية القراءة. مثلا يمكنك أن تخبرها أن السطر الأول يمثل اسم الأعمدة وأن الحقول مفصولة بينها بفراغات بمفتاح Tab وليس Space. مثلا : محتويات لـ .txt :

1	6	a
2	7	b
3	8	c
4	9	d
5	10	e

لاستدعائه نكتب الايماز :

```
df<-read.table("<FileName>.txt",header = FALSE)
```

لاحظ أنه باستخدام هذه الدالة، والبيانات من ملف ستصبح كائنا .data.frame وأن FileName ليس دائما اسم الملف، ولكن ربما يمكن أيضا أن تكون صفحة الويب التي تحتوي على البيانات. تحدد header ما إذا كنت قد حددت أسماء الأعمدة في ملف البيانات. وتكون النتيجة النهائية للاستيراد على النحو التالي:

	v1	v2	v3
1	1	6	a
2	2	7	b
3	3	8	c
4	4	9	d
5	5	10	e

مثال : ليكن لدينا ملف يحتوي على البيانات التالية :

chr	start	end	length	CpGcount	GCcontent	pctGC	obsExp
chr10	93098	93818	721	32	403	0.559	0.572
chr10	94002	94165	164	12	97	0.591	0.841
chr10	94527	95302	776	65	538	0.693	0.702
chr10	119652	120193	542	53	369	0.681	0.866
chr10	122133	122621	489	51	339	0.693	0.88
chr10	180265	180720	456	32	256	0.561	0.893
chr10	180865	182549	1685	230	1263	0.75	0.984
chr10	242994	243152	159	10	74	0.465	1.193
chr10	313778	313905	128	6	64	0.5	0.769

لنفرض مثلاً أن اسم الملف هو data.txt يمكننا قرائة كالتالي:

```
> data <- read.table(file="data.txt", header=TRUE, sep="\t")
> data
```

chr	start	end	length	CpGcount	GCcontent	pctGC	obsExp
1 chr10	93098	93818	721	32	403	0.559	0.572
2 chr10	94002	94165	164	12	97	0.591	0.841
3 chr10	94527	95302	776	65	538	0.693	0.702
4 chr10	119652	120193	542	53	369	0.681	0.866
5 chr10	122133	122621	489	51	339	0.693	0.880
6 chr10	180265	180720	456	32	256	0.561	0.893
7 chr10	180865	182549	1685	230	1263	0.750	0.984
8 chr10	242994	243152	159	10	74	0.465	1.193
9 chr10	313778	313905	128	6	64	0.500	0.769

إذا كان حجم البيانات كبيرا من الأحسن استعمال أمر head لاظهار الأسطر الأولى من الملف، في العادة للتأكد من أن القراءة تمت بالشكل الصحيح. يمكن أيضا معرفة عدد الأسطر وعدد الأعمدة باستعمال أمر dim() .

> head(data)

```
chr start end length CpGcount GCcontent pctGC obsExp
1 chr10 93098 93818 721 32 403 0.559 0.572
2 chr10 94002 94165 164 12 97 0.591 0.841
3 chr10 94527 95302 776 65 538 0.693 0.702
4 chr10 119652 120193 542 53 369 0.681 0.866
5 chr10 122133 122621 489 51 339 0.693 0.880
6 chr10 180265 180720 456 32 256 0.561 0.893
```

> dim(data)

[1] 9 8

وفي مثال اخر لملف من نوع txt. مخزون على حاسوبك في جزء الذاكرة d :

> t=read.table("d:/test.txt",header=T)

> t

	make	model	mpg	weight	price
1	AMC	Concord	22	2930	4099
2	AMC	Pacer	17	3350	4749
3	AMC	Spirit	22	2640	3799
4	Buick	Century	20	3250	4816
5	Buick	Electra	15	4080	7827

7.5 أستيراد ملفات إلى Excel

لتحميل ملفات Excel إلى R، تحتاج أولا إلى القيام ببعض الإستعداد في تهيئة المزيد من مساحة العمل ، بمعنى أن تحتاج إلى تثبيت الحزم . ببساطة نكتب

التعليمات البرمجية التالية: > install.packages("اسم الحزمة")

وبعد تحميل الحزمة ، يمكن فقط كتابة ما يلي لتفعيلها في مساحة العمل :

>")

> library("اسم الحزمة")

8.5 استيراد البيانات من الانترنت

كما وتم القراءة من الانترنت باستخدام الدالة `read.table()`، وفيما يلي امثلة لموقع من الانترنت:

```
> s <- read.table("http://www.google.com/finance/historical?
q=NASDAQ:AAPL&output=csv", header=TRUE, sep=",")
>cs    <-   read.table("http://www.portfolioprobe.com/R/blog/
xassetCountrySector.txt", sep="\t", header=TRUE)
>dat.csv<-read.csv("http://www.ats.ucla.edu/stat/data/hsb2.csv")
```

هذا الامر يعمل عند الاتصال بالانترنت ، ثم نسخ ولصق البيانات لجذة R

9.5 استيراد ملفات البيانات باستخدام وظيفة المسح الضوئي

وظيفة المسح الضوئي هو أداة مرنة للغاية لاستيراد البيانات. ويمكن استخدامه لقراءة في ما يقرب من أي نوع من البيانات، رقمي، حرفي أو معقدة ويمكن استخدامه لملفات المنسقة ثابتة أو الحرة. وعلاوة على ذلك، باستخدام وظيفة المسح الضوئي من الممكن إدخال البيانات مباشرةً من وحدة التحكم.

في الأمثلة التالية أننا الإدخال الأول بيانات رقمية ثم سلسلة البيانات مباشرةً من وحدة التحكم. ثم إدخال ملف نصي، `scan.txt`، حيث المتغير الأول هو متغير سلسلة والمتغير الثاني هو رقمي.

- ادخال البيانات مباشرةً من وحدة التحكم

```
>x <- scan()
1: 3 5 6 9
5: 2 5 6
8:
Read 7 items
>x
[1] 3 5 6 9 2 5 6
> mode(x)
[1] "numeric"
```

- ادخال سلسلة بيانات مباشرةً من وحدة التحكم :

```
>name.x <- scan(, what="")
1: red blue
```

```

3: green red
4: blue yellow
7:
Read 6 items
>name.x
[1] "red" "blue" "green" "red" "blue" "yellow"
> mode(name.x)
[1] "character"

```

- ادخال ملف نصي واخراج قائمة

```

> (x <- scan("http://www.ats.ucla.edu/stat/data/scan.txt", what =
+ list(age = 0,
+ name = "")))
Read 4 records
$age
[1] 12 24 35 20

$name
[1] "bobby"    "kate"     "david"    "michael"

```

10.5 تصدير البيانات Exporting data:

الامثلة التالية توضح كتابة وخزن ملفات البيانات :

```

> write.table(mydata,"C://data/mydata.txt", sep ="\t")
> write.csv(dat.csv, file =path/to/save/filename.csv")
> write.table(dat.csv, file ="path/to/save/filename.txt", sep =
"\t", na=".")
> write.dta(dat.csv, file ="path/to/save/filename.dta")
> write.xlsx(dat.csv, file ="path/to/save/filename.xlsx",
sheetName="hsb2")
> save(dat.csv, dat.dta, dat.spss, dat.txt, file ="path/to/save/
filename.RData")

```



اسئلة الفصل الخامس

س1: ما هي النقاط التي يجب اخذها بنظر الاعتبار في استيراد البيانات بشكل صحيح في R ؟

س2: اذكر التعليمات البرمجية في R لكل مما يلي :

(1) للاطلاع على محتويات مساحة العمل من قيم وبيانات ؟

(2) لحذف جميع الكائنات ؟

(3) لمعرفة المسار الذي تحصل فيه على النتائج ؟

(4) خزن قيم x ؟

(5) مسح x من الذاكرة ؟

(6) تحميل x من الذاكرة ؟

(7) فتح نافذة بالملفات واختيار الملف من المجلد او القرص ؟

(8) عرض نافذة بحزم البيانات المتوفرة ؟

(9) عرض مجموعة البيانات في الحزمة lattice ؟

(10) عرض جميع البيانات ؟

(11) عرض البيانات للاسطر الستة الاولى في mtcars ؟

(12) عرض البيانات للاسطر العشرة الاخيرة في mtcars ؟

س3: ما الفرق بين ملف ترتيبه CSV وآخر txt ؟ اذكر مثال لكل منهم ؟

س4: ما الخيار الذي يعين كيفية فصل وحدات البيانات ؟ عدد اشكالها ؟

س5: وضح كل من :

(1) استيراد ملفات Excel الى R

(2) استيراد البيانات من الانترنت

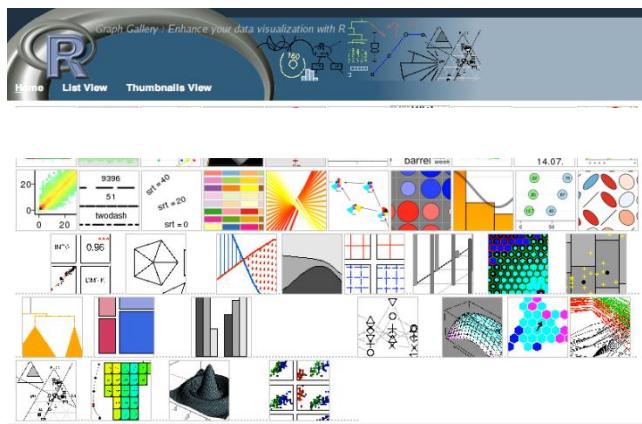
(3) المسح الضوئي في استيراد البيانات

(4) تصدير البيانات



الفصل السادس

الرسم البياني في R



مقدمة 1.6
hist() الدالة 2.6
plot(...) الدالة 3.6
Subplots 4.6
curve دالة المنحني 5.6
barplot الدالة 6.6
dotchart الدالة 7.6
Box plots الدالة 8.6
pairs() الرسم المزدوجة 9.6
layout() دالة 10.6
Pie charts التخطيطات الدائرية 11.6
رسم الكثافة 12.6
QQ رسم 13.6
Strip Chart الدالة 14.6
3D plot رسم ثلاثي الابعاد 15.6
Contour Plots 1.15.6
image plots 2.15.6
persp() 3.15.6
اسئلة الفصل السادس

1.6 مقدمة

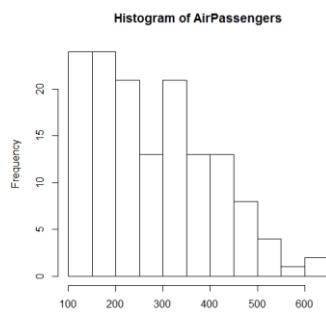
الرسم البياني هو تمثيل مرئي لتوزيع البيانات. حيث ان شكل الرسم البياني هو السمة المفيدة والأكثر وضوحا فهو يسمح لك أن ترى بسهولة حيث يقع على كمية كبيرة نسبيا من البيانات وحيث يوجد القليل جدا من البيانات التي يمكن العثور عليها. وبعبارة أخرى، يمكنك ان ترى فيها منتصف توزيع البيانات ومدى قرب هذه البيانات حول هذا الوسط، حيث يمكن العثور على القيم المتطرفة الممكنة. وبسبب كل هذا، فالرسوم البيانية هي طريقة رائعة للتعرف على البيانات . وباختصار، فإن الرسم البياني يتكون من المحور السيني x ، والمحور الصادي y ونقاط متعددة من ارتفاعات مختلفة. ويوضح المحور y عدد المرات التي تحدث القيم على المحور x في البيانات.

2.6 الدالة hist()

الرسم البياني للبيانات الفئوية يمكن انشاؤها باستخدام دالة `hist()` في لغة البرمجة R، تأخذ هذه الدالة في متجهات من القيم والتي يتم رسمها بيانياً. وان اسم مجموعة البيانات يوضع بين القوسين في هذه الدالة ، كما في الامثلة التالية :

مثال :

والذي يعطي الرسم البياني كما في الشكل(1-6) :



شكل(1-6) الرسم البياني لـ `hist(AirPassengers)`

يسمح R بعدة طرق سهلة وسريعة لتحسين صورة المخطط وكما في الدالة التالية :

```
hist(AirPassengers, main="Histogram for Air Passengers",
      xlab="Passengers", "border"=blue, "col"=green,
      xlim=c(100,700), las=1, breaks=5)
```

مثال : مجموعة بيانات airquality والتي فيها القياسات اليومية لنوعية الهواء في نيويورك ،مايو ، وثائق سبتمبر R-1973

> str(airquality)

'data.frame': 153 obs. of 6 variables:

\$ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...

\$ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...

\$ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...

\$ Temp : int 67 72 74 62 56 66 65 59 61 69 ...

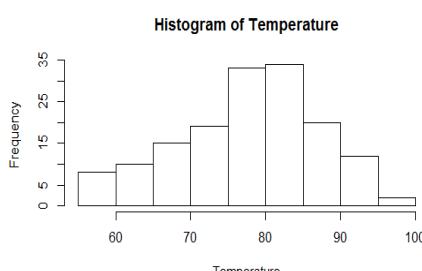
\$ Month : int 5 5 5 5 5 5 5 5 5 5 ...

\$ Day : int 1 2 3 4 5 6 7 8 9 10 ...

Temperature <- airquality\$Temp

hist(Temperature)

انظر الشكل(2-6)



شكل(2-6) الرسم البياني لـ *hist(Temperature)*

كما ويمكن اضافة المعلمات للرسم حيث:

تعطي العنوان main

و ylab لتوفير تسميات المحاور

xlim و ylim لتوفير مجموعة من المحاور

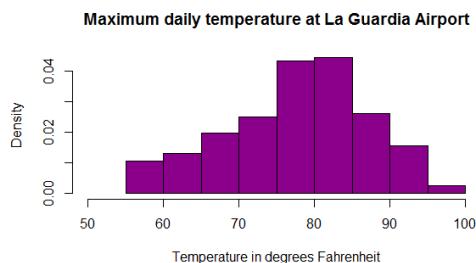
col لتحديد اللون

. للحصول على التوزيع الاحتمالي بدلاً من التردد Freq=FALSE

histogram with added parameters - الرسم مع اضافة المعلمات

hist(Temperature,main="Maximum daily temperature at La Guardia Airport", xlab="Temperature in degrees Fahrenheit", xlim=c(50,100),col="darkmagenta", freq=FALSE)

كما في شكل (3-6)



شكل (3-6) مع اضافة المعلومات
القيم المعلقة لدالة hist() : فهي تعطي قائمة لـ 6 مكونات يمكن الاطلاع عليها
بكتابة العبارات البرمجية التالية :

> h <- hist(Temperature)

> h

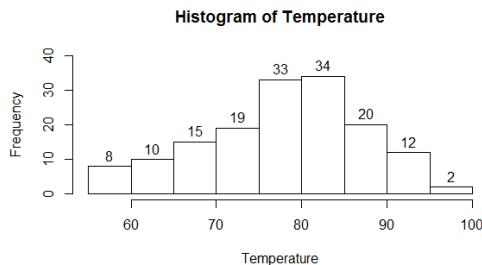
دالة الرسم البياني بتحديد احداثيات المحور y تكون كالتالي :

h <- hist(Temperature, ylim=c(0,40))

الدالة ادناه لوضع البيانات على الرسم:

text(h\$mid, h\$counts, labels=h\$counts, adj=c(0.5, -0.5))

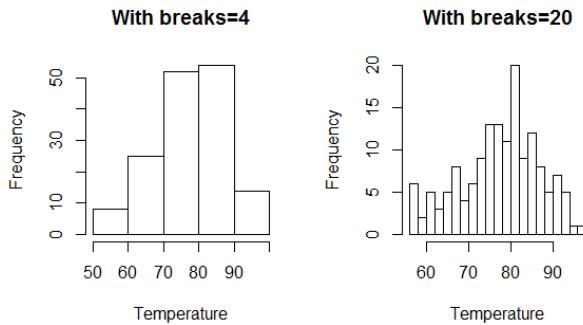
انظر الشكل (4-6)



شكل (4-6) وضع البيانات على الرسم
فيما يلي اثنين من الرسوم البيانية لنفس البيانات مع عدد مختلف من الخلايا وذلك
بزيادة عدد الفوائل من 4 الى 20 ، انظر شكل (5-6)

hist(Temperature, breaks=4, main="With breaks=4")

hist(Temperature, breaks=20, main="With breaks=20")



شكل (5-6) رسمين بعدد فوائل مختلفة

مثال :

```
> x <- c(12, 15, 13, 20, 14, 16, 10, 10, 8, 15)
> hist(x)

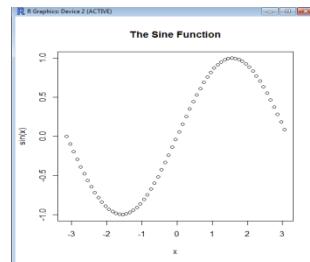
>hist(rnorm(100))
>hist(rnorm(100),breaks=50)
```

3.6 الدالة plot(...)

وهي دالة التخطيط الاكثر استخداماً في R ، فهي دالة عامة لوجود العديد من الاساليب التي تسمى وفقاً لنوع الكائن، وفي ابسط الحالات يمكن رسم المتغيرات بشكل مبعثر ، وعلى سبيل المثال : التعبير :

يرسم النقاطين : (1,3) و (2,5)
مثال (انظر الشكل 6-6) :

```
> x <- seq(1, 10)
> y <- x**2 - 10 * x
> plot(x, y)
```



شكل (6-6) رسم دالة plot()

وفيما يلي مثال اخر اكثرا واقعية عند رسم دالة الجيب للمدى بين $-\pi$ الى π

```
>x <- seq(-pi,pi,0.1)
>plot(x, sin(x))
```

```
>plot(x, sin(x),main="The Sine Function")
```

- تغيير لون ونوع الرسم
يمكنا أن نرى فيما سبق أن الرسم هو عبارة عن نقاط دائرية وسوداء اللون. هذا هو الافتراضي. ويمكن تغيير نوع الرسم ولونه وفيما يلي كل الرموز الممكن استخدامها والشكل الذي يرسمه:

"p"- نقاط

"l"- خطوط

"b"- نقاط وخطوط معاً

"c"- نقاط فارغة تربط بخطوط

"o"- دوائر صغيرة على نقاط الرسم وخطوط تصل بينها

"s" و "S"- خطوط درج

"h"- الخطوط العمودية مثل الرسم البياني

"n"- لا ترسم اي نقاط او خطوط

وبالمثل يمكن ان نحدد اللون باستخدام col

```
>plot(x, sin(x),main="The Sine Function",ylab="sin(x)",
```

```
type="l",col="blue")
```

```
plot(v,u,type,colour)
```

مثال

خلق سلسلة من 6- الى 6 مع زيادة 0.2

```
# v<-seq(-6,6,.2)
```

```
> u<-sin(v)
```

```
> plot(v,u,type='l',col='red')
```

```
# plot u=sin(v) with a red line 'l'
```

- تراكيب الرسومات

عند استخدام دالة plot() تعطي في كل مرة رسم بياني في نفس الاطار ليحل محل الرسم السابق ، وفي بعض الاحيان تحتاج الى وجود الرسمين معاً ولنفس الاحداثيات من اجل مقارنة النتائج ،والذي اصبح ممكناً مع الدالة lines() و points() وكما في المثال التالي :

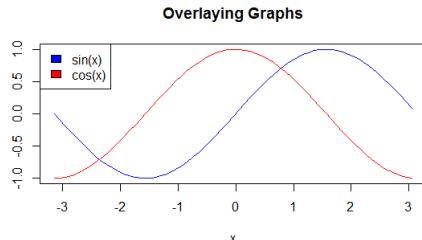
```
>x <- seq(-pi,pi,0.1)
```

```
>plot(x, sin(x),main="Overlaying Graphs", ylab="",type="l",
col="blue")
```

```
>lines(x,cos(x),col="red")
```

الدالة legend() تضيف وسيلة ايضاح الى الرسم ، حيث يحدد فيها موقع وسيلة الايضاح والتسمية لخطوط الرسم واللون لكل رسم، والمثال التالي يوضح ذلك (انظر شكل(7-6) :

```
legend("topleft",c("sin(x)","cos(x")),fill=c("blue","red"))
```



