البـرمـجـة بلغـة R

تأليف

ارشــد حميد حسن

تدريسي في كلية الادارة والاقتصاد - جامعة ديالى

قسم الاحصاء

2019م
الطبعة الاولى

اسم الكتاب : البرمجة بلغة R

اسم المؤلف : ارشد حميد حسن

الطبعة الاولى 2019

دار الاندلس للطباعة والنشر

بغداد – الاعظمية – مقابل المقبرة الملكية

الرقم المعياري الدولي للكتاب ISBN:978-9922-20-228-0

رقم الايداع في دار الكتب والوثائق ببغداد(199) لسنة 2019

الاهداء

الى روح والدي الطاهرة(رحمه اللّٰه)

الى والدتي الغالية

الى اخواني واخواتي

الى زوجتي العزيزة

بسم الله الرحمن الرحيم

﴿ وَمِنَ النَّاسِ وَالدَّوَابِّ وَالْأَنْعَامِ مُخْتَلِفٌ أَلْوَانُهُ كَذَلِكَ إِنَّمَا يَخْشَى اللَّهَ مِنْ عِبَادِهِ الْعُلَمَاءُ إِنَّ اللَّهَ عَزِيزٌ غَفُورٌ﴾

صدق الله العظيم
فاطر (28)

الحمد لله رب العالمين والصلاة والسلام على الرسول الكريم محمد (صل الله عليه وعلى اله وصحبه وسلم) ، احمدُ الله حمداً كثيراً طيباً مباركاً فيه الذي اعانني على انجاز كتابي الموسوم (البرمجة بلغة R) الذي يلبي جزءاً مهماً من حاجة الدارسين والباحثين لتعلم وفهم البرنامج ومعرفة تفاصيله ، وكذلك لحاجة المكاتب العراقية والعربية لمثل هكذا كتاب ،اذ ان هذا البرنامج يمثل لغة برمجية عالية المستوى وتستعمل في اغلب الكليات والمعاهد الأوربية في التحليل الاحصائي والمعلومات الحيوية، وبدأت الجامعات العربية باستعمالها ايضا في مجال التحليل الاحصائي والمعلوماتي ، لذلك يمثل هذا الكتاب ثمرة جهد لسنتين متتالتين من الدراسة بلغة البرمجة R وبداية لتأليفي هذا الكتاب ،حيث تجمعت مفردات وموضوعات واسعة وخبرة بسيطة على العمل بهذا البرنامج ،وقمتُ هذا الكتاب بأسلوب واضح وبسيط معززاً بالأمثلة ،فقد ضم اثنا عشر فصلاً تعتبر اللبنة الاساسية لهذا البرنامج كما وغطت جميع اساسيات هذا البرنامج .

احتوى الفصل الاول على الخوارزميات وخرائط سير العمليات في حين تضمن الفصل الثاني اسس ومميزات لغة البرمجة R واحتوى الفصل الثالث على الايعازات الخاصة بالإدخال والاخراج في حين احتوى الفصل الرابع على عبارات التحكم والفصل الخامس احتوى على جمل الدوران وحلقات التكرار اما الفصل السادس فقد احتوى على اهم الايعازات الخاصة بالمصفوفات والمتجهات وتضمن الفصل السابع فقد تضمن العمليات الحسابية الخاصة بالمصفوفات والمتجهات اما الفصل الثامن فقد تضمن بعض طرائق

التحليل العددي في حين تضمن الفصل التاسع على المحاكاة وطرق التوليد للأرقام العشوائية وشبه العشوائية واحتوى الفصل العاشر على تحليل الانحدار ومثال تطبيقي باستعمال لغة البرمجة R ، والفصل الحادي عشر تضمن الرسوم البيانية واهم الايعازات الخاصة بها في حين تضمن الفصل الثاني عشر التوزيعات الاحتمالية .