شكل(7-6) اضافة وسيلة ايضاح

Subplots 4.6

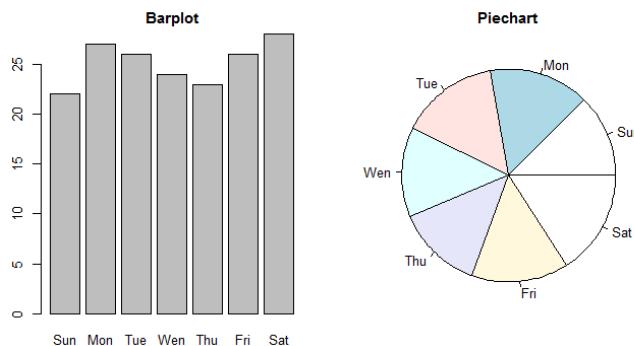
احياناً نحتاج الى وضع اثنين او اكثر من الرسوم البيانية في مكان واحد اي تقسيم مساحة الرسم ، البرمجة R لديها الكثير من المعلمات الرسمية التي تحكم في طريقة عرض الرسوم البيانية . دالة par() تعرض لائحة طويلة من المعلمات ولمعرفة عمل كل واحدة يمكن مراجعة التعليمات.

```
>par()
$xlog
[1] FALSE
...
$yaxt
[1] "s"
$ylbias
[1] 0.2
المعلمة الرسمية mfrow يمكن استخدامها لتحديد عدد subplot التي نحتاجها ،
فإن c(m,n) تقسم صفحة النتائج الى عدد من الرسوم وكما في المثال التالي(انظر
الشكل (8-6) :
par(mfrow=c(2,2)) # 2*2
تعين منطقة الرسم الى
```

```

> max.temp=c(22,27,26,24,23,26,28)
> names(max.temp)=c("Sun","Mon","Tue","Wen","Thu","Fri","Sat")
> max.temp
Sun Mun Tue Wen Thu Fri Sat
22 27 26 24 23 26 28
barplot(max.temp, main="Barplot")
pie(max.temp, main="Piechart", radius=1)

```



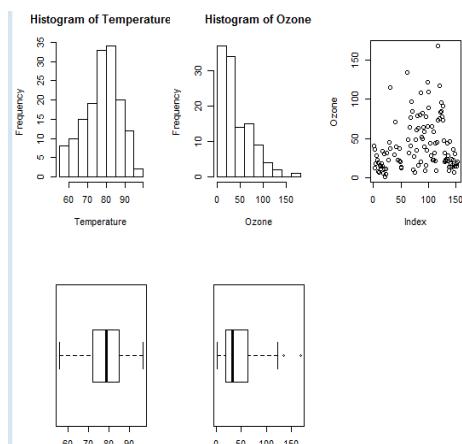
شكل(8-6) تقسيم صفحة النتائج

ويمكن تحقيق هذه الظاهرة نفسها مع `mfcoll` بفرق هو الرسم في الاعمدة اولاً .
فيما يلي ابعازات برسم 5 رسوم مختلفة في صفين وثلاث اعمدة :

```

>Temperature <- airquality$Temp
>Ozone <- airquality$Ozone
>par(mfcoll=c(2,3))
>hist(Temperature)
>boxplot(Temperature,
horizontal=TRUE)
>hist(Ozone)
>boxplot(Ozone,
horizontal=TRUE)
>plot(Ozone)

```



شكل(9-6) ابعاز mfcoll

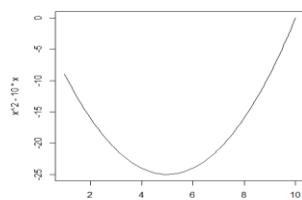
5.6 دالة المنحني curve

مثال: رسم دالة الجيب للفترة $[0, 6\pi]$

```
> curve(expr = sin, from=0, to = 6 * pi)
```

وفي مثال اخر انظر شكل (10-6)

```
> curve(x**2 - 10*x, from = 1, to = 10)
```



شكل (10-6) دالة curve

6.6 الدالة barplot

الدالة barplot() تستخدم في R للرسم الشريطي ، نفترض، لدينا متوجه من درجات الحرارة القصوى (في درجة مئوية) لمدة سبعة أيام على النحو التالي، انظر شكل (11-6).

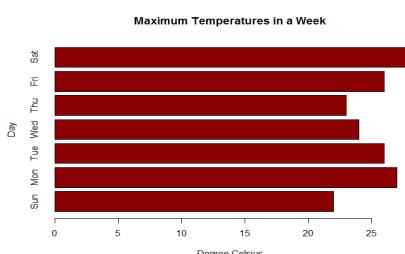
```
>max.temp <- c(22, 27, 26, 24, 23, 26, 28)
```

يمكن رسم شريطي من هذه البيانات :

```
>barplot(max.temp)
```

ويمكن اضافة معلمات رسومية الى تلك الدالة في اعطاء عنوان رئيسي وتنسيقات المحاور وتنمية كل شريط بيانات في الرسم وتحديد اللون وحتى يمكن الرسم الافقى من خلال المعلمة horiz=TRUE وفيما يلى مثال يوضح ذلك ، انظر شكل (12-6):

```
>barplot(max.temp,main="Maximum Temperatures in a Week",
```



شكل (12-6) تعليمية horiz=TRUE

Celsius",ylab="Day", names.arg =c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"), col="darkred",horiz=TRUE)

- الدالة table() : لمعرفة عدد المعطيات لكل شريط فعلى سبيل المثال : لدينا المتوجه التالي والذي يمثل اعمار مجموعة طلاب :

>age <- c(17,18,18,17,18,19,18,16,18,18)

لمعرفة عدد الطلاب في كل فئة عمرية

> table(age)

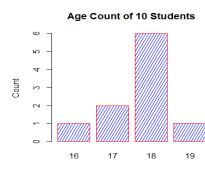
age

16 17 18 19

1 2 6 1

والآن نرسم عدد الطلاب لدالة table() انظر شكل (13-6):

>barplot(table(age),main="Age Count of 10 Students",xlab="Age",ylab="Count", border= "red" , col="blue", density=10)



شكل (13-6) دالة table()

- الرسم مع المصفوفة:

مثلاً : مجموعة البيانات VADeaths في R يحتوي على معدلات الوفيات (عدد الوفيات لكل 1000 من السكان سنوياً) في مختلف الفئات السكانية داخل ولاية فرجينيا في عام 1940. ويمكن رسم هذا بمثابة مخطط شريطي، انظر شكل (14-6): (rural:ريفي,urban:حضري)

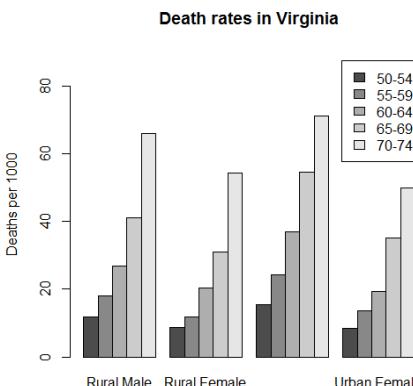
> VADeaths

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

```
> barplot(VADeaths, beside=TRUE, legend= TRUE, ylim=c(0, 90),ylab= "Deaths per 1000",main="Death rates in Virginia")
```

الاعمدة في الرسم تتوافق مع كل رقم في المصفوفة

الإياعز TRUE=beside يؤدي إلى أن قيم كل عمود ترسم جنباً إلى جنب
الإياعز legend=TRUE يعني أن وسيلة الإيضاح في أعلى اليمين تُضيق المقاييس
العمودي من الرسم البياني ، وأخيراً main تحدد العنوان الرئيسي للرسم.



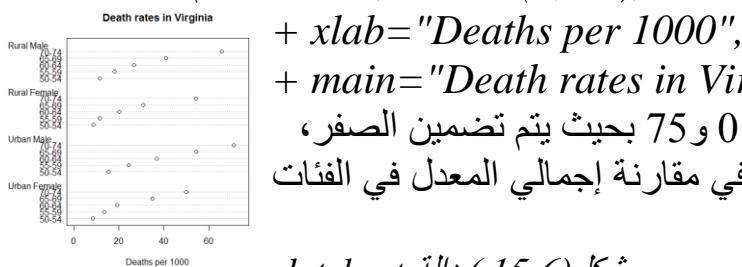
شكل(6-14) مخطط شريطي

dotchart الدالة 7.6

```
> dotchart(VADeaths, xlim=c(0, 75),
```

```
+ xlab="Deaths per 1000",
+ main="Death rates in Virginia")
```

هنا نعين حدود محور x بين 0 و75 بحيث يتم تضمين الصفر،
لأنه من الطبيعي أن ترغب في مقارنة إجمالي المعدل في الفئات
المختلفة .



شكل(6-15) دالة

مثال: ولتصنيف مجاميع ملونة ، انظر شكل (16-6)

```
> x <- mtcars[order(mtcars$mpg),] #mpg الترتيب حسب
```

```
>x$cyl <- factor(x$cyl) # يجب ان يكون عامل
```

```
>x$color[x$cyl==4] <- "red"
```

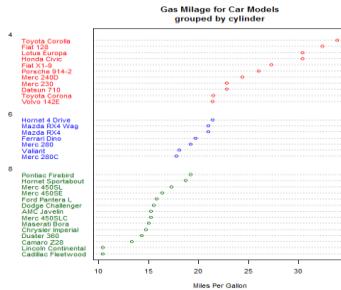
```
>x$color[x$cyl==6] <- "blue"
```

```
>x$color[x$cyl==8] <- "darkgreen"
```

```
>dotchart(x$mpg,labels=row.names(x) , cex=.7,groups=x$cyl,main= "Gas Milage for Car Models\ngrouped by
```

cylinder", xlab="Miles Per Gallon", gcolor="black", color=x\\$color)

- وان دالة colors() تعطي اسماء 657 من الالوان الموجودة في R



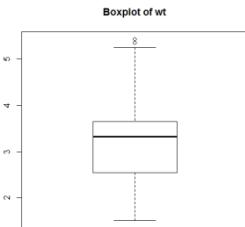
شكل (16-6) رسم مجامي ملونة

8.6 الدالة Box plots

مثال :

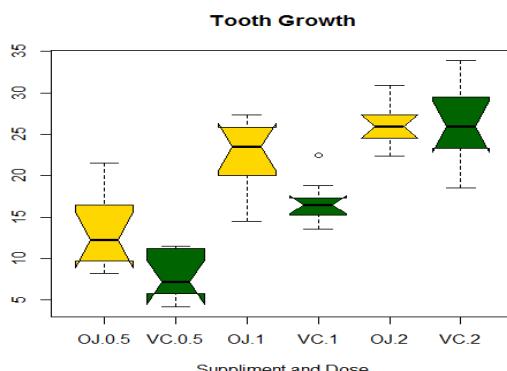
```
>attach(mtcars)
> boxplot(wt,main="Boxplot of wt")
```

مثال : # حق من نمو عبر عاملين
المربعات الملونة لسهولة التفسير



شكل (17-6) دالة boxplot

```
>boxplot(len~supp*dose,data=ToothGrowth,
notch=TRUE,
col=(c("gold","darkgreen")), main="Tooth Growth", xlab=
"Suppliment and hgfdhkhjDose")
```



شكل (18-6) بربعات ملونة boxplot

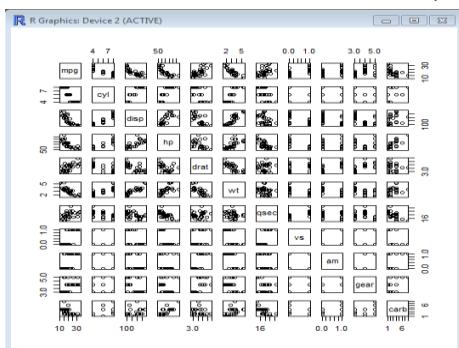
- دالة boxplot() تعطي قائمة من 6 عناصر عند كتابة العبارات البرمجية
التالية :

```
> b <- boxplot(airquality$Ozone)
> b
```

9.6 الرسوم المزدوجة pairs()

استخدمنا اطار البيانات "mtcars" مع 32 حالة على 11 متغير، لرسم مجموعة من الخرائط لكل زوج ممكن من هذه البيانات في شكل رسم بياني مبعثر بحيث كل زوج في مخطط متبادل الموقع حيث يظهر التمثيل في محاور x وy، يوضح الشكل التالي مثلاً نتائج تنفيذ هذا القانون عند تطبيقه على اطار البيانات (19-6 ، انظر شكل mtcars)

```
>pairs(mtcars)
```

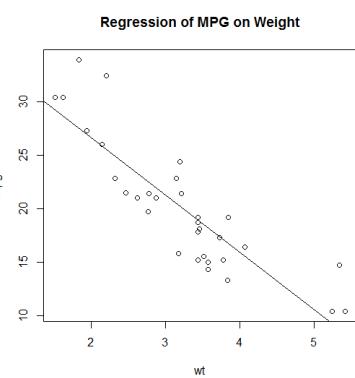


شكل(19-6) دالة pairs()

مثال :

إنشاء رسم بياني، انظر شكل (20-6)

```
>attach(mtcars)
>plot(wt, mpg)
>abline(lm(mpg~wt))
>title("Regression of MPG on Weight")
```

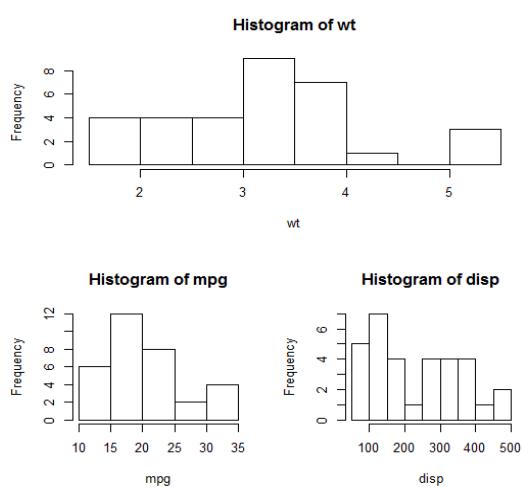


شكل(20-6) مخطط بين Weight و MPG

6 دالة layout()

ولها الصيغة `layout(mat)` ، حيث `mat` هي كائن مصفوفة تحدد موقع عدد من الاشكال في الرسم ، انظر شكل(21-6) # الشكل الاول في الصف الاول والشكليين في الصف الثاني

```
>attach(mtcars)
>layout(matrix(c(1,1,2,3), 2, 2,
byrow = TRUE))
>hist(wt)
>hist(mpg)
>hist(disp)
```

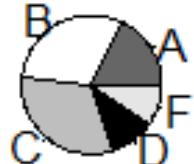


شكل(21-6) دالة layout()

11.6 التخطيطات الدائرية Pie charts

التخطيطات الدائرية تعرض متوجه من الأرقام بقطع قرص دائري إلى قطع حيث الزاوية (وبالتالي المنطقة) يتناسب مع كل عدد. على سبيل المثال، الدرجات الإلكتروني المخصصة لفئة قد تنشأ في النسب، الذي يرسم بإيعازات R التالية ، انظر شكل (22-6) :

```
> groupsizes <- c(18, 30, 32, 10, 10)
> labels <- c("A", "B", "C", "D", "F")
> pie(groupsizes, labels, col=c("grey40", "white", "grey",
"black", "grey90"))
```



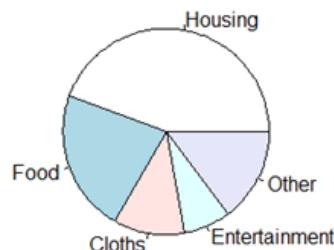
شكل(22-6) تخطيطات دائيرية

مثال: البيانات التالية تحتوي على انهيار الإنفاق الشهري للفرد، انظر شكل (22-6)

```
> expenditure
```

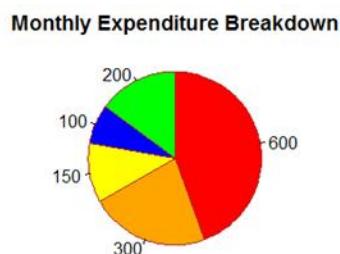
Housing	Food	Cloths	Entertainment	Other
600	300	150	100	200

الآن نرسم مخطط دائري بسيط من هذه البيانات باستخدام دالة pie()
pie(expenditure)



شكل(23-6) دالة pie()

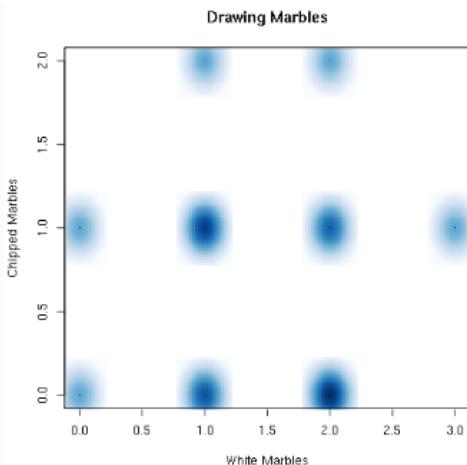
يمكن اضافة معلمات اخرى تؤثر على الرسم وكما يلي ، انظر شكل(24-6) :
> pie(expenditure, labels=as.character (expenditure), main="Monthly Expenditure Breakdown",col=c("red", "orange", "yellow" , "blue", "green"), border="brown", clockwise=TRUE)



شكل(24-6) دالة pie() واضافة المعلمات

12.6 رسم الكثافة

في اوقات معينة قد لا تحتاج لرسم نقاط مجده بل رسم كثافة النقاط ، ويتم ذلك باستخدام الامر smoothScatter وكما في المثال التالي، انظر شكل (25-6) :
> numberWhite <- rhyper(30,4,5,3)
> numberChipped <- rhyper(30,2,7,)
> smoothScatter(numberWhite,numberChipped,xlab="White +Marbles",ylab="Chipped Marbles",main="Drawing Marbles")

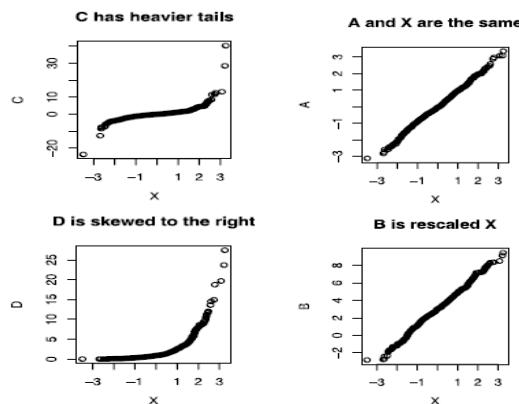


شكل(25-6) الامر *smoothScatter*(25-6)

13.6 رسم QQ plot()

هي نوع من الرسم مبعثر يستخدم لمقارنة التوزيعات من مجموعتين أو لمقارنة عينة مع توزيع إشارة في الحالة التي يكون فيها هناك مجموعتين متساوية في الحجم، يتم الحصول على الرسم QQ عن طريق الترتيب أولاً ، انظر شكل(26-6):

```
> X <- rnorm(1000); A <- rnorm(1000)
> qqplot(X, A, main="A and X are the same")
> B <- rnorm(1000, mean=3, sd=2)
> qqplot(X,B,main="B is rescaled X"); C <- rt(1000, df=2)
> qqplot(X, C, main="C has heavier tails")D=exp(rnorm (1000))
      > qqplot(X, D, main="D is skewed to the right")
```



شكل(26-6) دالة *qqplot()*

14.6 الدالة StripChart

يمكن انشاء مخطط شريطی باستخدام الدالة stripchart() في لغة البرمجة R، تأخذ هذه الدالة في المتوجه الرقمي او قائمة المتوجهات الرقمية ، والمثال التالي نستخدم البيانات اليومية لقياسات نوعية الهواء في نيويورك ، انظر شكل (27-6):

```
> str(airquality)
```

```
'data.frame': 153 obs. of 6 variables:
```

```
$ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...
```

```
$ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...
```

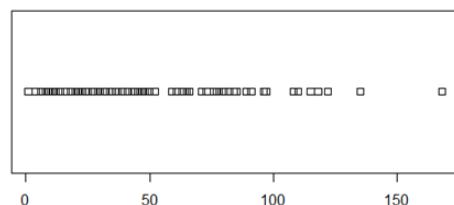
```
$ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
```

```
$ Temp : int 67 72 74 62 56 66 65 59 61 69 ...
```

```
$ Month : int 5 5 5 5 5 5 5 5 5 5 ...
```

```
$ Day : int 1 2 3 4 5 6 7 8 9 10 ...
```

```
>stripchart(airquality)
```



شكل(27-6) دالة stripchart()

- يمكن رسم قائمة متوجهات رقمية جنبا الى جنب للمقارنة، انظر شكل(28-6)

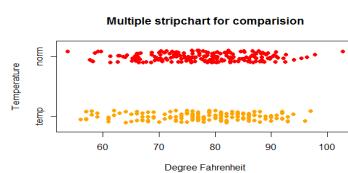
```
>temp <- airquality$Temp
```

```
>temp_norm <- rnorm(200,mean=mean (temp, na.rm=TRUE), sd=sd(temp, na.rm= TRUE))
```

```
>x <- list("temp"=temp, "norm"= temp_norm)
```

- وفيما يلي نرسم اثنين من stripchart معاً وكما يلي، انظر شكل (28-6)

```
>stripchart(x, main="Multiple stripchart for comparision", xlab="Degree Fahrenheit",ylab= "Temperature",method="jitter", col=c("orange","red"), pch=16)
```



شكل(28-6) اثنين من stripchart

15.6 رسم ثلاثي الابعاد 3D plot

يتمثل في ثلات متغيرات وهي x ، y ، z ، حيث x و y متجهان يحددان على المستوى اما الارتفاع فهو المتجه z والذي يتكون من مصفوفة للبعدين x و y .

Contour Plots 1.15.6

R لديها اثنين من الدوال في رسم contour وهي filled و contour ويمكن رؤية أمثلة على استخدامها من التعليمات البرمجية أدناه ، انظر شكل(29-6):

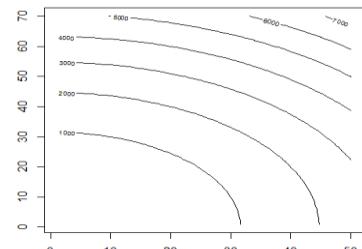
```
>x <- 1:50 ; y <- 1:70
```

```
>z <- matrix(expand.grid(x,y)$Var1^2+expand.grid(x,y)
```

```
>$Var2^2,50,70)
```

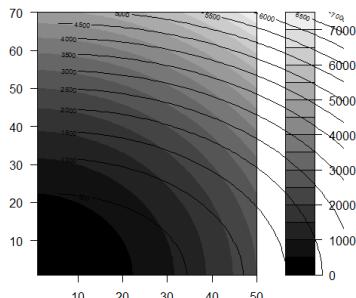
```
>contour(x,y,z)
```

الرسم



شكل(29-6) دالة contour() # مستويات للرسم

```
>mylevels <- seq(0,7500,500)
```



شكل(30-6) دالة filled.contour

```
>contour(x,y,z,levels=mylevels,xaxs='i',yaxs='i')
```

filled contours

```
>filled.contour(x,y,z,color.palette=heat.colors)
```

```
>filled.contour(x,y,z,col=grey(seq(0,1,length=length(mylevels)))
```

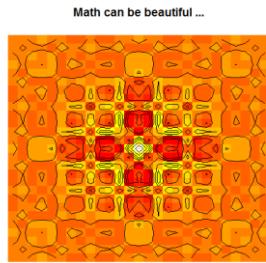
))

اضافة هذه العبارة الى العبارات اعلاه لنرى ما يحدث

```
>contour(x,y,z,levels=mylevels,add=T)
```

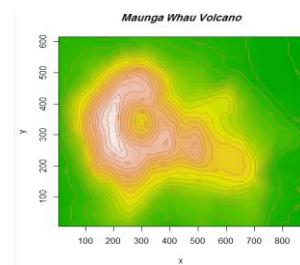
image plots 2.15.6

```
>x <- y <- seq(-4*pi, 4*pi, len = 27)
>r <- sqrt(outer(x^2, y^2, "+")) ; z <- cos(r^2)*exp(-r/6)
>image(z = z <- cos(r^2)*exp(-r/6), col = gray((0:32)/32))
>image(z, axes = FALSE, main = "Math can be beautiful ...",
+   xlab = expression(cos(r^2) * e^{-r/6}))
>contour(z, add = TRUE, drawlabels = FALSE)
# رسم لبيانات البركان والمتمثلة بمصفوفة ، انظر شكل(31-6)
image(t(volcano)[ncol(volcano):1,])
```



شكل(31-6) رسم بيانات برkan

```
# وفي عرض اخر للبركان من العبارات البرمجية التالية ،انظر شكل(32-6) :
x <- 10*(1:nrow(volcano)) ; y <- 10*(1:ncol(volcano))
image(x, y, volcano, col = terrain.colors(100), axes = FALSE)
contour(x, y, volcano, levels = seq(90, 200, by = 5),
        add = TRUE, col = "peru")
axis(1, at = seq(100, 800, by = 100)) axis(2, at = seq(100, 600,
by = 100)) box()
title(main = "Maunga Whau Volcano", font.main = 4)
```



شكل (32-6) رسم برkan بشكل اجمل

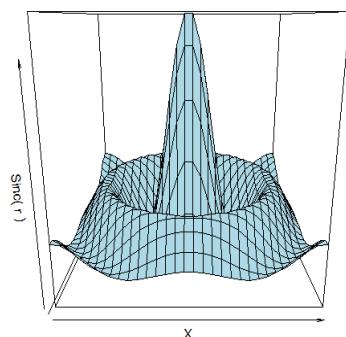
m = matrix(runif(100),10,10); image(m) مثال اخر:

persp() 3.15.6

هذه الدالة ترسم قطع السطوح فوق مستوى x-y وكما في المثال التالي :

```
>x <- seq(-10, 10, length= 30) :y <- x
>f <- function(x,y) { r <- sqrt(x^2+y ^2); 10 * sin(r)/r }
>z <- outer(x, y, f)
>z[is.na(z)] <- 1
>persp (x, y, z, aspect=c(1, 1, 0.5), col = "lightblue",xlab = "X",
ylab = "Y", zlab = "Sinc( r )")
```

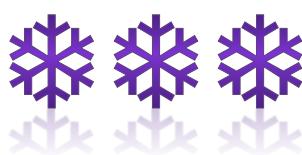
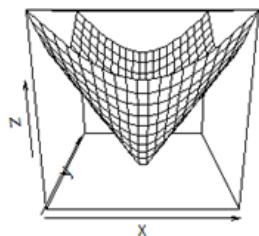
شكل(33-6) دالة persp()



يمكن الحصول على رسم مخروط دائري قائم من الدالة التالية، انظر شكل(34-6):
مثال:

```
cone <- function(x, y){ sqrt(x^2+y^2)}
x <- y <- seq(-1, 1, length= 20)
z <- outer(x, y, cone)
persp(x, y, z)
```

شكل(34-6) رسم مخروطي قائم



اسئلة الفصل السادس

- س1: عدد انواع الرسم البياني الثنائي الابعاد في R ؟ ذكر اسم الدالة ؟
 س2: وضح استخدام الاوامر التالية في R ، مع ذكر مثال ؟

(1) hist()	(5) par()	(9) Color()
(2) plot()	(6) barplot()	(10) boxplot
(3) lines()	(7) pie()	(11) pairs()
(4) legend()	(8) dotchart()	(12) layout()
(13) pie()	(14) qqplot()	(15) stripchart()
(16) image()	(17) persp()	(18) smoothScatter

س3: ما استخدام المعلمات التالية للرسم في R :

(1) main	(5) ylim	(9) mfcoll
(2) xlab	(6) col	(10) horiz
(3) ylab	(7) freq=FALSE	(11) border
(4) xlim	(8) mfrow	

- س4: الدالة hist() تعطي قائمة لـ 6 مكونات ؟ عددها فقط ؟
 س5: اجب عن ما يلي:
 (1) دالة الرسم البياني لتحديد احداثيات المحور y ؟
 (2) دالة وضع البيانات على الرسم ؟
 (3) دالة تضييف مخطط اخر ولنفس الاحاديث الى رسم سابق ؟
 (4) دالة تضييف وسيلة ايضاح ؟
 (5) دالة تقسيم مساحة الرسم ليستوعب اكثر من رسم بياني ؟

س6: اعط مثال لكل مما يلي :

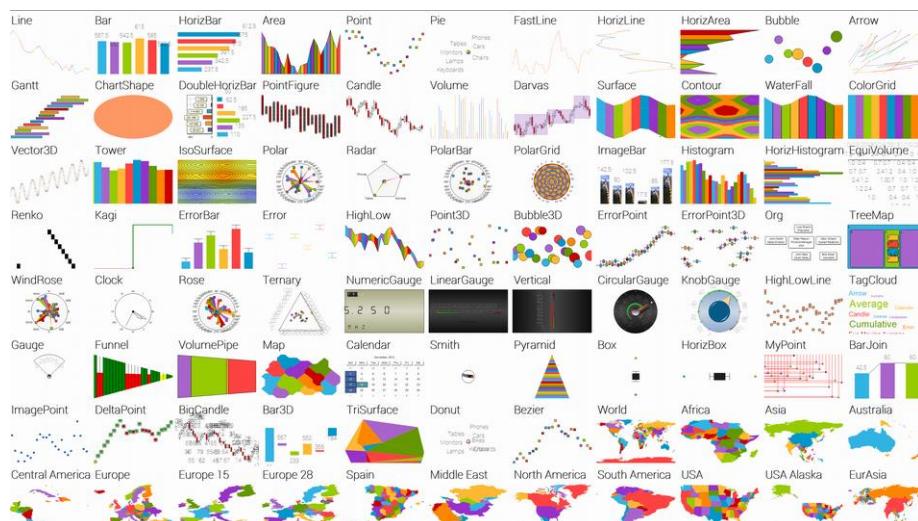
- (1) رسم دالة الجيب للمدى بين pi الى pi
- (2) لاربعة رسوم بياني متغيرة في مساحة العمل
- (3) مخططين لنفس الاحداثيات
- (4) رسم ثلاثي الابعاد
- (5) رسم الكثافة

س7: وضح استخدام الرموز التالية في الرسم البياني ؟

(1) p	(4) c	(7) h
(2) l	(5) o	(8) n
(3) b	(6) s , S	

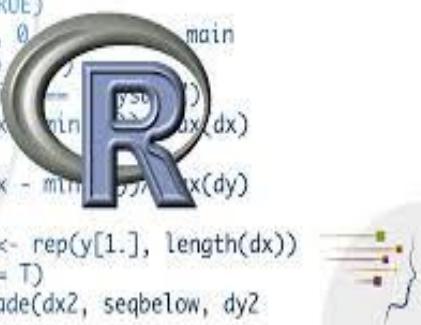
س8: الدالة boxplot() تعطي قائمة من 6 عناصر ؟ عددها ؟

س9: عدد انواع الرسم البياني الثلاثي الابعاد في R ؟ واذكر اسم الدالة ؟



الفصل السابع البرمجة في R

```
dens <- density(data, n = npts)
dx <- dens$x
dy <- dens$y
if(add == TRUE) {
  plot(0., 0, main = "", ylab = "density")
  if(orientation == "vertical") {
    dx2 <- (dx - min(dx)) / max(dx)
    x[1.]
    dy2 <- (dx - min(dy)) / max(dy)
    y[1.]
    seqbelow <- rep(y[1.], length(dx))
    if(Fill == T)
      confshade(dx2, seqbelow, dy2
                , fill = T, border = NA)
  }
}
```



1.7 مقدمة
2.7 عبارات البرمجة في R
3.7 عبارات IF
1.3.7 عبارة if statement
2.3.7 عبارة if...else
3.3.7 عبارة if...else المتداخلة
4.7 الحلقات Loops
1.4.7 عبارة for
2.4.7 عبارة while
3.4.7 عبارة break
4.4.7 عبارة Repeat
5.4.7 عبارة next
5.7 عبارة switch
6.7 الدوال Functions
7.7 دوال اخرى
8.7 RStudio
9.7 الفرق بين R و RStudio
اسئلة الفصل السابع

1.7 مقدمة

اهم نقاط الجذب الرئيسية في استخدام بيئة البرمجة R هو السهولة التي يمكن للمستخدمين كتابة البرامج والمهام الخاصة بهم ، كما ان بناء الجملة البرمجية غاية في سهولة التعلم ، حتى بالنسبة للمستخدمين الذين ليس لديهم خبرة سابقة في البرمجة .

وان واحدة من الامور الاساسية في برمجة R هو الوضوح والسهولة في مراقبة هيكل البرنامج . كما تُستخدم لغة R كبيئة قوية لإجراء تحليلات مخصصة معقدة من اي نوع تقريباً من البيانات .

2.7 عبارات البرمجة في R

لغة البرمجة R مثل اي لغة برمجة اخرى تمتلك عبارات برمجية خاصة تُستخدم لحل مشكلة ما وتحاكي الحاسوب لتحصل على النتائج وفيما يلي عبارات البرمجة المستعملة في لغة R وبناء الجملة والمخطط لكل عبارة

3.7 عبارات IF

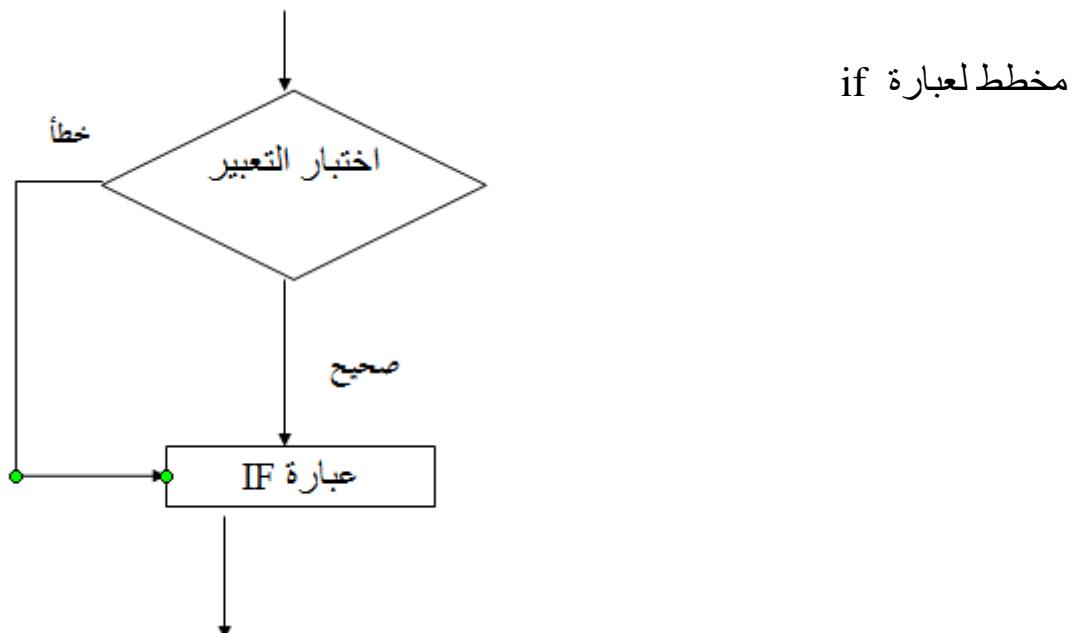
اتخاذ القرار هو جزء مهم من البرمجة . ويمكن تحقيق ذلك في البرمجة باستخدام R ، وفيما يلي توضيح لعبارات IF :

1.3.7 عبارة if statement

وهي تعبير منطقي يليه بيانات واحدة او اكثر - بناء جملة If

```
if (test_expression) {  
    statement  
}
```

إذا كان test_expression هو صحيح، يحصل تنفيذ العبارة. ولكن إذا كان خطأ، لا يحدث أي شيء. هنا، يمكن ان يكون test_expression متوجه منطقي أو رقمي، ولكن يؤخذ فقط العنصر الأول في الاعتبار. وفيما يلي مخطط يوضح هيكل تلك العبارة:



```

> if(1==0) {
+   print(1)
+ } else {
+   print(2)
+ }
[1] 2
  
```

- مثال :

```

x <- 5
if(x > 0){
  print("Positive number")
}
[1] "Positive number"
  
```

- مثال :

2.3.7 عبارة if...else

وهي عبارة if يليها عبارة else والتي تنفذ عند تعبير منطقي غير صحيح
- بناء جملة if...else

```

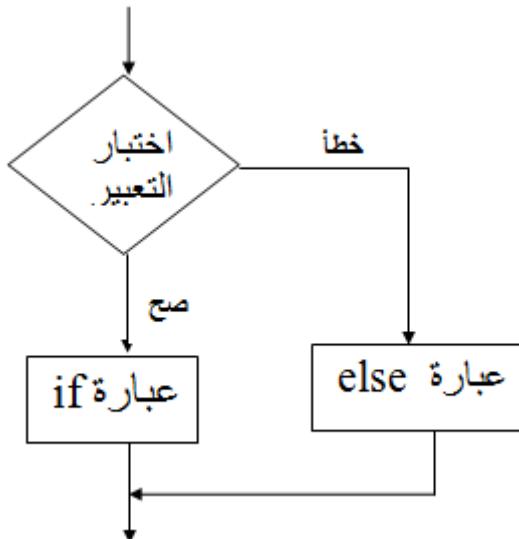
if (test_expression) {
  statement1
} else {
  
```

statement2

}

الجزء else اختياري ، ويتم التقييم اذا (test_expression) هو صحيح، ينفذ ما بعده وهو موجود بين الاقواس {}, ومن المهم ان نلاحظ ان else يجب ان يكون في نفس خط القوس المغلق لعبارة if والذي يُنفذ ما بعده عندما يكون التعبير خطأ.

- مخطط عبارة if...else



- مثال

```
x <- -5
if(x > 0){
  print("Non-negative number")
} else {
  print("Negative number")
}
[1] "Negative number"
```

يمكن ان يُكتب الشرط اعلاه في سطر واحد وكما يلي:

```
if(x > 0) print("Non-negative number") else print("Negative number")
```

هذه الميزة من R تسمح لنا بكتابة عبارات البرمجة كما هو مبين أدناه.

```
> x <- -5
> y <- if(x > 0) 5 else 6
> y
[1] 6
```

```
> x <- 1:10
> ifelse(x<5 | x>8, x, 0)
[1]  1  2  3  4  0  0  0  0  9 10
```

- مثال -

```
> even.odd = function(x) {
+ if (!is.numeric(x)) {
+ print("neither") }
+ else if (x%%2 == 0) {
+ print("زوجي")
+ else {
+ print("فردي")
+ } }
> even.odd(3)
[1] "odd"
> even.odd(4)
[1] "even"
> even.odd("A")
[1] "neither"
Or in other way:
> a = c(5,7,2,9)
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd" "odd" "even" "odd"
```

- مثال -

3.3.7 عبارة if...else المتداخلة

يمكن تداخل عدد من عبارات if...else كما ثرید وعلى النحو التالي:

- بناء جملة if...else المتداخلة

```
if ( test_expression1) {
    statement1
} else if ( test_expression2) {
    statement2
} else if ( test_expression3) {
    statement3
} else
    statement4
```

فقط جملة واحدة سوف تُنفذ اعتماداً على .test_expressions
مثال

```
x <- 0
if (x < 0) {
    print("Negative number")
} else if (x > 0) {
    print("Positive number")
} else
    print("Zero")
[1] "Zero"
```

مثال

```
> x <- 3
> if (x > 2) y <- 2 * x else y <- 3 * x
> y
[1] 6
> x=1
> if (x > 2) y <- 2 * x else y <- 3 * x
> y
[1] 3
```

مثال

```

> a=-2
> if(a>=1) b=3 else b=5
> b
[1] 5

```

4.7 الحلقات Loops

ان الاكثر شيوعاً في استخدام هيكل الحلقات في R هو عبارتي البرمجة for و while ، اما الاقل شيوعاً هو عبارة repeat ، وان عبارة break تستخدم للخروج من الحلقة

1.4.7 عبارة for

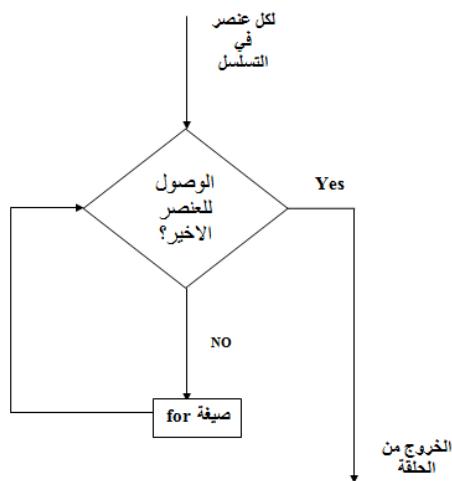
الحلقة تُستخدم لتكرار المتوجه في برمجة R
- بناء جملة for -

```

for (variable in sequence) {
  statements
}

```

هنا، sequence هو متوجه و variable يأخذ في كل مرة قيمته خلال الحلقة.
- مخطط لحلقة for -



- مثال : فيما يلي مثال لحساب عدد الارقام الزوجية في المتوجه التالي :
 $x <- c(2,5,3,9,8,11,6)$
 $count <- 0$

```
for (val in x) {
  if(val %% 2 == 0) count = count+1
}
print(count)
[1] 3
```

في المثال اعلاه لدينا 7 ارقام في التجه x ، اخذنا المتغير count ليكون عداد لحساب عدد الارقام الزوجية باختبار باقي القسمة على 2 اذا كان الناتج صفر فالعدد هو رقم زوجي ،وبذلك اصبح الناتج 3 .
- مثال -

```
x <- 1:10
z <- NULL
for(i in seq(along=x)) {
  if(x[i] < 5) {
    z <- c(z, x[i] - 1)
  } else {
    z <- c(z, x[i] / x[i])
  }
}
z
[1] 0 1 2 3 1 1 1 1 1 1
```