وفي الختام اتمنى قد قدمت ُشيءً مهماً لخدمة الطلبة والباحثين العراقيين ومن الله التوفيق

المؤلف

2019 م

المحتويات

الفصل الاول
الخوارزميات وخرائط السير

1-1مقدمة

رغم أن الحاسب الالكتروني يتميز بقدرته على إنجاز العمليات الحسابية حسب الأوامر و التعليمات المعطاة له بسرعة فائقة و بدقة متناهية و كذلك بإمكانياته الكبيرة في حفظ المعلومات الواسعة و المختلفة التي يعجز الإنسان عن حفظها و استعادتها باستعمال ذاكرته العادية. فهو يعجز عن أن يقوم بشكل ذاتي بحل أي مسألة مهما كانت بسيطة، أي أن عمله ينحصر في إنجاز الحلول للمسائل التي تبرمج له بشكل صحيح يتوافق مع الأسس العلمية الصحيحة التي تعتد عليها هذه الحلول.

لذا سوف نستعرض في هذا الفصل الخطوات الضرورية اللازمة لحل المسائل باستخدام الحاسب الالكتروني وكذلك توضيحاً مفصلاً لمفهوم الخوارزميات و خرائط سير العمليات التي تشكل العنصر الأساسي لكيفية البرمجة

2-1 خطوات حل مسألة باستخدام الحاسب:

عند حل أي مسألة باستعمال الحاسب الالكتروني تتم المعالجة بإتباع خطوات نبينها بإيجاز فيما يلي:

1-2-1تعريف وتحليل المسألة:

إن تعريف المسألة هو عبارة عن دقة التعبير في تطبيق المسألة بحيث يجعلها معروفة ومفهومة بصورة واضحة وبدون أي غموض لجميع الأشخاص العاملين ضمن مجال الاختصاص الذي تخضع له المسألة.

أما تحليل المسألة ووضع طريقة الحل فهو أصعب المصاعب و أشق الخطوات، و من أجل الوصول إلى حل صحيح فإن كثير من القوانين والطرق الرياضية المناسبة لحل المسألة يجب أن تستعمل و لربما تحتاج أيضاً إلى تطوير هذه القوانين والطرق لنجعلها تناسب الحل في كثير من الأحيان ففي هذه الخطوة يجب تحديد:

- طبيعة المخرجات(النتائج) و تنظيم كتابتها.
- المدخلات (البيانات أو المعلومات) و تحديد نوعها و تنظيم إدخالها إلى الحاسب الالكتروني.
- طرق الحل المناسبة و تقييمها بما يتلاءم مع كيفية تنفيذها بالحاسب الالكتروني و في ضوء ذلك يتم اختيار الحل الأفضل.

2-2-1 برمجة الحل خطياً:

بعد اختيار طريقة الحل المثالية و تحديد كل ما تشمله من علاقات رياضية، يتم التعبير عنها على شكل خطوات متسلسلة ومترابطة منطقياً، دقيقة الوصف تؤدي إلى الوصول إلى حل المسألة.و هذه الخطوات المتسلسلة تعرف بخوارزمية المسألة

Algorithm of the Problem و يمكن تمثيل هذه الخوارزمية بعد إيضاح جميع التعليمات والأوامر المتسلسلة التي يراد تنفيذها في كل خطوة بمخطط وصفي تسلسلي يدعى بمخطط سير العمليات Flowchart وذلك باستخدام مجموعة من الأشكال الاصطلاحية الرمزية. إن كلمة Algorithm مشتقة نسبة إلى العالم العربي المشهور الخوارزمي الذي قام بوضع أسس حل المسائل بشكل تتابعي.

3-2-1 برمجة الحل باستخدام إحدى لغات البرمجة:

إن مخطط سير العمليات هو عبارة عن تخطيط تصوري مساعد سهل الملاحظة بالنسبة للمبرمج و لكنه غير مفهوم عند الحاسب الالكتروني، لذلك فإن طريقة الحل الممثلة بمخطط سير العمليات يجب أن تكتب بإحدى لغات الحاسب التي يفهمها و التي تتلاءم مع حل المسألة.

و يلي ذلك كتابة البرنامج على الوسط الخارجي المناسب و إدخال البرنامج إلى الحاسب و البرنامج الناتج من هذه الخطوة و المكتوبة بإحدى اللغات الرمزية أو الراقية يسمى بالبرنامج المصدري

- ترجمة البرنامج المصدري:

بعد الانتهاء من كتابة البرنامج المصدري يتعين إدخاله إلى الحاسب للتأكد من صحة كتابته من جهة، ثم لترجمته إلى لغة الآلة بواسطة برنامج الترجمة الخاص في حالة عدم وجود أخطاء في البرنامج المصدري. و تمر عملية الترجمة في المراحل الآتية:

1. مرحلة التحليل المعجمي Lexical analysis:

في هذه المرحلة يتم مطابقة مفردات برنامج المصدر والعلاقات و الأسماء مع تلك المسموح بها في اللغة و اكتشاف أي أخطاء فيها.