- مثال -

```
> for (x in 1:3) {
  print(x +
+ }
[1] 1
[1] 2
[1] 3
```

مثال

```
> for (x in c("hello", "goodbye")) {
  + print(x)
```

```
+ }
[1] "hello"
[1] "goodbye"
```

مثال

```
> m = matrix(1:4, nrow = 2, ncol = 2)
> for (x in m) print(x)
[1] 1
[1] 2
[1] 3
[1] 4
```

مثال : على متالية فيبوناتشي Fibonacci sequence

$$F_n = F_{n-1} + F_{n-2}, \\ F_0 = 0 \text{ و } F_1 = 1$$

```
> F=numeric(10)
>   F[1]=F[2]=1
>   for(i in 3:10)F[i]=F[i-1]+F[i-2]
> F
[1] 1 1 2 3 5 8 13 21 34 55
```

مثال : الدالة التالية تقوم على sieve of Eratosthenes ، اقدم اسلوب منهجي لادراج الاعداد الاولية تصل الى قيمة n

```

> Eratosthenes
function(n) {
  # Return all prime numbers up to n (based on the sieve of Eratosthenes)
  if (n >= 2) {
    sieve <- seq(2, n)
    primes <- c()
    for (i in seq(2, n)) {
      if (any(sieve == i)) {
        primes <- c(primes, i)
        sieve <- c(sieve[(sieve %% i) != 0], i)
      }
    }
    return(primes)
  } else {
    stop("Input value of n should be at least 2.")
  }
}
> Eratosthenes (50)
[1] 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
> Eratosthenes (-50)
Error in Eratosthenes(-50) : Input value of n should be at least 2.
> Eratosthenes (100)
[1] 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

```

2.4.7 عبارة while

وهي على غرار حلقة for ، لكن يتم التحكم في التكرار من قبل عبارة شرطية :
- بناء جملة حلقة while

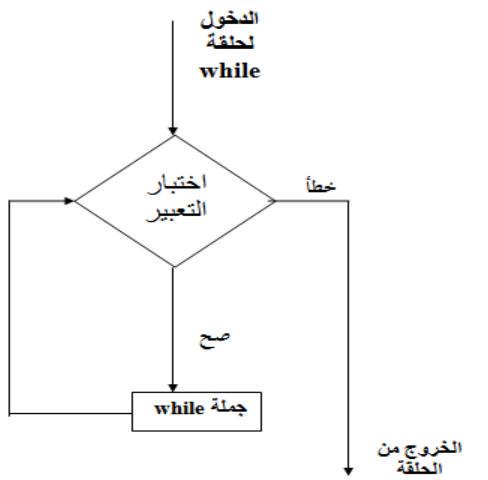
```

while (test_expression) {
  statement
}

```

هنا، يتم تقييم test_expression فإذا كانت النتيجة صحيحة يتم تنفيذ التعليمات داخل الحلقة ويستمر تقييم التعبير حتى تصبح النتيجة خاطئة ، في هذه الحالة، يتم إنتهاء الحلقة.

- مخطط حلقة while -



- مثال -

```

i <- 1
while (i < 6) {
  print(i)
  i = i+1
}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
  
```

في المثال أعلاه، يتم تهيئة i إلى 1. وهذا `test_expression` هو $i < 6$ الذي يقيم صحيح كلما i هو أقل من 6. لذا، تم تنفيذ داخل الحلقة وطباعتها وتستمر الحلقة حتى i يأخذ قيمة 6. فيصبح الشرط $i < 6$ وهذا يعطي النتيجة خطأ ويتم الخروج من الحلقة.

مثال

```

> x = 1
> while (x < 3) {
+ print(x)
+ x = x + 1
+ }
  
```

[1] 1
[1] 2

مثال : طريقة نيوتن لایجاد الجذر
وهي طريق شعبية لایجاد جذور معادلة جبرية

$$f(x) = 0$$

اذا (x) لها المشقة $(x)'$ ، ثم التكرار التالي سوف تتقرب الى جذر المعادلة المذكورة اعلاه اذا قريبة بما فيه الكفاية الى الجذر.

x_0 = تخمين اولي

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}.$$

وتعتمد الفكرة على تقرير تاييلور

$$f(x_n) \approx f(x_{n-1}) + (x_n - x_{n-1}) f'(x_{n-1}).$$

مثلاً : لنفترض $f(x) = x^3 + 2x^2 - 7$ ثم اذا x_0 قريبة بما فيه الكفاية لواحد من ثلاثة جذور لهذه المعادلة

$$x_n = x_{n-1} - \frac{x_{n-1}^3 + 2x_{n-1}^2 - 7}{3x_{n-1}^2 + 4x_{n-1}}$$

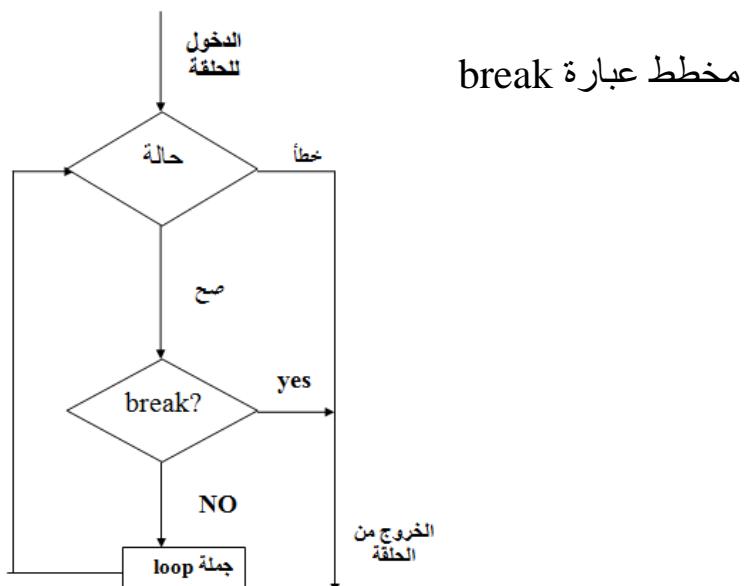
```
> x0=2
> x <- x0
> f <- x***3 + 2 * x**2 - 7
> tolerance <- 0.000001
> while (abs(f) > tolerance) {
+ f.prime <- 3 * x**2 + 4 * x
+ x <- x - f / f.prime
+ f <- x***3 + 2 * x**2 - 7
+
> x
[1] 1.428818
```

مثال : ليكن $f(x) = x - \cos(x)$

```
> X0=1
> X=x0
> f=x-cos(x)
> tol=0.000001
> while(abs(f)>tol){f.p=1+sin(x)
+ x=x-f/f.p
+ f=x-cos(x)}
> x
[1] 0.7390852
```

3.4.7 عبارة break

في البرمجة R سلسلة الحلقات الطبيعية يمكن تغييرها باستخدام `break`.



يستخدم عبارة `break` داخل حلقة لوقف تكرار وتدفق حالة الحلقة.

مثال :

```
x <- 1:5
for (val in x) {
  if (val == 3){
    break
```

```

    }
    print(val)
}
[1] 1
[1] 2

```

في المثال اعلاه التكرار لاعداد المتوجه x من 1 الى 5، داخل الحلقة يوجد شرط لكسر الحلقة وهو القيمة تساوي 3 .

4.4.7 عبارة Repeat

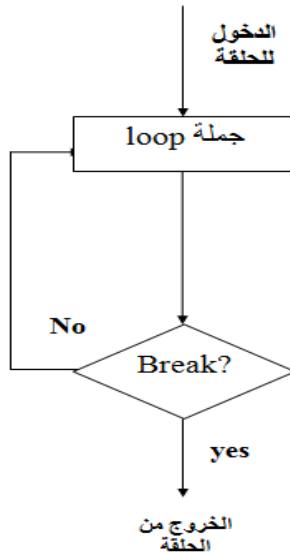
يعطي حلقة لانهائية من التكرار الا في حالة استخدام العبارة break ، وهذا يعني ان هناك حاجة الى عبارة ثانية لاختبار ما اذا كان او لم يكن لإيقاف التكرار و فيما يلي بناء الجملة البرمجية :

- مخطط الجملة Repeat

```

repeat {
  statements
}

```



- مثال

```

x <- 1
repeat {
  print(x)
  x = x+1
}

```

```

if (x == 6){
  break
}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5

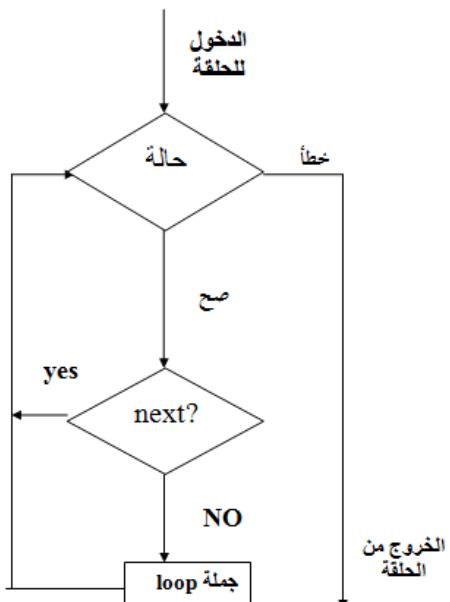
```

في المثال أعلاه، وقد استخدمنا شرطاً للتحقق والخروج من حلقة عندما x يأخذ قيمة 6. وبهذا سيكون الناتج فقط 5-1 مطبوع.

- مثال

```

z <- 0
repeat {
  z <- z + 1
  print(z)
  if(z > 100) break()
}
```



5.4.7 عبارة next

مفید عندما نريد تخطي التكرار الحالي للحلقة دون انهائها.

- مخطط عبارة next -

```
for (val in x) {
  if (val == 3){
    next
  }
  print(val)
}
[1] 1
[1] 2
[1] 4
[1] 5
```

في المثال أعلاه نستخدم العبارة `next` داخل شرط للتحقق مما إذا كانت القيمة تساوي 3 يتوقف التقييم الحالي (لم يتم طباعة قيمة) ولكن تستمر حلقة التكرار مع القيم الأخرى.

5.7 عبارة `switch`

لها تعبير وارجاع قيمة في قائمة اعتماداً على قيمة التعبير، وهذا يعتمد على نوع بيانات التعبير.

- بناء الجملة الاساسي هو :

`switch(statement,item1,item2,item3,...,itemN)`

إذا كان نتيجة التعبير هي رقم ثم تقوم بارجاع العنصر في القائمة مع نفس المؤشر .

- مثال

```
> x <- as.integer(2)
> x
[1] 2
> z = switch(x,1,2,3,4,5)
> z
[1] 2
> x <- 3.5
```

```
> z = switch(x,1,2,3,4,5)
> z
[1] 3
> for(i in c(-1:3, 9)) print(switch(i, 1, 2 , 3, 4))
NULL
NULL
[1] 1
[1] 2
[1] 3
NULL
```

6.7 الدوال Functions

تقوم بوظيفة معينة او محددة ومن فوائد الدوال functions هو تصغير الكود البرمجي والتخلص من تكرار بعض المهام التي تتكرر باستمرار، وكذلك توجد إمكانية إنشاء دوال تؤدي وظيفة خاصة

- بناء الجملة function

```
Function_name<-functions(arg1,arg2,...){  
    Function -body  
}
```

القيمة التي تم ارجاعها بواسطة Function -body هي قيمة return() والذي عادة ما يكون تعبير نهائي غير معين ، مثلاً :

- مثال

```
(1)> square = function(x) { return(x^2)}
> square(5)
[1] 25
> square(1:5)
[1] 1 4 9 16 25
```

```
(2)>a=1:10
>a^2
```

```
>sum(a^2)
>Fct<-function(x){ sum(x^2)}
>Fct(1:5)
```

```
(3)> cube = function(x) x^3
> cube(2)
[1] 8
> cube(1:5)
[1] 1 8 27 64 125
```

```
* (two parameter)
(4)>fct<-function(x,y){
>res<-x^2+y
>res}
>fct(2,1)
```

```
(5)> pow = function(x, y = 2) x^y
> pow(2)
[1] 4
> pow(2, 4)
[1] 16
> pow(y= 4,2)
[1] 16
> pow(y =3, x = 3)
[1] 27
```

```
(6) f.bad <- function(x, y) {
z1 <- 2*x + y
z2 <- x + 2*y
z3 <- 2*x + 2*y
```

```
z4 <- x/y}  
f.bad(1, 2)  
[1] 0.5
```

```
(7) f.good <- function(x, y) {  
  z1 <- 2*x + y  
  z2 <- x + 2*y  
  z3 <- 2*x + 2*y  
  z4 <- x/y  
  return(c(z1, z2, z3, z4))}  
f.good(1, 2)  
$z1:  
[1] 4  
$z2:  
[1] 5  
$z3:  
[1] 6  
$z4:  
[1] 0.5
```

```
(8) f2 <- function(x, y) {  
  z1 <- x + y  
  z2 <- x + 2*y  
  list(z1, z2) }  
f2(2, 5)  
[[1]]:  
[1] 7  
[[2]]:  
[1] 12  
f2(2, 5)[[1]]
```

```
[1] 7  
f2(2, 5)[[2]]  
[2] 12  
f2(2, 5)$z1  
NULL
```

```
(9) f3 <- function(x, y) {  
  z1 <- x + y  
  z2 <- x + 2*y  
  list(result1=z1, result2=z2)}  
f3(2, 5)  
$result1:  
[1] 7  
$result2:  
[1] 12  
f3(2, 5)$result1  
[1] 7  
f3(2, 5)$result2  
[1] 12
```

```
(10) y <- f3(1, 4)  
names(y)  
[1] "result1" "result2"  
y$result2  
[1] 9  
y[[2]]  
[1] 9  
#Using vectors
```

```
(11) v1 <- seq(1:5)
```

```
v1  
[1] 1 2 3 4 5 6  
v2 <- seq(2, 12, 2)  
v2  
[1] 2 4 6 8 10 12  
f3(v1, v2)  
$result1:  
[1] 3 6 9 12 15 18  
$result2:  
[1] 5 10 15 20 25 30  
#Using matrices
```

```
(12) mat1 <- matrix( c(1, 2, 3, 4, 5, 6), ncol=2)  
mat1  
[,1] [,2]  
[1,] 1 4  
[2,] 2 5  
[3,] 3 6  
mat2 <- matrix(c(2, 4, 6, 8, 10, 12), ncol = 2)  
mat2  
[,1] [,2]  
[1,] 2 8  
[2,] 4 10  
[3,] 6 12  
f3(mat1, mat2)  
$result1:  
[,1] [,2]  
[1,] 3 12  
[2,] 6 15  
[3,] 9 18
```

\$result2:

```
[,1] [,2]  
[1,] 5 20  
[2,] 10 25  
[3,] 15 30
```

```
(13) f4 <- function(x=3, y=2) {  
  z1 <- x + y  
  z2 <- x + 2*y  
  list(result1=z1, result2=z2)  
}  
f4()  
$result1:  
[1] 5  
$result2:  
[1] 7  
f4(1, )$result1  
[1] 3  
f4(x=1)$result1  
[1] 3  
f4(, 1)$result1  
[1] 4  
f4(y=1)$result1  
[1] 4  
f4(y = 1, x = 2)$result2  
[1] 4
```

امثلة باستخدام الحلقة

```
(1) for(i in c(1, 3, 6, 9)) {  
  z <- i + 1  
}
```

```
z  
[1] 10  
# باستخدام عبارة الطباعة لرؤيه النتيجه في كل تكرارا  
for(i in 3:5) {  
  z <- i + 1  
  print(z)  
}  
[1] 4  
[1] 5  
[1] 6
```

```
(2) f5 <- function(x) {  
  for(i in 1:x) {  
    y <- i*2  
    print(y) }  
  return(y*2)}  
f5(3)  
[1] 2  
[1] 4  
[1] 6  
[1] 12
```

```
(3) names1 <- c("Dave", "John", "Ann", "Roger", "Bill",  
"Kathy")  
f.names <- function(x) {  
  for(name in x){  
    if(name=="Roger")  
      break  
    print(name)}  
}
```

```
f.names(names1)
[1] "Dave"
[1] "John"
[1] "Ann"
```

باستخدام حلقة While

```
(1) f6 <- function(x) {
      i <- 0
      while(i < x) {
          i <- i+1
          y <- i*2
          print(y) }
      return(y*2)}
f6(3)
[1] 2
[1] 4
[1] 6
[1] 12
```

```
(2) names1 <- c("Dave", "John", "Ann", "Roger", "Bill",
"Kathy")
f.names.while <- function(x) {
    i <- 1
    while( x[i] != "Roger"){
        print(x[i])
        i <- i+1
    }
}
f.names.while(names1)
[1] "Dave"
[1] "John"
```

[1] "Ann"

: repeat باستخدام حلقة

(1) names1 <- c("Dave", "John", "Ann", "Roger", "Bill",
"Kathy")

```
f.names.repeat <- function(x) {
  i <- 1
  repeat {
    print(x[i])
    i <- i+1
    if(x[i] == "Roger")
      break
  }
}
```

f.names.repeat(names1)

[1] "Dave"

[1] "John"

[1] "Ann"

(2) random.sample1 <- function(epsilon) {

i <- 0

repeat {

i = i+1

mean.test <- abs(mean(rnorm(100)))

if (mean.test < epsilon)

break

}

list(mean=mean.test, number.iterations=i)

}

random.sample1(0.0001)

\$mean:

```
[1] 0.00001373388
$number.iterations:
[1] 6033
y.plot(x, y)
```

* دالة لدمج متغيرين او اكثر

مثال:

```
> a="Hello"
> b="world !"
> a+b
Error in a + b : non-numeric argument to binary operator
> '%fct%'<-function(x,y){paste(x,y,seq="")}
> a%fct%b
[1] "Hello world ! "
> concat<-function(x,y){paste(x,y,seq="")}
> concat(a,b)
[1] "Hello world ! "
```

7.7 دوال اخرى

(1) دالة paste() : تستخدم للجمع بين عدد من السلسل في R

مثال :

```
> a='ff'
> b='gg'
> c='hh'
> paste(a,b,c)
[1] "ff gg hh"
```

(2) دالة format() : تستخدم لتنسيق الارقام والسلسل بنمط معين

مثال :

```
> format(23.123456789, digits = 5)
[1] "23.123"
> format(9)
[1] "9"
```

(3) دالة nchar() : تستخدم لحساب عدد الرموز وبضمنها الفراغ في السلسلة

مثال :

```
> nchar("university of Basrah")
[1] 20
```

(4) دالة Print() : تستخدم لطباعة النتائج على الشاشة

> print(3+4)

مثال :

مثال :

```
> for(i in 1:3) print(1:i)
[1] 1
[1] 1 2
[1] 1 2 3
```

مثال : ممكن الحصول على الناتج بشكل مبسط و مباشر بدون عبارة الطباعة كما يلي:

```
> t1 <- round(abs(rt(200, df = 1.8)))
> t2 <- round(abs(rt(200, df = 1.4)))
> table(t1, t2) # simple
    t2
t1   0   1   2   3   4   5   6   7   8   10  11  20  21
  0 27 28  9  3  2  1  1  0  0  1  1  0  1
  1 26 26  7  4  3  1  0  3  0  0  2  1  0
  2  9 13  5  0  3  1  0  1  0  0  0  0  0
  3  2  4  1  0  0  1  0  0  0  0  0  0  0
  4  0  1  0  0  0  0  0  0  0  0  0  0  0
  5  1  0  0  0  0  0  0  0  0  0  0  0  0
  6  1  0  0  1  0  0  0  0  0  0  0  0  0
  7  0  0  2  0  0  0  0  0  0  0  0  0  0
  8  0  0  0  0  0  0  0  0  1  0  0  0  0
  9  0  1  0  0  0  0  0  0  0  0  0  0  0
 11  1  0  0  0  0  0  0  0  0  0  0  0  0
 13  1  0  0  0  0  0  0  0  0  0  0  0  0
 15  1  0  0  0  0  0  0  0  0  0  0  0  0
 16  1  0  0  0  0  0  0  0  0  0  0  0  0
 29  1  0  0  0  0  0  0  0  0  0  0  0  0
```

اما في حالة التحكم في شكل النتائج للحصول على مظهر اجمل لقرائتها كما يلي :

```
> print(table(t1, t2), zero.print = ".")
      t2
t1   0  1  2  3  4  5  6  7  8 10 11 20 21
  0 27 28  9  3  2  1  1 . .  1  1 .  1
  1 26 26  7  4  3  1 .  3 . .  2  1 .
  2  9 13  5 .  3  1 .  1 . . . .
  3  2  4  1 . .  1 . . . . .
  4 .  1 . . . . . . . .
  5  1 . . . . . . . .
  6  1 . .  1 . . . . . .
  7 . .  2 . . . . . . .
  8 . . . . . . . .  1 . .
  9 .  1 . . . . . . .
 11  1 . . . . . . .
 13  1 . . . . . . .
 15  1 . . . . . . .
 16  1 . . . . . . .
 29  1 . . . . . . .
```

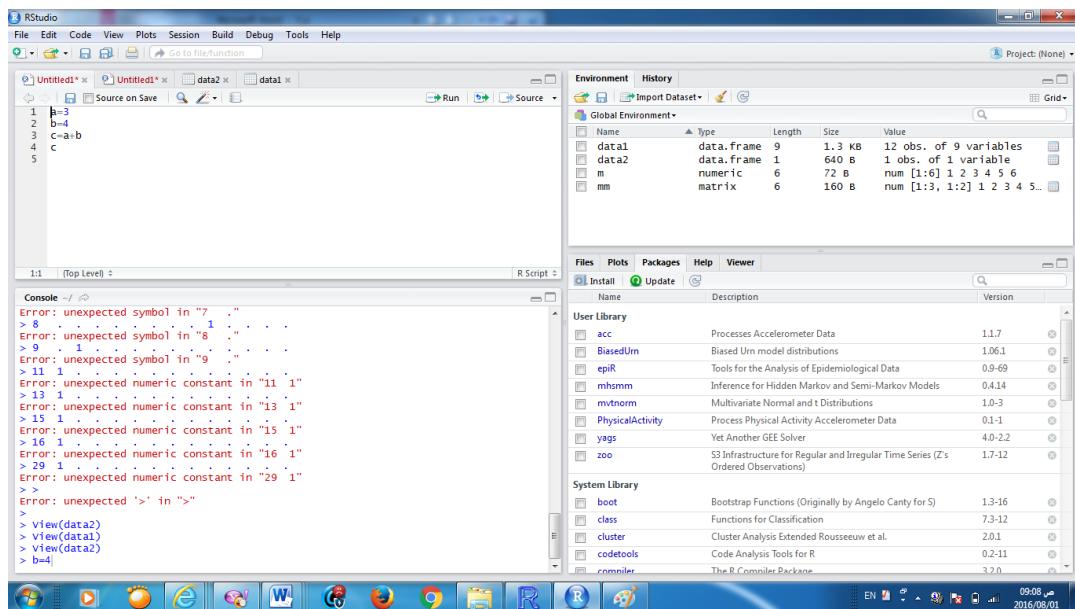
RStudio 8.7

يعتبر RStudio واجهة بديلة لـ R حيث يمكن تثبيت RStudio كتطبيق سطح المكتب أو كتطبيق الخادم الذي هو في متناول الآخرين عبر الإنترنت. فهو مفتوح المصدر (أي مجاناً) ومتاح من الموقع <http://www.rstudio.org>.

RStudio يقدم واجهة احسن لبرنامج R فهو بيئة التطوير المتكاملة (/ IDE) (Integrated development environment) وتطبيق البرمجيات التي توفر مرافق شاملة لمبرمجي الكمبيوتر لتطوير البرمجيات. وIDE يتكون عادة من محرر شفرة المصدر، وبناء أدوات التشغيل الآلي ومصحح.

يمكنك استخدام R دون RStudio باستخدام أي محرر لكتابة البرامج النصية. ولكن RStudio في حد ذاته ليس من المفيد جداً دون R. وهو البرنامج المكتوب في لغة البرمجة C++.

RStudio هو الواجهة لـ R. والذي يجعل استخدام R أجمل كثيراً كما يضيف بعض المكونات السهلة الاستخدام حيث يتيح لك الكثير من الحزم بسهولة. كما يوفر الوصول إلى R في متصفح الويب ، وفي حالة تشغيل برنامج RStudio تظهر الشاشة كما في شكل (1-7) وهي مقسمة إلى أربعة نوافذ وان في كل نافذة عدة تبويبات:



شكل(1-7) شاشة RStudio

وفيما يلي شرح موجز لواجهة المستخدم الرسومية Graphic Users Interface /GUI والتي قد تتغير حسب اعداد او اصدار البرنامج :

اولاً: نافذة المحرر The Editor

ويكون موقعها المربع اليسير في الاعلى ، وفيه يتم كتابة تعبير او دالة او حزمة بلغة R نريد الاحتفاظ بها في وقت لاحق ، اي يمكن نقلها الى نافذة وحدة التحكم لتنفيذها وايجاد الناتج بشكل اسرع ، وهذا يكون عن طريق ضغطة واحدة لمفتاحي . Ctrl+Enter

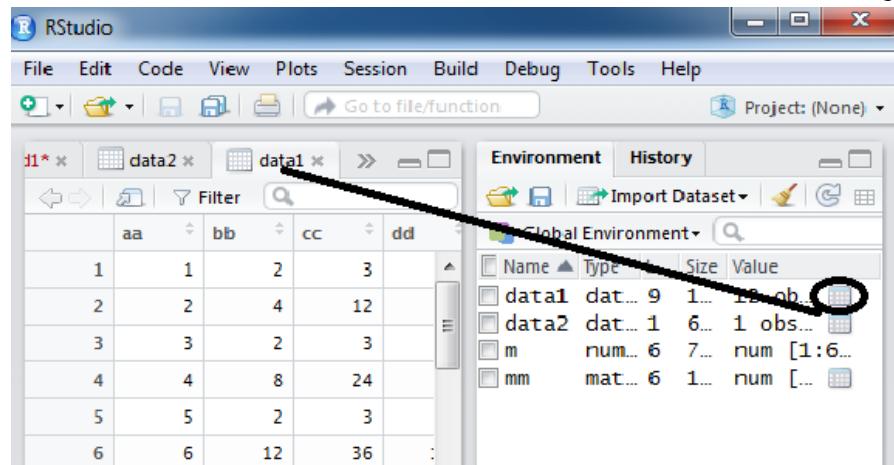
ثانياً : نافذة وحدة التحكم Console

ويكون موقعها المربع اليسير في الاسفل وهي النافذة الشبيهة بنافذة برنامج R حيث يتم فيها كتابة الاوامر وتنفيذ التعليمات البرمجية ومختلف العمليات في R ومشاهدة الناتج .

ثالثاً : نافذة بيئة العمل Environment والتاريخ History

وموقعها المربع اليمين في الاعلى ويحتوي على تبويبين هما :

(1) تبويب بيئة العمل Environment : في هذا الجزء من الشاشة يتم خزن اي كائن او دالة او اي رمز قمنا بتعريفه في وحدة التحكم كما ويتم سرد مجموعة بيانات او استيراد قواعد بيانات وجعلها متاحة على الفور في وحدة التحكم وكما في الشكل(7-2) تظهر مربعات منقطة مقابل اسم كل كائن او متغير في بيئة العمل ، وعند النقر عليها يمكنك الاطلاع على البيانات في جزء الشاشة المجاور الى اليسار.



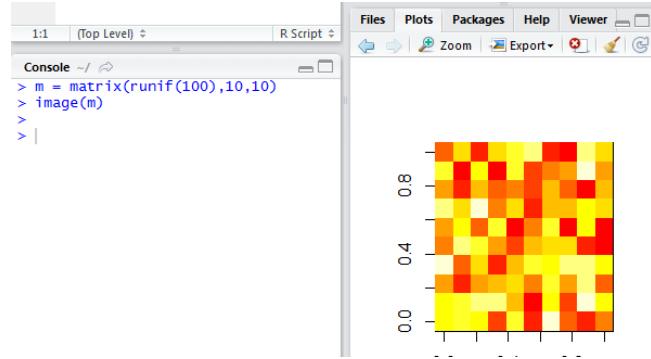
شكل(7-2) تبويب بيئة العمل

(2) تبويب التاريخ History يحتفظ بسجل لجميع الأوامر السابقة. كما أنه يساعد عند اختبار عمليات التشغيل. حيث يمكنك حفظ إما اللائحة بأكملها أو اختيار الأوامر التي تريد وإرسالها إلى برنامج نصي R لمتابعة عملك.

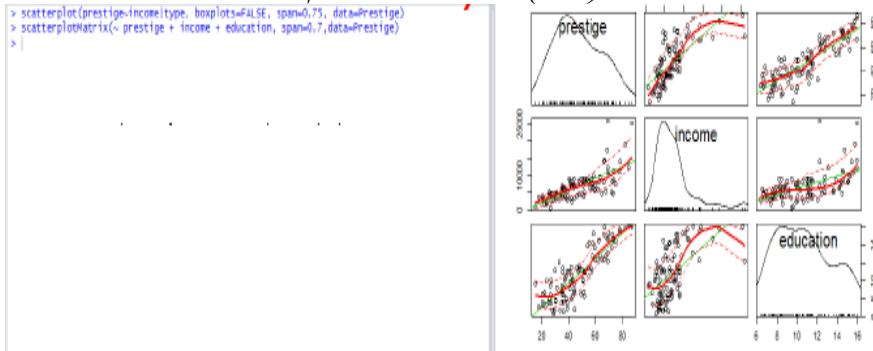
رابعاً : نافذة Files, plots, packages , help, Viewer موقعها الى اليمين في الاسفل وتحتوي على خمس تبويبات وكما يلي :

(1) تبويب الملفات Files وهو غني عن التعريف ، حيث يظهر معه شريط يحتوي ايقونات Rename و Delete و New Folder وغيرها ، كما ويعرض اسماء الملفات

(2) تبويب Plots لعرض الرسوم البيانية التي انشأت في R والتي قد تكون رسم بياني واحد او اكثراً وكما في الشكلين (3-7) و(4-7) التاليين :



شكل(3-7) تبويب Plot لرسم واحد



شكل(4-7) تبويب Plot لعدة رسوم

كما يمكن حفظ الرسم البياني بضرب ايقونة Export وفيها الحفظ كصورة او ملف تنسيق PDF او نسخه الى Clipboard .

(3) تبويب حزم Packages

تعرض قائمة من الحزم الإضافية المدرجة في تركيب RStudio. والتي يتم تحميلها ايضاً إلى R، ووصف موجز بجوار كل حزمة. يمكنك أيضاً تثبيت الوظائف الإضافية الأخرى عن طريق النقر على ايقونة "Install Package" .

(4) تبويب المساعدة Help

يتيح لك البحث في دليل مساعدة واسعة بشكل لا يصدق وسوف تفتح تلقائياً عند الاتصال على تعليمات حول أمر تم الاستفهام عنه في وحدة التحكم بالابتعاز مثل factor

(5) تبويب عارض Viewer

هو أساساً RStudio في المتصفح ، حيث يمكنك تطوير تطبيقات الويب مع R وحتى إطلاق شبكة استضافة محلية لتطبيقات داخلها.

9.7 الفرق بين R و RStudio

R هي لغة البرمجة أو بيئة البرمجيات كما يسمونها عن الحوسبة العلمية والرسومات.

RStudio يعتبر فعال لإنجاز الأمور بسرعة ويقدم واجهة احسن لبرنامج R فهو بيئة التطوير المتكاملة لـ R (Integrated development /IDE) وتطبيق البرمجيات التي توفر مراافق شاملة لمبرمجي الكمبيوتر لتطوير البرمجيات. IDE يتكون عادة من محرر شفارة المصدر، وبناء أدوات التشغيل الآلي ومصحح.

يمكنك استخدام R دون RStudio باستخدام أي محرر لكتابة البرامج النصية. ولكن RStudio في حد ذاته ليس من المفيد جدا دون R .

RStudio هو الواجهة الأمامية لـ R. والذي يجعل استخدام R أجمل وأكثر فعالية، ويتتيح لك استخدام الكثير من الحزم بسهولة .

اضافة الى هذا فان RStudio يضم العديد من النوافذ المفتوحة في نفس الوقت وادوات لتنظيم العمل في المشاريع وغيرها. واخيراً فإن كلاهما برامج بُنيت للحوسبة الاحصائية ،ولها مصدر مفتوح مجاني .



اسئلة الفصل السابع

س1: اذكر بناء الجملة والمخطط (ان وجد) لكل عبارة من العبارات التالية:

(2) عبارة if...else	(1) عبارة IF
(4) عبارة for	(3) عبارة if...else المتداخلة
(6) عبارة break	(5) عبارة while
(8) عبارة next	(7) عبارة Repeat
(10) عبارة function	(9) عبارة switch

س2: اكتب برنامج لكل مما يلي :

(1) يختبر الارقام من 1-10 ويطبعها بحيث يطبع الارقام الاكبر من 6 والصغر من 9 اصفاراً؟

(2) يدخل كل مرة عدد ويختبره ليطبع فردي او زوجي ؟

(3) يدخل مجموعة من الاعداد ويختبرها ليطبع لكل منها فردي او زوجي ؟

(4) يحسب الارقام الزوجية في المتجه التالي :

23 ، 2 ، 98 ، 45 ، 6 ، 30 ؟

(5) يحسب متتالية فيبوناتشي للعدد 20 ؟

(6) اذا كان $x=5$ اختبر واطبع الاعداد الاقل من 15 ؟

(7) دالة لدمج متغيرين او اكثر ؟

س3: وضح استخدام الاوامر التالية في R ، مع ذكر مثال ؟

print() ، nchar() ، format() ، paste()

س4: ما هي RStudio ؟ ووضح مكونات واجهة المستخدم الرسمية ؟

س5: اذكر الفرق بين R و RStudio ؟



الفصل الثامن

الحزم في R

1.8 مقدمة

2.8 كيفية استخدام الحزم

3.8 تعريف الحزم في R

4.8 تحميل حزم البيانات

5.8 تحميل حزم البرامج

Packages 6.8

Load package... 1.6.8

Set CRAN mirror... 2.6.8

Select repositories... 3.6.8

Install package(s)... 4.6.8

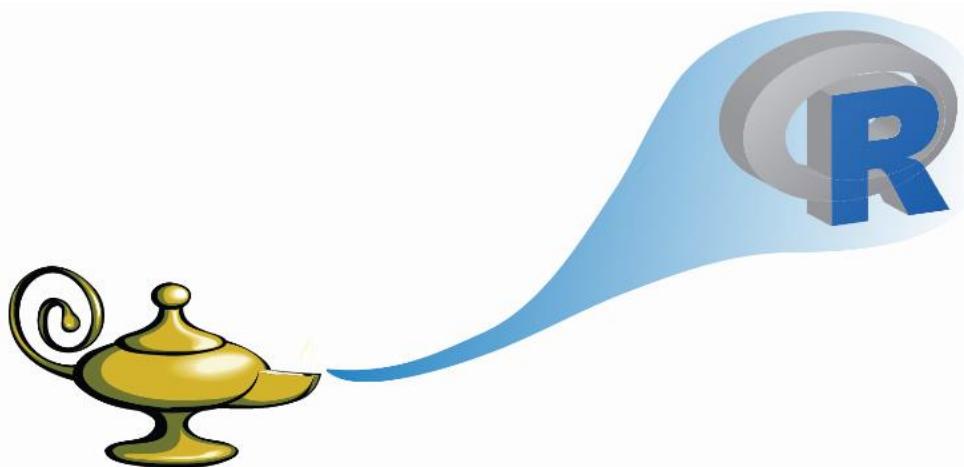
Update packages... 5.6.8

Install package(s) from local zip files... 6.6.8

7.8 اضافة حزم لبرنامج R على حاسوبك

8.8 مستودعات لحزم اضافية

اسئلة الفصل الثامن



1.8 مقدمة

في R، الوحدة الأساسية للمشاركة هي الحزمة. وتمثل الحزمة باقة من التعليمات البرمجية والبيانات والوثائق، والاختبارات، سهلة للمشاركة مع الآخرين. فقد اضاف الاف الخبراء في مجال العلوم الاحصائية وظائف على R في شكل حزم والتي هي ايضاً متاحة بحرية على صفحات الانترنت ، واعتبارا من بداية عام 2015، كان هناك أكثر من 6000 حزمة متوفرة على شبكة الارشيف الشامل CRAN أو كرا او CRAN، وان توفر مجموعة كبيرة من الحزم من بيانات وتطبيقات تمتد عبر العديد من التخصصات المختلفة في الدراسة هو احد الاسباب التي جعلت لغة R ناجحة للغاية ، حيث ان هناك احتمالات بأن شخصا ما قد حل بالفعل المشكلة التي نعمل عليها، ويمكنك الاستفادة من عملهم من خلال تحميل تلك الحزمة .

2.8 كيفية استخدام الحزم

قبل ان تتمكن من استخدام الحزمة ، عليك اولاً تثبيتها ، فبعض الحزم هي اساسية ويتم تثبيتها تلقائياً عند تثبيت حزم R ، والحزم الاخرى لن تأتي مع تركيب R لذلك فهي تحتاج الى تثبيت .

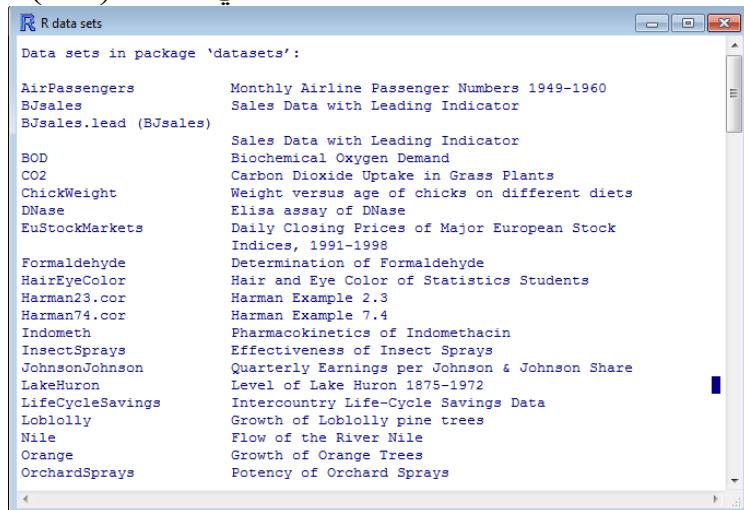
- توجد ثلاث دوال مهمة في استخدام الحزم (ليكن اسمها مثلاً : x) وهي :
- (1) تحميل الحزمة من داخل R: >install.packages("x")
 - (2) استخدام الحزمة في R : >library("x")
 - (3) للحصول على المساعدة يكون اما بـ >?x او

3.8 تعريف الحزم في R

بالإضافة إلى الميزات الأساسية يوفر R العديد من الحزم والتي هي مجموعة من الدوال او الرموز او البيانات في لغة البرمجة R ، اي ملفات اضافية جاهزة لحل مشاكل معقدة في R. وان النسخة المحمولة لـ R تحتوي على مجموعة محددة من الحزم التي تمكنا من قراءة وتحليل البيانات والقيام بالاختبارات الاحصائية.

4.8 تحميل حزم البيانات

قد تكون الحزم المتوفرة في R هي بيانات مخزونة في R ولاستدعاءها نكتب الاياعز : >`data()` فتظهر مجموعات البيانات في حزمة : (1-8) ضمن نافذة مجموعات بيانات R وكما في الشكل



شكل(1-8) اياعز (`data()`)

ولادرارج وعرض اي من حزم البيانات المتوفرة ، نكتب في سطر الاوامر اسم الحزمة .

مثال : لتكن حزمة البيانات BOD ، نكتب >`BOD` وكما يلي :

> BOD		
	Time	demand
1	1	8.3
2	2	10.3
3	3	19.0
4	4	16.0
5	5	15.6
6	7	19.8

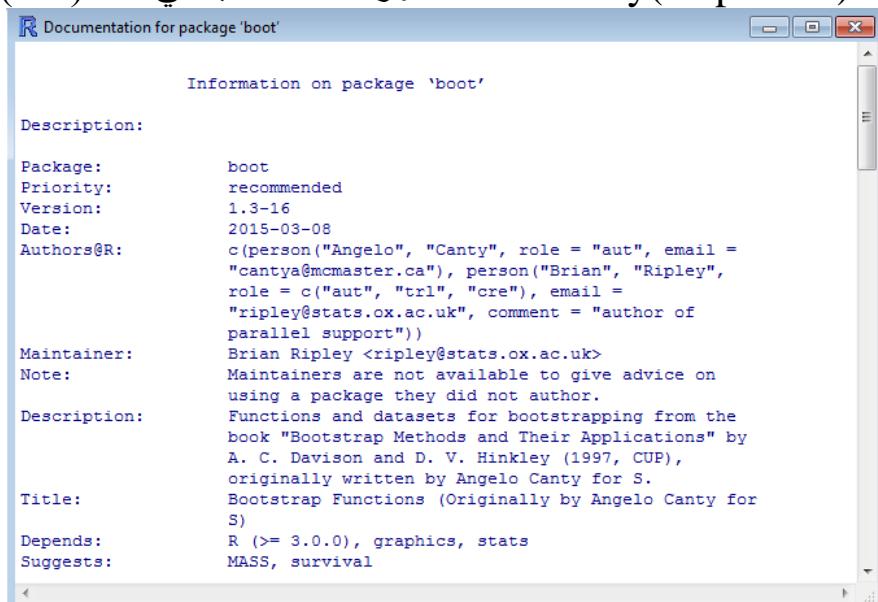
5.8 تحميل حزم البرامج

يتضمن برنامج R برامج اضافية لها امكانية اوسع في الحل ، وللحصول على قائمة بجميع الحزم المثبتة في R نكتب الامر : >`library()` فتظهر النافذة في الشكل(2-8) والتي قد تختلف تبعا للإعدادات المحلية في جهاز الكمبيوتر الخاص بك.