2. مرحلة التحليل اللغوي والنحوي Syntax analysis:

في هذه المرحلة تجري عملية مطابقة تعليمات البرنامج المصدري مع القواعد اللغوية المستخدمة، و اكتشاف أي أخطاء فيها، بالإضافة إلى عملية تحويل البرنامج المصدري إلى تعليمات و أوامر رمزية بلغة التجميع.

3.مرحلة ترجمة البرنامج إلى لغة الآلة:

في هذه المرحلة نحصل على البرنامج الهدفي object program و الذي بموجبه يمكن البدء في عملية التنفيذ.

4-2-1 تجربة البرنامج و تنفيذه:

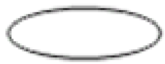
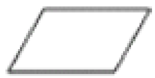
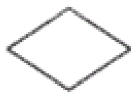
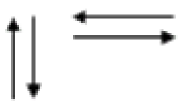
بعد الحصول على البرنامج الهدفي، تتم تجربته للتأكد من صحته منطقياً وذلك باستخدام عينة من المعطيات الاختبارية Test Data فإذا ثبت صحة طريقة الحل

بمطابقة النتائج الخارجة من الحاسب مع النتائج التي تم الحصول عليها يدوياً على سبيل المثال، يمكن تنفيذ البرنامج على المعطيات الحقيقية.

3-1 مفهوم خرائط سير العمليات:

الخوارزمية هي عبارة عن مجموعة من الخطوات المتسلسلة التي تصف بصورة مضبوطة وبدون أي غموض جميع الخطوات الرياضية والمنطقية اللازمة لحل مسألة ما. ولكن هذا الوصف في كثير من الأحيان يكون معقداً وصعب الملاحظة والتتبع لذلك فإن خريطة سير العمليات التي تمثل وصفاً تصويرياً لخطوات الخوارزمية تكون أكثر وضوحاً. وخريطة سير العمليات تقوم مقام الخوارزمية ويمكن بواسطتها ملاحظة تتبع التسلسل المنطقي لحل المسألة بكل سهولة، وغالباً ما تكون استخراج الخوارزمية من خريطة سير العمليات أسهل بكثير من كتابة الخوارزمية مباشرة.

و عند رسم خريطة سير العمليات لمسألة معينة فإننا نستخدم مجموعة من الأشكال الرمزية الاصطلاحية المبينة في الجدول التالي:

الرمز	الحدث الذي يمثله	مثال
	حدث طرفي Terminal لبيان بدء (Start) أو انتهاء (Stop) خريطة سير العمليات	START STOP
	عملية حسابية (Process)	LET X+Y
	إدخال/ إخراج INPUT \ OUTPUT لبيان إدخال/ إخراج معلومات من/ إلى الحاسب	INPUT X, Y PRINT Z
	اتخاذ قرار Decision	NO X=Y YES
	اتجاه تدفق (سريان) Flow line	
	تكرار أو دوران Loop	FOR I= 1 to 10

من أهم فوائد استخدام خرائط سير العمليات قبل كتابة البرنامج لمسألة ما، ما يأتي:

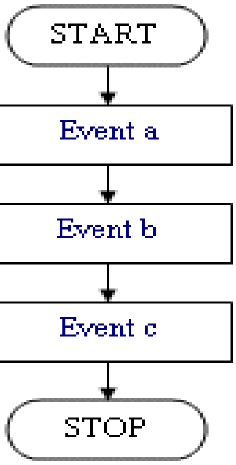
1. تمكن المبرمج من الإلمام الكامل بالمسألة المراد حلها و السيطرة على كل أجزائها بحيث تساعده على اكتشاف الأخطاء المنطقية (Logic Error) و التي تعتبر من أهم الأخطاء التي تجهد المبرمج.
2. تساعد بيسر و سهولة على تعديل البرامج الموضوعة بمجرد النظر.
3. يعتبر الاحتفاظ برسوم خرائط سير العمليات لحلول مسائل معينة أمراً مهماً إذ يكون مرجعاً عند إجراء تعديلات عليها أو استخدامها لحل مسائل أخرى مشابهة دون الحاجة إلى الرجوع إلى المبرمج الأول باعتبار أن الحلول الأولى قد صيغت في خطوات واضحة بسيطة و مفهومة.
4. توفير وسيلة مناسبة ومساعدة في كتابة البرامج ذات التفرعات الكثيرة.