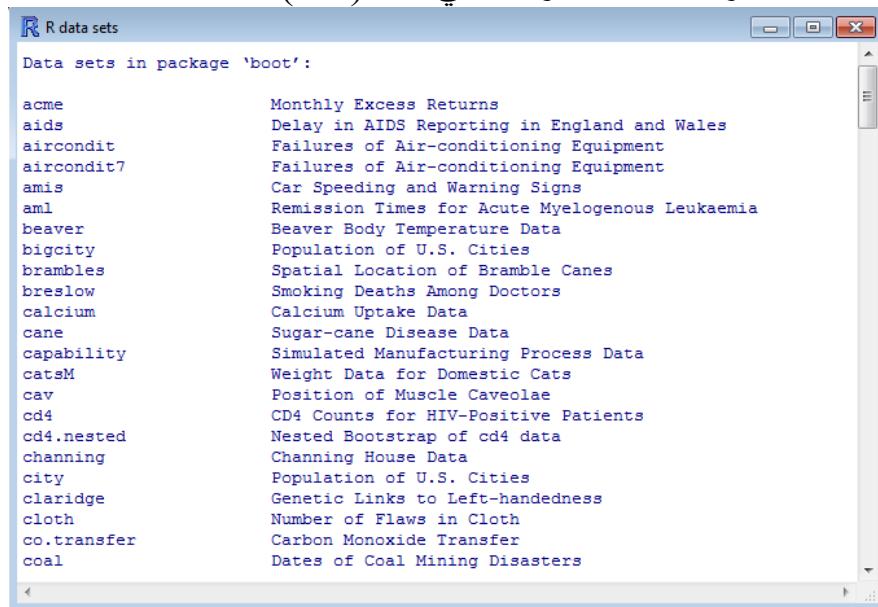
شكل (2-8) الامر `library()`

ولتحميل اي حزمة من الحزم المدرجة في النافذة اعلاه ولتكن "boot" نستخدم الاياعاز : >`library(boot)` وللحصول على معلومات حول الحزمة المطلوبة وهي في مثالنا "boot" ، نكتب الاياعاز : >`library(help=boot)` فقطر النافذة التالية في شكل (3-8) :



شكل (3-8) `library(help=boot)`

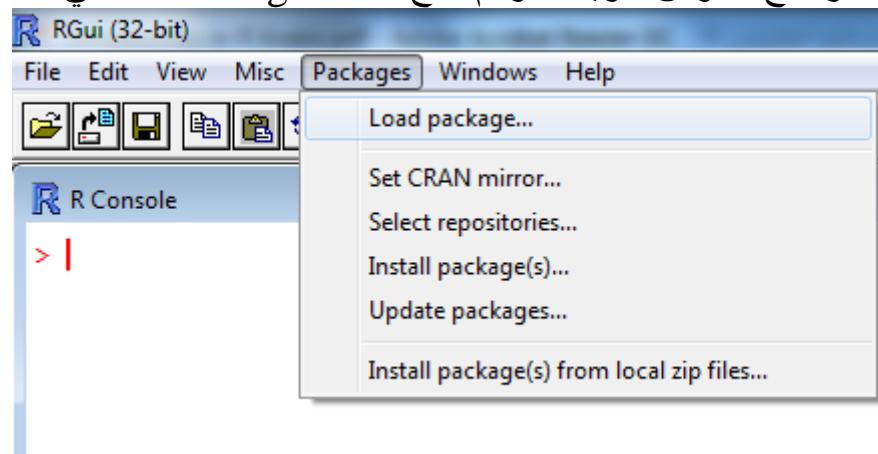
ولحصول على بيانات تلك الحزمة نستخدم ايعاز البيانات : `>data()` فقط يظهر البيانات الخاصة بحزمة "boot" وكما في شكل(4-8) :



شكل(4-8) بيانات الحزمة boot

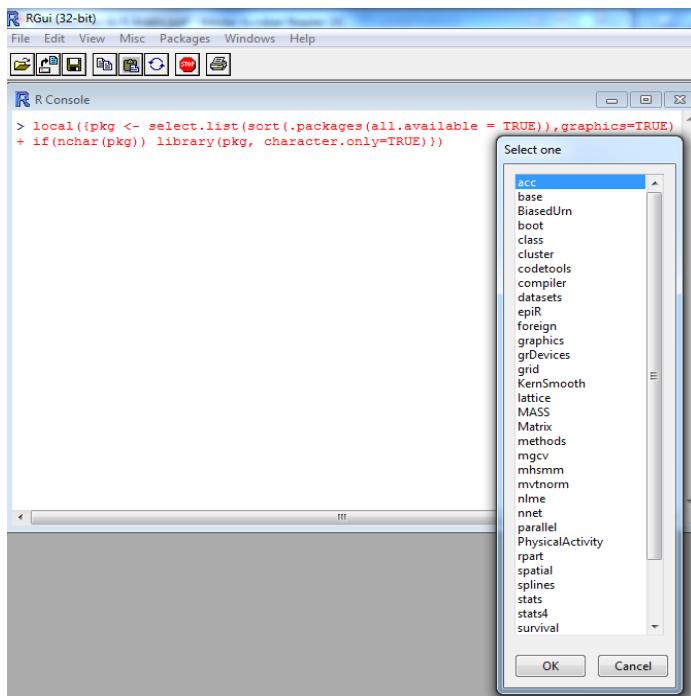
6.8 قائمة Packages

يمكن ادراج الحزم المتوفرة على حاسوبك في لغة البرمجة R وذلك من الشاشة الافتتاحية لبرنامج R ومن شريط القوائم نفتح قائمة Packages كما في شكل(5-8)



شكل(5-8) قائمة Packages

حيث تحتوي على 6 اوامر ، وفيما يلي توضيح استخدام كل منها ؟



Load package... 1.6.8

نختار...
فتشير نافذة (Select one)
تعرض جميع الحزم المتوفرة
وكما في شكل(6-8):

شكل (6-8) نافذة Select one

Set CRAN mirror... 2.6.8

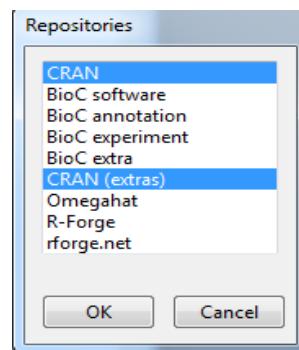
عند اختياره تظهر قائمة باسم "CRAN mirror" ، كما في شكل (7-8) ليتم فيها اختيار اسم البلد الذي سيتم تحميل الحزمة منها.



شكل (7-8) نافذة CRAN mirror

Select repositories... 3.6.8

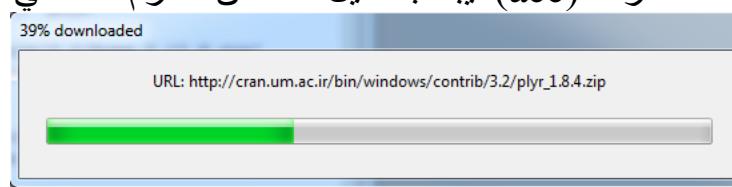
تعرض أسماء الحزم الموجودة والتي تحتاج لتجديد وكما في الشكل(8-8) :



شكل(8-8) نافذة *Repositories*

Install package(s)... 4.6.8

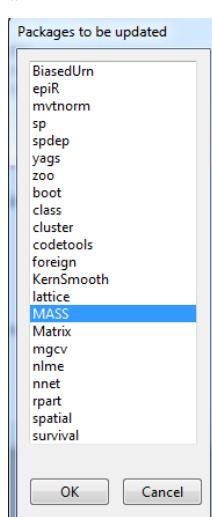
عند اختيارها تظهر القائمة CRAN mirror كما في اعلاه و عند اختيار الجهة او البلد التي نريد تحميل الحزم منها ، تظهر قائمة اخرى Packages تشمل اسماء الحزم ، نختار مثلاً حزمة (acc) ليبدأ بتحميل عدد من الحزم ، كما في شكل(9-8).



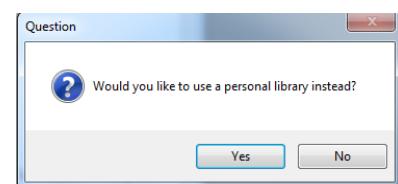
شكل(9-8) تحميل عدد من الحزم

Update packages... 5.6.8

عند اختياره تظهر ايضاً قائمة CRAN mirrorr و عند اختيار الجهة التي نريد تحديث الحزم التابعة لها ، تظهر قائمة Packages to be updated كما في شكل(10-8) و نختار مثلاً MASS ثم MASS و نختار مثلاً الحزمة Question و كما في شكل(11-8) :



شكل(10-8) قائمة
Packages to be updated



شكل(11-8) مربع حوار Question

وتظهر ايضاً في نافذة R Console العبارات التالية :

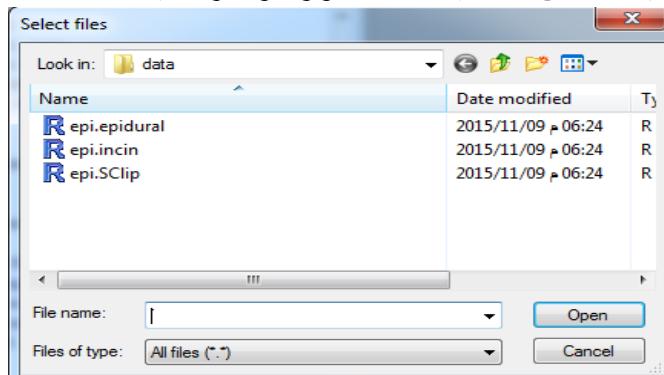
```
> update.packages(ask='graphics',checkBuilt=TRUE)
Warning: package 'MASS' in library 'C:/Program Files/R/R-3.2.0/library' will no$ 
Warning in install.packages(update[instlib == 1, "Package"], 1, repos = repos, $ 
  'lib = "C:/Program Files/R/R-3.2.0/library"' is not writable
trying URL 'http://ftp.uni-sofia.bg/CRAN/bin/windows/contrib/3.2/boot_1.3-18.zip'
Content type 'application/zip' length 592031 bytes (578 KB)
downloaded 578 KB

package 'boot' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\dell\AppData\Local\Temp\RtmpI1btbn\downloaded_packages
```

Install package(s) from local zip files... 6.6.8

وهنا يظهر مربع حوار Select files ، كما في شكل (12-8) وفيه اسماء فضاءات عمل مخزونة في R وتاريخ تعديلها ، يمكن فتح احدها في فضاء العمل الحالي . وقد نحتاج لحزم اضافية لبعض العمليات المتقدمة والبرمجيات المعقدة .



شكل (12-8) مربع حوار

7.8 اضافة حزم لبرنامج R على حاسوبك

تسمح لغة البرمجة R بتطوير سريع لأدوات جديدة حسب الطلب ، وتمثل تلك الأدوات بشكل حزم Packages جاهزة للاستخدام وسهلة التقاسم مع الآخرين ، لتنصيب حزمة يجب توفر شيئين :

- أولاً : الاتصال بالأنترنت لتحميل الحزمة
- ثانياً : اسم الحزمة المراد تنصيبها

وان القوة والمرنة في لغة R هو في توفر العدد الهائل من حزم البيانات والحزام لمعظم التطبيقات الاحصائية الادارية والمالية والاقتصادية والعلمية والتكنولوجية فتجد

الاحتمال الكبير لحل المشكلة التي تعمل عليها ، وبذلك يمكنك الاستفادة من عملهم عن طريق تحميل الحزمة الخاصة بهم. حيث أن أغلبية الباحثين بمجال البرمجة الحيوية يقومون بنشر خوارزمياتهم بلغة R في مستودع الحزم (bioconductor) ، ويمكن تطبيق تعليمات برمجية بسيطة للمشاركة مع الآخرين في إرسال الحزمة الخاصة بك .
نحتاج أولاً لتنصيب الحزم إلى تعيين الدليل لتخزين الحزم وفيما يلي الأوامر ليتم استخدامها لفحص والتحقق منها واستخدام حزم R :
أولاً: التحقق من الحزم المتوفرة:

- الدالة library() والتي تم توضيحها سابقاً

- الدالة search() للحصول على كافة الحزم التي تم تحميلها حالياً في بيئة R :

```
> search()
[1] ".GlobalEnv"           "package:stats"      "package:graphics"
[4] "package:grDevices"    "package:utils"       "package:datasets"
[7] "package:methods"       "Autoloads"         "package:base"
```

- تنصيب حزمة جديدة : هناك طريقة لإضافة حزم R الجديدة :

1) تنصيب مباشر من كرا CRAN :

الأمر التالي يمثل الحصول على حزم مباشرة من صفحة ويب كرا وتنصيب الحزمة في بيئة R. قد يتطلب منك اختيار الأقرب و المناسب لموقعك.

الصيغة العامة : install.packages("package Name")

مثال : ليكن اسم الحزمة yags

```
install.packages("yags")
```

2) تنصيب حزمة يدوياً :

ننتقل إلى برنامج R لتحميل الحزمة التي تحتاجها ، وتحفظ الحزمة كملف Zip في موقع مناسب في النظام المحلي ، ثم نستخدم الامر التالي لتنصيب الحزمة في بيئة R :

```
install.packages(file_name_with_path, repos = NULL, type =
"source")
```

مثال : لتحميل الحزمة " yags "

```
install.packages("C:/yags_3.98-1.3.zip", repos = NULL, type =
"source")
```

8.8 مستودعات لحزم اضافية

وفيما يلي اسماء مخازن تحتوي على حزم اضافية ومجموعة ابحاث مفتوحة المصدر مطورة من قبل باحثين .

(1) (Comprehensive R Archive Network) CRAN

وهو موقع لحزم تخدم في عدة مجالات مثل المعلومات الحيوية ويسرد مجموعة واسعة من المهام في العلوم الاجتماعية والاحصاء وعلم الوراثة وارتفاع اداء الحاسوبات واللة التعلم والتصوير الطبي كما وُتستخدم من قبل ادارة الاغذية والعقاقير باعتبارها مناسبة لتفسير البيانات في البحوث السريرية ولتحميل حزمة من موقع كرا

نختار قائمة الحزم packages في شريط القوائم وكما في شكل(5-8)
من قائمة packages نختار install packages... --- يظهر مربع حوار mirror لاختيار حزمة واحدة ثم يظهر مربع حوار packages ويمكن اختيار اكثر من حزمة .

ولتحميل الحزم من الانترنت لها الموقع التالي، كما في شكل(13-8) :

<http://cran.r-project.org/>



شكل(13-8) موقع cran

Bioconductor (2)

هو موقع مفتوح المصدر يوفر حزم R لتحليل بيانات الجينوم عالية الانتاجية الا انه يقبل الحزم التي تخضع لشروط الموقع من اضافة لتوضيح الحزمة والدوال مع امثلة وبترتيب خاص .. لهذا نجد اغلب الباحثي يتعاملوا مع الموقع CRAN لسرعة نشر برامجهم . يتم تحميل الحزم بخطوتين :

- اولاً : تحميل موقع bioconductor :

> source("http://bioconductor.org/biocLite.R")

- ثانياً : تحميل الحزم باستعمال الدالة > biocLite("Packages Name")



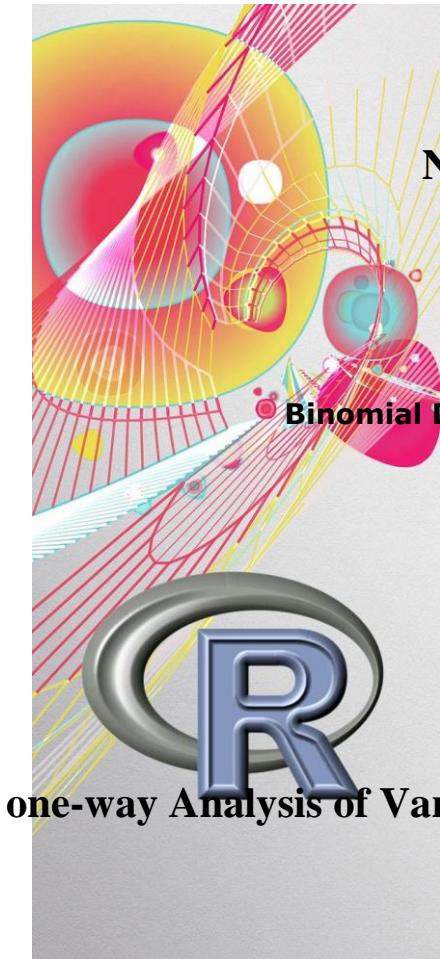
اسئلة الفصل الثامن

- س1: اذكر ثلاث دوالي مهمة لاستخدام الحزم في R ؟
- س2: ما السبب الرئيسي الذي جعل لغة R هي اللغة الناجحة ؟
- س3: ما هو تعريف الحزم في R ؟ وما انواعها ؟
- س4: كيف يتم تحميل حزم البيانات في R ؟ اذكر مثال ؟
- س5: كيف يتم تحميل حزم البرامج في R ؟ اذكر مثال ؟
- س6: اين توجد قائمة packages ؟ وما الاوامر المدرجة فيها ؟
- س7: اذكر اسماء مخازن تحتوي على حزم اضافية ومجموعة ابحاث ؟
- س8: عرف كل من : CRAN ، Bioconductor ، مع توضيح لطريقة تحميل الحزم ؟
- س9: اكتب الدالة في R لكل ما يلي :
 - (1) الحصول على كافة الحزم التي تم تحميلها حالياً ؟
 - (2) تحميل الحزم من داخل R ؟
 - (3) الحصول على المساعدة ؟
 - (4) تحميل حزم البيانات ؟
 - (5) تحميل حزم البرامج ؟
 - (6) تحميل موقع bioconductor ؟



الفصل التاسع

تطبيقات احصائية



- 1.9 مقدمة
- 2.9 المتوسط Mean والوسيط Median والمنوال Mode
- 3.9 الانحدار الخطي Linear Regression
- 4.9 دالة التنبؤ predict()
- 5.9 الانحدار المتعدد Multiple Regression
- 6.9 التوزيع الطبيعي Normal Distribution
 - 1.6.9 دالة dnrom()
 - 2.6.9 دالة pnorm()
 - 3.6.9 دالة qnorm()
 - 4.6.9 دالة rnorm()
- 7.9 التوزيع الثنائي (ذي الحدين) Binomial Distribution
 - 1.7.9 دالة dbinom()
 - 2.7.9 دالة pbinom()
 - 3.7.9 دالة qbinom()
 - 4.7.9 دالة rbinom()
- 8.9 انحدار بواسون Poisson Regression
- 9.9 التغاير Covariance
- 10.9 الارتباط Correlation
- 11.9 تحليل التباين الاحادي one-way Analysis of Variance ANOVA
- 12.9 اختبار t t test
- 13.9 توزيع f f distribution
- اسئلة الفصل التاسع

1.9 مقدمة

التطبيقات الاحصائية باستخدام برنامج R هي الجانب الذي تظهر فيه بوضوح اهمية هذه اللغة في اغلب واهم العمليات الاحصائية وذلك بتهيئة اوامر سهلة وخلالية من التعقيد من حيث كتابة الصيغة او العباره البرمجية للحصول على نتائج واضحة ومفصلة

يقدم هذا الفصل عدد من التطبيقات الاحصائية بشرح مختصر لتقنيه الاحصائية وبناء الجملة الاساسية لحساب الدوال مع وصف للمعايير المستخدمة في الدالة ، اضافة الى الامثلة لكل تطبيق .

2.9 المتوسط Mean والوسيط Median والمنوال

1.2.9 المتوسط

ويتم حساب المتوسط عن طريق ايجاد مجموع القيم وتقسيمها على عدد القيم في سلسلة بيانات .

الدالة mean() تُستخدم لحساب المتوسط في R وبناء الجملة الأساسية لحساب المتوسط في R هي :

`mean(x, trim = 0, na.rm = FALSE, ...)`

وفيما يلي وصف المعايير المستخدمة:
x هو متوجه المدخلات.

trim: يستخدم للتخلص من بعض القيم من كل نهاية للمتجه بعد فرزها .

na.rm: يستخدم لإزالة القيم المفقودة من المتوجه .

مثال:

`>x<-c(12,5,9,34,-3,-7,-14,28)`

`>mean(x)`

`[1] 8`

مثال

عند استخدام المعلمة trim لحساب المتوسط حيث trim=0.3 يعني سيتم اسقاط 3 قيم من طرفي المتوجه بعد ترتيبه:

`>x<-c(12,7,3,4,2,18,2,54,-21,8,-5)`

`>mean(x,trim=0.3)`

بعد الترتيب يكون : $>x<-c(-21, -5, 2, 3, 4.2, 7, 8, 12, 18, 54)$
 نزيل من المتوجه القيم (-21, 5, 2) من اليسار و(18, 12) فيصبح الناتج :
 [1] 5.55
 مثل

في حالة وجود قيم مفقودة تطبيق الخيار NA لإسقاط القيم المفقودة نستخدم na.rm = TRUE. وهو يعني إزالة القيم NA.

```
>x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)
```

```
>mean(x,na.rm=TRUE)
```

```
[1] 8.22
```

اما في حالة استخدام الدالة mean() فيكون الناتج :

```
>mean(x)
```

```
[1] NA
```

2.2.9 الوسيط Median

هو الرقم الذي يفصل النصف الأعلى من العينة أو المجتمع عن النصف الأقل بحيث يتساوى على طرفه عدد القيم بعد ترتيبها تصاعدياً. فإذا كان عدد هذه القيم فردياً فالوسيط هو الرقم النصفي الذي يقسم هذه القيم، أما إذا كان عدد القيم زوجياً فالوسيط هو الوسط الحسابي لمجموع الرقمان الوسيطين.
 بناء الجملة الأساسية لحساب الوسيط في R هو :

```
median(x,na.rm=FALSE)
```

وفيما يلي وصف المعايير المستخدمة :
 x هو متوجه المدخلات.

na.rm يستخدم لإزالة القيم المفقودة من المتوجه.

مثال

```
>x<-c(-5,12,-8,12,7,3,4.2,18,2,54)
```

```
> median(x)
```

```
[1] 5.6
```

3.2.9 Mode

هو القيمة الأكثر تكراراً في مجموعة من البيانات . في R لا توجد دالة لحساب المنوال ، لذلك ننشئ دالة المنوال حيث تأخذ المتوجه كمدخل وتعطي قيمة المنوال كأخرج .

مثال

```
> getmode <- function(v) {
+uniqv <- unique(v)
+uniqv[which.max(tabulate(match(v, uniqv)))]}
> v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)
> result <- getmode(v)
> print(result)
[1] 2
> charv <- c("o","it","the","it","it")
<result <- getmode(charv)
<print(result)
[1] "it"
```

انشاء دالة

انشاء متوجه رقمي

انشاء متوجه حرفي

3.9 الانحدار الخطى Linear Regression

تحليل الانحدار هو أداة إحصائية تستخدم على نطاق واسع جدا لإنشاء نموذج العلاقة بين متغيرين. يسمى أحد المتغيرات (متغير تابع) والآخر (متغير مستقل) .

في الانحدار الخطى ترتبط هذين المتغيرين من خلال معادلة، حيث الأس لكل من المتغيرين هي 1. رياضيا تمثل العلاقة بخط مستقيم عند رسمها بيانيأً . والعلاقة

الغير خطية يكون الأس لأى متغير لا يساوي 1 حيث ينشئ منحنى .

المعادلة الرياضية العامة للانحدار الخطى هي: $y = ax + b$

و فيما يلى وصف المعايير المستخدمة :

y : هو المتغير المعتمد

x : هو المتغير المستقل

a و b الثوابت التي تسمى معاملات.

1.3.9 خطوات لإنشاء الانحدار

في مثال بسيط عن الانحدار هو توقيع وزن الشخص عندما يعرف طول قامته ؟

لذلك نحن بحاجة إلى العلاقة بين الطول والوزن للشخص. وفيما يلي خطوات لإنشاء العلاقة :

- 1) تُنفذ تجربة بجمع عينة من قيم الطول والوزن
- 2) إنشاء نموذج العلاقة باستخدام دالة `lm()` في R.
- 3) ايجاد معاملات من النموذج اعلاه وانشاء معادلة رياضية باستخدامها
- 4) احصل على ملخص لنموذج العلاقة لمعرفة متوسط الخطأ في التنبؤ.
- 5) للتنبؤ بالوزن لأشخاص جدد نستخدم دالة `predict()` في R

2.3.9 ادخال البيانات

وفيما يلي عينة من البيانات ولتكن قيم الطول :

131، 151، 131، 174، 138، 186، 128، 136، 179، 136، 163، 152، 131

وقيم الوزن : 63، 81، 56، 91، 47، 57، 76، 72، 62، 48

3.3.9 دالة `lm()`

بناء الجملة الاساسية لدالة `lm()` في الانحدار الخطي :

`lm(formula,data)`

وفيما يلي وصف المعايير المستخدمة :

- `formula` : هي رمز عرض العلاقة بين x و y.

- `data` : هي متغيرات والتي سيتم تطبيق الصيغة.

4.3.9 إنشاء نموذج العلاقة والحصول على المعاملات

```
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> lm(y~x)
```

```
Call:
lm(formula = y ~ x)
```

```
Coefficients:
(Intercept)          x
-38.4551        0.6746
```

5.3.9 ملخص العلاقة

تستخدم دالة `summary()` للحصول على النتائج التالية :

```
> summary(lm(y~x))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.3002 -1.6629  0.0412  1.8944  3.9775 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -38.45509   8.04901 -4.778  0.00139 **  
x             0.67461   0.05191 12.997 1.16e-06 *** 
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491 
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

4.9 دالة التنبؤ predict()

بناء الجملة الأساسية للتنبؤ predict() في الانحدار الخطي :

`predict(object , newdata)`

وفيما يلي وصف المعايير المستخدمة :

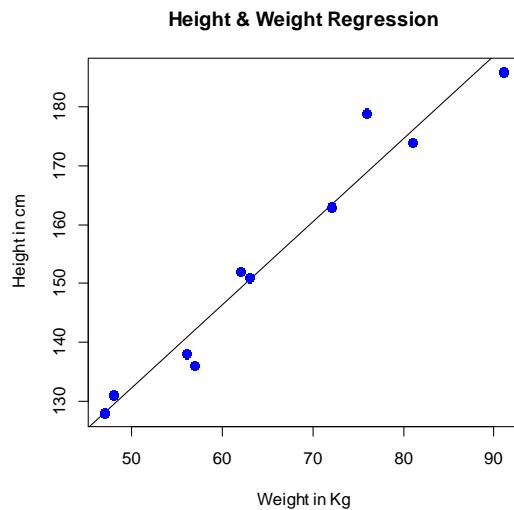
`object` : هو الصيغة التي تم إنشاؤها بالفعل باستخدام دالة lm()

`newdata` : هو متوجه تحتوي على قيمة جديدة للمتغير المتتبى

```
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> relation<-lm(y~x)
> a=data.frame(x=170)
> result=predict(relation,a)
> result
1
76.22869
```

ولرسم الانحدار بيانيًّا، انظر شكل (1-9)

```
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> plot(y,x,col = "blue",main = "Height & Weight Regression",
+ abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",
+ ylab = "Height in cm")
```



شكل(9-1) رسم الانحدار

5.9 الانحدار المتعدد Multiple Regression

ويكون فيه المتغير Y يعتمد على أكثر من متغير مستقل ،المعادلة الرياضية العامة للانحدار المتعدد هي :

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

وفيما يلى وصف المعايير المستخدمة :

هو متغير معتمد : y

معلمات a, b_1, b_2, \dots, b_n هی a,b1,b2...bn

x_1, x_2, \dots, x_n هی متغیرات مستقلة

بناء الصيغة الاساسية لدالة Im في الانحدار المتعدد هي كما يلى:

```
lm(y ~ x1+x2+x3...,data)
```

مثال

لتكن مجموعة البيانات هي mtcars المتاحة في البيئة R

```

> lm(mpg~disp+hp+wt, data = input)

Call:
lm(formula = mpg ~ disp + hp + wt, data = input)

Coefficients:
(Intercept)          disp            hp            wt
 37.105505     -0.000937    -0.031157    -3.800891

> coef(model)[1]
(Intercept)
37.10551
> coef(model)[2]
      disp
-0.0009370091
> coef(model)[3]
      hp
-0.03115655
> coef(model)[4]
      wt
-3.800891

```

6.9 التوزيع الطبيعي Normal Distribution

في مجموعة عشوائية من البيانات من مصادر مستقلة، لوحظ بشكل عام أن توزيع البيانات أمر طبيعي. وهو ما يعني، عند الرسم البياني مع قيمة المتغير في محور x الأفقي وعدد من القيم في المحور الرأسي نحصل على منحنى بشكل جرس. وسط المنحنى يمثل متوسط مجموعة البيانات. في الرسم البياني، خمسين في المئة من القيم تقع في يسار الوسط والخمسون في المئة الأخرى تقع على يمين الرسم البياني. ويشار إلى هذا النحو بالتوزيع الطبيعي في الإحصاءات.

R لها أربعة دوال لتوليد التوزيع الطبيعي، وفي أدناه وصف لها:

`dnorm(x, mean, sd)`
`pnorm(x, mean, sd)`
`qnorm(p, mean, sd)`
`rnorm(n, mean, sd)`

وفيما يلي وصف المعايير المستخدمة في الدوال المذكورة أعلاه :

x هو متوجه من الأرقام.

p هو متوجه من الاحتمالات.

n عدد من الملاحظات (حجم العينة).

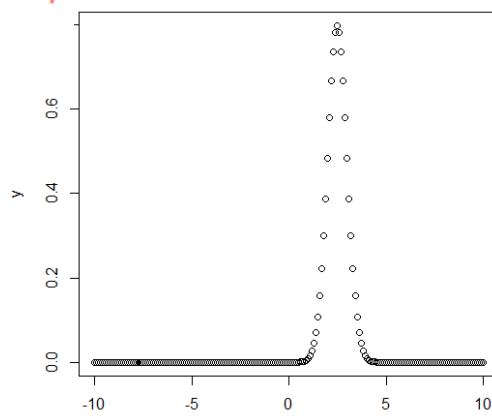
Mean : القيمة المتوسطة لعينة البيانات ، القيمة الافتراضية هي صفر.

Sd : الانحراف المعياري. ولها قيمة افتراضية هي 1.

1.6.9 دالة dnorm()

هذه الدالة تعطي ذروة التوزيع الاحتمالي في كل نقطة لوسيلة معينة، والانحراف المعياري. انظر شكل (2-9):

```
x <- seq(-10, 10, by = .1)
y <- dnorm(x, mean = 2.5, sd = 0.5)
plot(x,y)
```



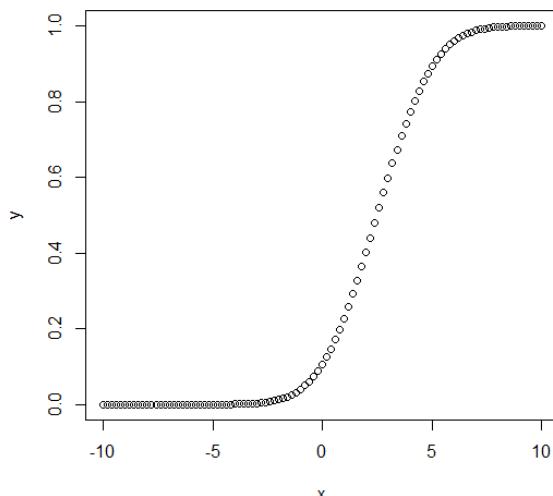
شكل (2-9) دالة $dnorm$

2.6.9 دالة pnorm()

هذه الدالة تعطي احتمال وجود رقم عشوائي موزع بشكل طبيعي لتكون أقل من قيمة عدد معين. ويسمى أيضا "دالة التوزيع التراكمي"، انظر شكل (3-9)

مثال

```
> x <- seq(-10, 10, by = .2)
> y <- pnorm(x, mean = 2.5, sd = 2)
> plot(x,y)
```



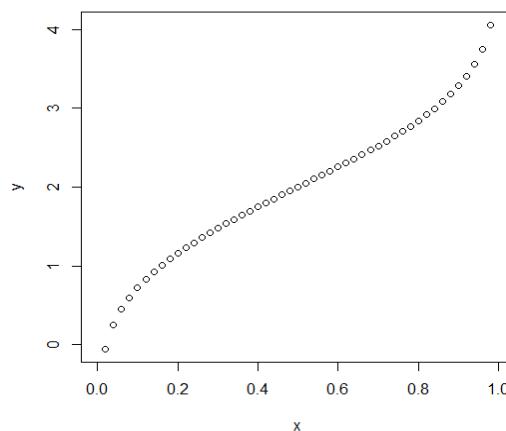
شكل(3-9) دالة $pnorm$

3.6.9 دالة $qnorm()$

تحتاج هذه الدالة قيمة احتمال وتعطي رقم يطابق قيمة احتمال قيمة تراكمية، انظر شكل(4-9).

مثال

```
> x <- seq(0, 1, by = 0.02)
> y <- qnorm(x, mean = 2, sd = 1)
> plot(x,y)
```



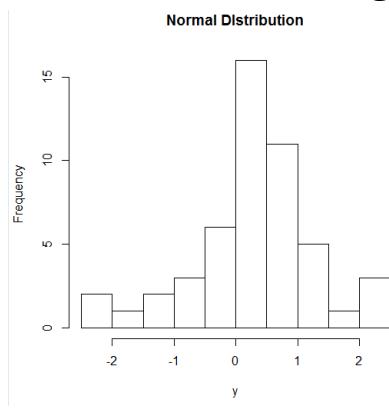
شكل(4-9) دالة $qnorm$

4.6.9 دالة $rnorm()$

يتم استخدام هذه الدالة لتوليد أرقام عشوائية ذات توزيع طبيعي. فإنه يأخذ حجم العينة كمدخل ويولد العديد من الأرقام العشوائية، انظر شكل(5-9).

```
> y=rnorm(50)
> hist(y,main = "Normal DIstribution")
`
```

مثال

شكل(5-9) دالة *rnorm*

7.9 التوزيع الثنائي (ذي الحدين) Binomial Distribution

نموذج التوزيع ذي الحدين يتعامل مع العثور على ناتج من الاختبار احتمال نجاح هذا الحدث والآخر هو الفشل ، مثل رمي قطعة نقود ، والاسئلة التي تعتمد الاجابة بنعم او لا .

في R توجد أربع دوال لتوليد التوزيع ذي الحدين. وهي كما في ادناه :

dbinom(x, size, prob)
pbinom(x, size, prob)
qbinom(p, size, prob)
rbinom(n, size, prob)

و فيما يلي وصف المعايير المستخدمة :

x هو متوجه من الأرقام.

p غير متوجه من الاحتمالات.

n عدد من الملاحظات.

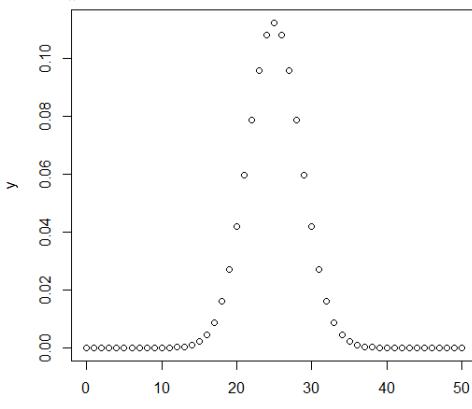
Size عدد الاختبارات

Prob هو احتمال نجاح كل تجربة.

1.7.9 دالة dbinom()

هذه الدالة تعطي احتمال توزيع الكثافة في كل نقطة.

مثال : انشاء عينة من 50 رقم والتي تترايد بمقدار 1
 $>x<-seq(0,50,by=1)$
 $> y<-dbinom(x,50,0.5)$ لانشاء توزيع ذو حدین :
 $>plot(x,y)$ الرسم البياني لهذه العينة :
 وعند تنفيذ التعليمات اعلاه ، نحصل على النتيجة كما في شكل (6-9) :



شكل (6-9) دالة $dbinom$

2.7.9 دالة pbinom()

مثلاً، هذه الدالة تعطي الاحتمال التراكمي للحدث. فهي قيمة واحدة تمثل الاحتمالات.

```
# احتمال الحصول على 26 أو أقل لظهور صورة h من 51 قذفة لعملة واحدة.
>x<-rbinom(26,51,0.5)
>print(x)
[1] 0.610116
```

دالة qbinom()

تحتاج هذه الدالة قيمة احتمال ويعطي رقم يطابق القيمة التراكمية.
مثال : كم عدد ظهور الصورة h لاحتمال 0.25 عند قذف العملة المعدنية 51 مرات.

```
>x<-qbinom(0.25,51,1/2)
>print(x)
[1] 23
```

4.7.9 rbinom() دالة

هذه الدالة تولد العدد المطلوب من القيم العشوائية من احتمال معين من عينة معينة.
مثال : جد 8 قيم عشوائية من عينة من 150 مع احتمال 0.4؟

```
>rbinom(8,150,.4)
[1] 58 54 64 51 51 67 56 48
```

8.9 انحدار بواسون Poisson Regression

يتضمن نماذج انحدار تستخدم في نماذج العد وجدائل الاحتمالات، على سبيل المثال، عدد من المواليد أو عدد من الانتصارات في سلسلة مباراة لكرة القدم. أيضاً قيم المتغيرات المعتمدة يتبع توزيع بواسون.
المعادلة الرياضية العامة لانحدار بواسون هي :

$$\log(y) = a + b_1x_1 + b_2x_2 + b_nx_n.....$$

حيث ان :

y هو متغير معتمد

a و b هي معاملات رقمية.

x هو متغير توقع.

(glm() هي الدالة التي تستخدم في انشاء نموذج انحدار وان بناء الجملة الأساسية للدالة هو :

`glm(formula,data,family)`

وفيما يلي وصف المعايير المستخدمة في أعلاه :

formula: هي رمز عرض العلاقة بين المتغيرات.

data: هي مجموعة بيانات تعطي قيم هذه المتغيرات.

family : هي كائن في R لتحديد تفاصيل هذا النموذج. قيمتها " بواسون " للانحدار اللوجستي.

مثال

لدينا مجموعة بيانات "warpbreaks" الذي يصنف تأثير نوع الصوف (A أو B) والتوتر (منخفضة، متوسطة أو عالية) على عدد من فواصل تشوه في النول. دعونا النظر "فواصل" كمتغير استجابة وهو عدد من عدد من التخفيضات. يتمأخذ الصوف "نوع" و "التوتر" كمتغيرات تنبؤ. فيكون نموذج الانحدار كما يلي :

`glm(formula=breaks ~ wool+tension,data=warpbreaks,family =poisson)`

ويمكن تنفيذ دالة summary للحصول على النتائج التالية :

```
> summary(glm(formula=breaks ~ wool+tension,data=warpbreaks,family =poisson))

Call:
glm(formula = breaks ~ wool + tension, family = poisson, data = warpbreaks)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-3.6871 -1.6503 -0.4269  1.1902  4.2616 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) 3.69196   0.04541  81.302 < 2e-16 ***
woolB       -0.20599   0.05157  -3.994 6.49e-05 ***
tensionM    -0.32132   0.06027  -5.332 9.73e-08 ***
tensionH    -0.51849   0.06396  -8.107 5.21e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 297.37 on 53 degrees of freedom
Residual deviance: 210.39 on 50 degrees of freedom
AIC: 493.06