هذا و يمكن تصنيف خرائط سير العمليات بما يلي:
- خرائط التتابع البسيط (Simple sequential Flowchart).
- خرائط التفرع (Branched Flowchart).
- خرائط الدوران البسيط (Loop Flowchart).
- خرائط الدورانات المتداخلة (Nested).

و يمكن للبرنامج الواحد أن يشتمل على أكثر من نوع واحد من هذه الأنواع. و سنتناول فيما يأتي شرح هذه الأنواع بشيء من التفصيل.

1-3-1خرائط التتابع البسيط:

يخلو هذا النوع من التفرعات Branches و الدورانات loops، و يكون الشكل العام لهذا النوع كما هو مبين في الشكل 1-1:



الشكل (1-1) خرائط التتبع البسيط

و كلمة Event الواردة في شكل 12-1 تعني الحدث أو العملية المطلوب تنفيذها.

مثال : أرسم خريطة سير العمليات لإيجاد مساحة و محيط دائرة نصف قطرها معلوم R.

مساحة الدائرة = πR^2

محيط الدائرة = $2\pi R$

حيث π = النسبة التقريبية

وقيمتها العددية ثابتة و تساوي 3.14 بينما R متغير.

الحل:

تكون خطوات الحل المبينة في الشكل 1-2 كما يلي:

1.ابدأ.

2.اقرأ قيمةR .

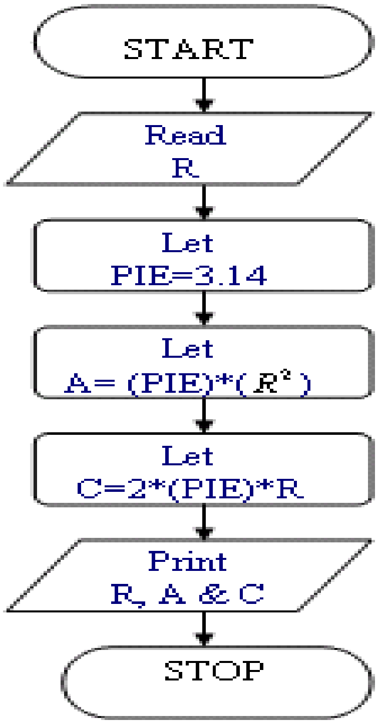
3.اجعل قيمة PI=3.14

4. احسب المساحة(A) من المعادلة A =(PI)*R^2.

5. احسب المحيط (C) من المعادلة $C = 2*(PI)*R$.

6. اطبع قيم كل من C, A, R.

7. توقف.او النهاية



الشكل (2-1)

مثال: ارسم خريطة سير العمليات لحساب قيمة كل من المتغيرات C, B, A في المعادلة الآتية:

$A = X^2 + 2y \ldots (1)$

$B = 2X - 3A \ldots (2)$

$C = A^2 + XB \ldots (3)$

إذا علمت أن قيم كل من Y, X معطاة (معلومة)، ثم اطبع قيم كل من C, B, A, Y, X.

الحل:

من الواضح أنه يمكننا من حساب قيمة المتغير A في المعادلة(1) لمعرفتنا بقيم المعطيات الأولية X ,Y، ويمكننا من حساب قيمة المتغير B في المعادلة (2) بالاعتماد على قيمة X المعلومة لدينا وقيمة المتغير Aالمحسوبة في الخطوة السابقة، أما قيمة المتغير C في المعادلة (3) بالاعتماد على قيم كل من المتغيرات X ,A ,B وكلهامعلومة.
وتكون خطوات حل المسألة كما هو مبين في الشكل 1-3 كما يلي:

1.ابدأ.

2.اقرأ قيمة كل من X ,Y.