Number of Fisher Scoring iterations: 4
```

في دالة summary() نبحث عن القيمة p في العمود الأخير إلى أن تكون أقل من 0.05 للنظر في تأثير متغير توقع على متغير معتمد كما شهدت wooltype بوجود نوع من التوتر M و H يكون لها تأثير على عدد من فواصل .

9.9 التغاير Covariance

هو مقياس لكمية تغيير متحولين مع بعضهما ، وتكون قيمته موجبة ، عندما يتغير المتحولان فوق القيمة المتوقعة معاً ، وسالبة عندما يكون احد المتحولين فوق القيمة المتوقعة والآخر انى منها .

والصيغة الاحصائية للتغاير هي :

$$\text{Cov}(X,Y)E((X-\mu)(Y-\nu))$$

حيث : X ، Y متحولين عشوائيين
قيمة X المتوقعة $E(X)=\mu$

$E(Y) = \nu$ قيمة Y المتوقعة و تستخدم في R الدالة `cov()` لحساب التغاير والتبالين : هو حالة خاصة يكون فيها المتحولان متساويان

10.9 الارتباط Correlation

ان هذه العلاقة هي تقنية احصائية يمكن أن تظهر كيف ترتبط بقوة أزواج من المتغيرات. على سبيل المثال، الطول والوزن على أن هذه العلاقة غير واضحة إلى حد ، يتم احتساب الارتباط بما يعرف باسم معامل الارتباط ، التي لديها قيمة يجب أن تقع بين -1 و +1. بالضبط في منتصف الطريق. وهذا صحيح خاصة إذا كنت قد وصفت منتصف نقطة من النطاق الخاص بك (مثلاً تقدير "الجيد" هو نصف الطريق بين "ممتاز" و "مقبول").

في R تُستخدم الدالة `cor()` لحساب معامل الارتباط ، والصيغة البسطة هي :

`Cor(x,use=,method=)`

حيث ان :

X : مصفوفة او اطار بيانات

Use: يحدد التعامل مع البيانات المفقودة ، الخيارات في اعطاء طريقة لحساب التغايرات ويكون اختصاراً واحداً مما يلي :

"everything", "all.obs", "complete.obs", "na.or.complete", or
"pairwise.complete.obs
pearson, spearman : طريقة تحديد نوع الارتباط. الخيارات هي : method
or kendall

1.10.9 التغيير والارتباط بين المتغيرات الرقمية :

في اطار البيانات `mtcars` ، نستخدم الخيار `complete.obs` لحذف البيانات المفقودة:

```
>cor(mtcars, use="complete.obs", method="kendall")
>cov(mtcars, use="complete.obs")
```

2.10.9 ارتباط مصفوفة :

من `mtcars` ، حيث ان `mpg`, `cyl`, `disp` هي الصور اعمدة `hp`, `drat`, `wt`

```
>x<-mtcars[1:3]
```

	hp	drat	wt
mpg	-0.7761684	0.6811719	-0.8676594
cyl	0.8324475	-0.6999381	0.7824958
disp	0.7909486	-0.7102139	0.8879799

```
>y<-mtcars[4:6]
>cor(x,y)
```

3.10.9 الارتباط بين متغيرين

العبارة `if()` غالباً ما تُستخدم داخل دالات معروفة من قبل المستخدم ، وفيما يلي مثال نموذجي :

- مثال : الارتباط بين متغيرين من الارقام غالباً ما يُحسب باستخدام دالة `cor()`. يمكن ان نضيف رسم مبغي لبيانات وكما يلي :

```
> corplot <- function(x, y, plotit) {
+ if(plotit == TRUE) plot(x, y)
+ cor(x, y)
+ }
```

يمكن تطبيق هذه الدالة لمتغيرين بدون رسم بالكتابة كما يلي:

```
> corplot(c(2, 5, 7), c(5, 6, 8), FALSE)
[1] 0.953821
```

مثال

```
x <- rnorm(10, sd=5, mean=20)
> y <- 2.5*x - 1.0 + rnorm(10, sd=9, mean=0)
> cor(x,y)
[1] 0.8667313
```

11.9 تحليل التباين الاحادي one-way Analysis of Variance ANOVA هي دراسة تأثير عامل واحد له عدة مستويات مختلفة ، وتكرر التجربة في كل مستوى لعدد من المرات. وان الدالة `aov()` هي دالة تحليل التباين في R ، حيث يستخدم R بسهولة في تحليل ANOVA لتحليل التصاميم في الحزم الاحصائية ، وان الباحثون يستخدمون ANOVA في عدة طرق :

- (1) اتجاه واحد :
 - (2) تصميم مجموعة عشوائياً
 - (3) اتجاهين
 - (4) تحليل التغير
- ```
>fit<-aov(y~A,data=mydataframe)
>fit<-aov(y~A+B,data=mydataframe)
>fit<-aov(y~A+B+A:B,data=mydataframe)
>fit<-aov(y~A*B,data=mydataframe)
>fit<-aov(y~A+x,data=mydataframe)
```

(5) واحد ضمن عامل

&gt;fit&lt;-aov(y~A+Error(Subject/A),data=mydataframe

**t test t 12.9**في R تستخدم دالة `t.test()` لانتاج مجموعة متنوعة من اختبارات `t`(1) اختبار `t` لمجموعتين مستقلة : `>t.test(y~x)`حيث `y` رقمي و `x` عامل ثانوي(2) اختبار `t` لمجموعتين مستقلة : `>t.test(y1,y2)`حيث `y1` و `y2` ارقام(3) مزدوج اختبار `t` : `>t.test(y1,y2,paired=TRUE)`(4) اختبار `t` عينة واحدة : `>t.test(y,mu=3)`

وفيما يلي توضيح للمعلمات اعلاه :

`x` : متوجه رقمي لقيم البيانات (غير فارغة)`y` : متوجه رقمي لقيم البيانات (غير فارغة) وهو اختياري`paired` : منطقية تشير إلى ما إذا كنت ترغب في اقتراح اختبار `t`.`mu` : رقم يشير إلى القيمة الحقيقية للمتوسط (أو الاختلاف في الوسائل إذا كنت تتفذ

اختبار اثنين من العينة).

امثلة

```
> t.test(1:10, y = c(7:20))

Welch Two Sample t-test

data: 1:10 and c(7:20)
t = -5.4349, df = 21.982, p-value = 1.855e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-11.052802 -4.947198
sample estimates:
mean of x mean of y
5.5 13.5
```

```
> t.test(1:10, y = c(7:20, 200))

 Welch Two Sample t-test

data: 1:10 and c(7:20, 200)
t = -1.6329, df = 14.165, p-value = 0.1245
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-47.242900 6.376233
sample estimates:
mean of x mean of y
5.50000 25.93333
```

```
> plot(extra ~ group, data = sleep)
> with(sleep, t.test(extra[group == 1], extra[group == 2]))

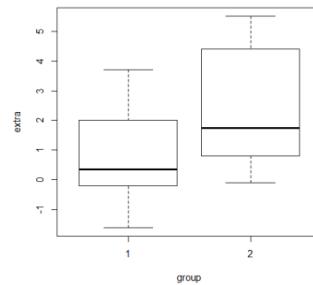
 Welch Two Sample t-test

data: extra[group == 1] and extra[group == 2]
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.3654832 0.2054832
sample estimates:
mean of x mean of y
0.75 2.33

> t.test(extra ~ group, data = sleep)

 Welch Two Sample t-test

data: extra by group
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.3654832 0.2054832
sample estimates:
mean in group 1 mean in group 2
0.75 2.33
```



شكل 7-9 (t-test)

### 13.9 توزيع f distribution

هو التوزيع الاحتمالي المرتبط بالمتغيران العشوائيان المستقلان  $x_1, x_2$  ولعينة عشوائية  $n_1, n_2$  على التوالي وبدرجات حرية  $V_1 = n_1 - 1$  للبسط و  $V_2 = n_2 - 1$  للمقام ، ويستخدم توزيع f لاختبار تساوي مجتمعين . ودالة  $qf()$  هي المستخدمة في R لحساب توزيع f.

مثال : جد المئوي 95 (المئوي هو القيمة التي يقطع 95 هنا المئة الأولى من قيم البيانات عندما يتم فرزها في ترتيب تصاعدي). لتوزيع F مع (5، 2) درجات الحرية ؟

```
> qf(.95, df1=5, df2=2)
[1] 19.29641
```

### 1.13.9 اختبار f بين اثنين من الفروق

ينفذ اختبار F لمقارنة الفروق من عينتين من السكان وكما يلي حيث نستخدم دالة var.test() في R :

```
>var.test(x, y, ratio = 1,
 alternative = c("two.sided", "less", "greater"),
 conf.level = 0.95, ...)
```

لثة "صيغة"

>var.test(formula, data, subset, na.action, ...)

وفيما يلي توضيح للمعلمات اعلاه:

x,y متوجهات رقمية من قيم البيانات او تركيب اجسام النموذج الخطى

ratio : نسبة افترض من فروق السكان x و y .

alternative : سلسلة أحرف تحديد الفرضية البديلة، يجب أن يكون واحداً من ("الافتراضي")، "greater" or "less". يمكن فقط تحديد الحرف الأولي.

conf.level : مستوى الثقة للفاصل الثقة التي تم إرجاعها.

formula : صيغة صيغة النموذج  $LHS \sim RHS$  حيث LHS هو متغير رقمي وإعطاء قيم البيانات و RHS عملاً مع اثنين من المستويات.

Data : مصفوفة اختياري أو إطار بيانات (أو ما شابه ذلك) التي تحتوي على المتغيرات في الصيغة formula افتراضيا يتمأخذ المتغيرات من البيئة (formula).

Subset : متوجه اختياري يحدد مجموعة فرعية من الملاحظات التي سيتم استخدامها.

na.action : دالة تدل على ما ينبغي أن يحدث عندما تحتوي على بيانات تحتوي. افترضات إلى ("getOption ("na.action")

مثال  

```
> x <- rnorm(50, mean = 0, sd = 2)
> y <- rnorm(30, mean = 1, sd = 1)
> var.test(x, y)
```

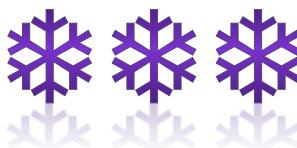
```
F test to compare two variances

data: x and y
F = 3.974, num df = 49, denom df = 29, p-value = 0.0001656
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
1.996651 7.476829
sample estimates:
ratio of variances
3.974042

> var.test(lm(x ~ 1), lm(y ~ 1))

F test to compare two variances

data: lm(x ~ 1) and lm(y ~ 1)
F = 3.974, num df = 49, denom df = 29, p-value = 0.0001656
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
1.996651 7.476829
sample estimates:
ratio of variances
3.974042
```



## اسئلة الفصل التاسع

س1: اذكر الدالة في لغة البرمجة R لكل مما يلي :

|                           |                    |             |              |
|---------------------------|--------------------|-------------|--------------|
| (10) التغاير              | (7) التنبؤ         | (4) المنوال | (1) المتوسط  |
| (11) الانحدار المعتمد     | (8) الانحدار الخطي | (5) الوسيط  | (2) اختبار t |
| (9) تحليل التباين الاحادي | f                  | (6) توزيع f | (3) اختبار t |

س2: ما الغرض من استخدام كل من الدوال التالية :

|              |              |             |             |
|--------------|--------------|-------------|-------------|
| qbinom() (7) | dbinom() (5) | qnorm() (3) | dnorm() (1) |
| rbinom() (8) | pbinom() (6) | rnorm() (4) | pnorm() (2) |

س3: ما الغرض من استخدام كل من المعلمات التالية :

|                 |             |            |           |
|-----------------|-------------|------------|-----------|
| prob(10)        | newdata (7) | formula(4) | trim (1)  |
| family (11)     | size (8)    | object (5) | na.rm (2) |
| alternative(12) | Ratio (9)   | mu (6)     | paired(3) |

س4: اكتب التعليمية البرمجية في لغة R لحساب المتوسط للقيم التالية حيث  

$$(8,23,-68,15,3,19,-42,0,-5,77) \quad ? \quad trim=0.2$$

س5: اكتب التعليمية البرمجية في لغة R لحساب كل من المتوسط والوسيط للقيم  
 التالية :  $(4,-9,NA,27,-33,56,-3,NA) \quad ?$

س6: انشئ متغير رقمي واخر حرفي وكتابة التعليمات البرمجية بلغة R لحساب  
 المنوال لكل منهم ؟

س7: اكتب التعليمات البرمجية لحساب الانحدار الخطي وملخص العلاقة للبيانات  
 التالية بين الكميات المعروضة من السلعة (a) وسعر السلعة (b) ؟

$32, 21, 11, 7, 4, 6 : a$

1، 3، 5، 7، 2 : b

س8: لديك المتجهان التاليان :  $n=123,167,135, 189, 151,144, 156, 188$

$m=54,34,76,42, 51,72, 37,19$

اكتب التعليمات البرمجية بلغة R لدالة التنبؤ والرسم البياني ؟

س9: اعط مثال لحساب الانحدار المتعدد لأي بيانات متاحة في R ؟

س10: جد 5 قيم عشوائية من عينة من 80 مع احتمال 0.3 ؟

س11: طبق دالتين من دوال توليد التوزيع الطبيعي والرسم البياني على سلسلة البيانات a حيث :  $a=\text{seq}(-7,9,by=.2)$  ؟

س12: اذكر دوال لتوليد التوزيع ذي الحدين ؟

س13: ليكن  $m,n$  متجهين من الارقام ، اكتب دالة لحساب الارتباط ورسم بياني للمتجهين بين المتجهين ؟

س14: يستخدم الباحثون ANOVA في عدة طرق ، اذكرها واكتب بناء الجملة الاساسية لكل الدالة ؟

س5: لدراسة علاقة الاستهلاك المحلي (y) بانتاج (x) لمادة الاسفلت(بالمليون برميل) خلال عدة سنوات ، اخذنا عشر قراءات تقريبية كما يلى :

|   |    |    |    |    |   |   |   |   |   |   |
|---|----|----|----|----|---|---|---|---|---|---|
| y | 6  | 8  | 9  | 8  | 7 | 6 | 5 | 6 | 5 | 5 |
| x | 10 | 13 | 15 | 14 | 9 | 7 | 6 | 6 | 5 | 5 |

اكتب برنامج في R يحسب معاذلة الانحدار الخطي البسيط وتوقع قيمة الاستهلاك عندما يصل الانتاج 16.000.000 برميل؟









## فهرست الاشكال

| الصفحة |    | اسم الشكل                         | ت    |
|--------|----|-----------------------------------|------|
| 10     |    | <i>MRAN</i>                       | 1-1  |
| 11     |    | مجالات عمل الحوسبة الاحصائية      | 2-1  |
| 22     |    | الشاشة الافتتاحية في لغة <i>R</i> | 1-2  |
| 23     |    | قائمة <i>File</i>                 | 2-2  |
| 23     |    | قائمة <i>Edit</i>                 | 3-2  |
| 24     |    | مربع حوار <i>GUI preferences</i>  | 4-2  |
| 24     |    | قائمة <i>View</i>                 | 5-2  |
| 24     |    | قائمة <i>Misc</i>                 | 6-2  |
| 25     |    | قائمة <i>Packages</i>             | 7-2  |
| 25     |    | قائمة <i>Windows</i>              | 8-2  |
| 25     |    | قائمة <i>Help</i>                 | 9-2  |
| 28     |    | مربع حوار <i>Question</i>         | 10-2 |
| 29     |    | شاشة تحليل البيانات الاحصائية     | 11-2 |
|        |    | <i>R</i>                          |      |
| 30     |    | شاشة الاعياز <i>citation</i>      | 12-2 |
| 87     |    | نافذة حزم البيانات في <i>R</i>    | 1-5  |
| 87     |    | نافذة الحزمة <i>lattice</i>       | 2-5  |
| 88     |    | نافذة مجموعة البيانات             | 3-5  |
| 99     |    | الرسم البياني                     | 1-6  |
| 100    |    | لـ <i>hist(AirPassengers)</i>     |      |
| 101    |    | الرسم البياني                     | 2-6  |
| 101    | مع | لـ <i>hist(Temperature)</i>       |      |
| 101    |    | اضافة المعلمات                    | 3-6  |
| 101    |    | وضع البيانات على الرسم            | 4-6  |

|           |                                         |
|-----------|-----------------------------------------|
| 102 ..... | رسمين بعدد فوائل مختلفة 5-6             |
| 102 ..... | رسم دالة <i>Plot()</i> 6-6              |
| 104 ..... | اضافة وسيلة ايضاح 7-6                   |
| 105 ..... | تقسيم صفحة النتائج 8-6                  |
| 105 ..... | اياعز <i>mfcol</i> 9-6                  |
| 106 ..... | دالة <i>curve</i> 10-6                  |
| 106 ..... | دالة <i>barplot</i> 11-6                |
| 106 ..... | تعليمية <i>horiz=TRUE</i> 12-6          |
| 107 ..... | دالة <i>table()</i> 13-6                |
| 108 ..... | مخطط شريطي 14-6                         |
| 108 ..... | دالة <i>dotchart()</i> 15-6             |
| 109 ..... | رسم مجاميع ملونة 16-6                   |
| 109 ..... | دالة <i>boxplot()</i> 17-6              |
| 109 ..... | <i>boxplot</i> بربعات ملونة 18-6        |
| 110 ..... | دالة <i>pairs()</i> 19-6                |
| 110 ..... | مخطط بين <i>G</i> و <i>MPG</i> 20-6     |
| 111 ..... | دالة <i>layout()</i> 21-6               |
| 111 ..... | تخطيطات دائيرية 22-6                    |
| 112 ..... | دالة <i>Pie()</i> 23-6                  |
| 112 ..... | دالة <i>(pie)</i> و اضافة المعلمات 24-6 |
| 113 ..... | الامر <i>smoothScatter</i> 25-6         |
| 113 ..... | دالة <i>Oqqplot</i> 26-6                |
| 114 ..... | دالة <i>stripchart()</i> 27-6           |
| 114 ..... | اثنين من <i>(stripchart()</i> 28-6      |
| 115 ..... | دالة <i>contour()</i> 29-6              |
| 115 ..... | دالة <i>filled.contour</i> 30-6         |
| 116 ..... | رسم بيانات برkan 31-6                   |
| 116 ..... | رسم برkan بشكل اجمل 32-6                |
| 117 ..... | دالة <i>(opersp</i> 33-6                |
| 117 ..... | رسم مخروطي قائمه 34-6                   |

|                               |       |                               |      |
|-------------------------------|-------|-------------------------------|------|
| 150                           | ..... | <i>RStudio</i> شاشة           | 1-7  |
| 151                           | ..... | تبويب بيئة العمل              | 2-7  |
| 152                           | ..... | تبويب <i>Plot</i> لرسم واحد   | 3-7  |
| 152                           | ..... | تبويب <i>Plot</i> لعدة رسوم   | 4-7  |
| 158                           | ..... | ابعاد <i>(data)</i>           | 1-8  |
| 159                           | ..... | الامر <i>(library)</i>        | 2-8  |
| 159                           | ..... | <i>library(help=boot)</i>     | 3-8  |
| 160                           | ..... | بيانات الحزمة <i>boot</i>     | 4-8  |
| 160                           | ..... | قائمة <i>Packages</i>         | 5-8  |
| 161                           | ..... | نافذة <i>Select one</i>       | 6-8  |
| 161                           | ..... | نافذة <i>CRAN mirror</i>      | 7-8  |
| 162                           | ..... | نافذة <i>Repositories</i>     | 8-8  |
| 162                           | ..... | تحميل عدد من الحزم            | 9-8  |
| 162                           | ..... | قائمة                         | 10-8 |
| <i>Packages to be updated</i> |       |                               |      |
| 162                           | ..... | مربع حوار <i>Question</i>     | 11-8 |
| 163                           | ..... | مربع حوار <i>Select files</i> | 12-8 |
| 165                           | ..... | موقع <i>cran</i>              | 13-8 |
| 174                           | ..... | رسم الانحدار                  | 1-9  |
| 176                           | ..... | دالة <i>dnorm</i>             | 2-9  |
| 177                           | ..... | دالة <i>pnorm</i>             | 3-9  |
| 177                           | ..... | دالة <i>qnorm</i>             | 4-9  |
| 178                           | ..... | دالة <i>rnorm</i>             | 5-9  |
| 179                           | ..... | دالة <i>dbinom</i>            | 6-9  |
| 185                           | ..... | دالة <i>t.test</i>            | 7-9  |





**مؤلفاتي**

- (1) الحاسوب الالكتروني .. مقاهم واجراءات الجزء الاول ، الطبعة الاولى 2005
- (2) الحاسوب الالكتروني .. مقاهم واجراءات الجزء الثاني ، الطبعة الاولى 2005
- (3) الحاسوب الالكتروني .. مقاهم واجراءات الجزء الاول ، الطبعة الثانية 2009
- (4) الحاسوب الالكتروني .. مقاهم واجراءات الجزء الثاني ، الطبعة الثانية 2009
- (5) الحاسوب الالكتروني .. مقاهم واجراءات الجزء الثالث ، الطبعة الثانية 2009
- (6) مهارات الحاسوب / 2011
- (7) IT وتطبيقات VISUAL BASIC 2011 / 2012
- (8) التطبيقات المحاسبية بالحاسوب / 2012
- (9) الرياضيات المتقدمة / 2013