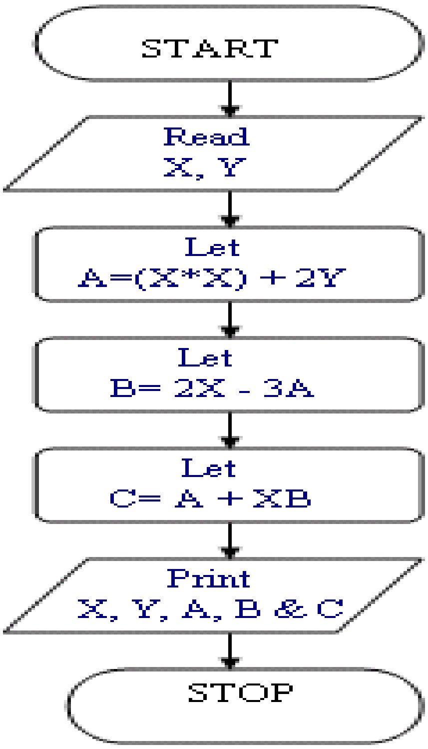
3.احسب قيمة A من المعادلة (1).

4.احسب قيمة B من المعادلة (2).

5.احسب قيمة C من المعادلة (3).

6.اطبع قيمة كل من X ,Y ,A ,B ,C.

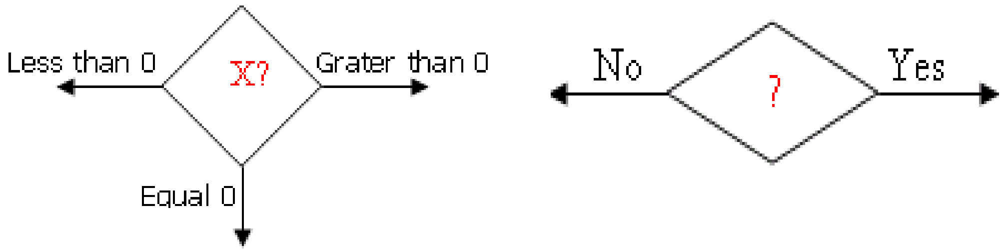
7.توقف.



الشكل (3-1)

2-3-1 خرائط التفرع:

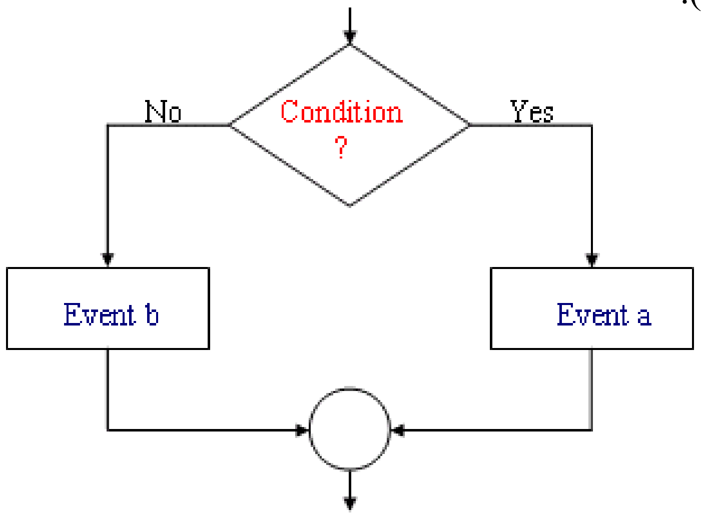
ويحدث التفرع في البرامج بسبب الحاجة لاتخاذ قرار أو مفاضلة بين اختيارين أو أكثر، وهناك أسلوبان في تنفيذ القرار(انظر شكل 1-4).



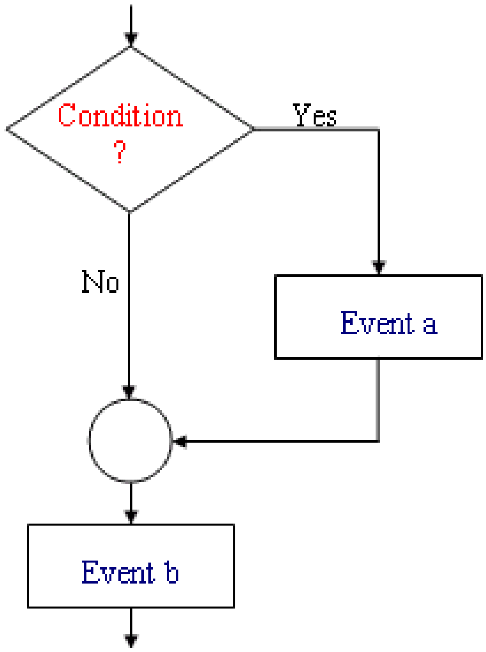
قرار ذو تفرعين   قرار ذو ثلاثة تفرعات

الشكل (1-4)

وبشكل عام فإن خرائط التفرع يمكن أن تأخذ إحدى الصورتين (انظر شكل 1-5 و الشكل 1-6).



الشكل(1-5)



الشكل (6-1)

يمكننا ملاحظة أن شكل 5-1 يبين أنه إذا كان جواب الشرط (Condition) YSE فإن الحدث التالي في التنفيذ يكون الحدث (a) أما إذا كان الجواب NO فإن الحدث التالي يكون الحدث(b) كما يمكننا أن نلاحظ في الشكل 6-1 أنه إذا كان جواب الشرط YSE فإن الحدث التالي في التنفيذ يكون الحدث (a) ثم يتبعه الحدث (b) أما إذا كان جواب الشرط NO فإن الحدث التالي يكون الحدث (b) مباشرة.

مثال: ارسم خريطة سير العمليات لإيجاد قيمة الاقتران F(x) المعرف حسب القاعدة التالية:

$$f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

الحل:

خطوات الحل المبينة في الشكل 1-7 تكون:

1. ابدأ

2. اقرأ قيمة المتغير X .

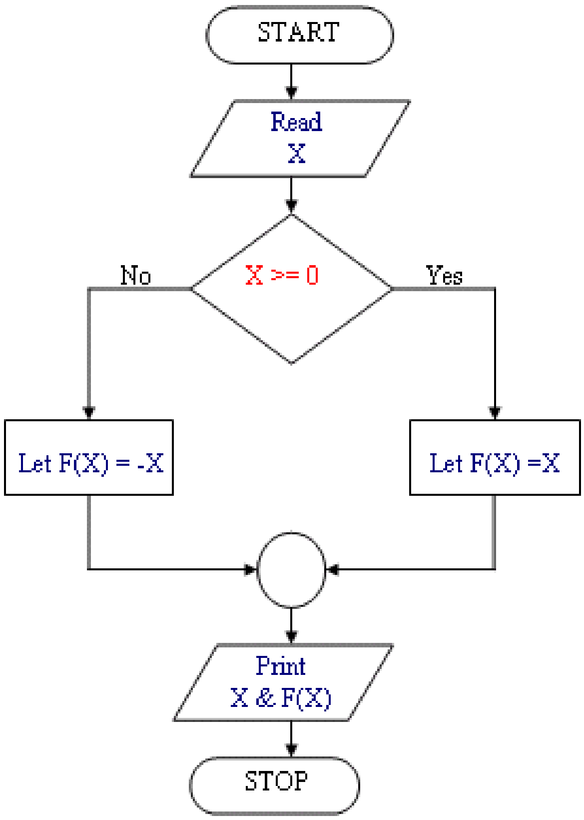
3. إذا كانت X أكبر أو تساوي صفرًا اذهب إلى خطوة(4) وإلا فأذهب إلى الخطوة(5).

4.احسب قيمة الاقتران من F(X)=X ثم اذهب إلى الخطوة(6).

5.احسب قيمة الاقتران من F(x)= -X.

6. اطبع قيمة كل من F(x), X.

7. توقف.



الشكل (7-1)

مثال: ارسم خريطة سير العمليات لحساب قيمة W طبقًا للمعادلات الآتية علمًا بأن قيمة المتغير X معطاة معلومة:

$$W = \begin{cases} X^2 + 1 & X > 0 \\ X + 5 & X = 0 \\ 2X^3 - 1 & X < 0 \end{cases}$$

الحل:

خطوات الحل كما هي مبينة في الشكل (1-8) :

1. ابدأ.

2. اقرأ قيمة المتغير X.

3. إذا كانت X أكبر من صفر فاذهب إلى الخطوة 4 أما إذا كانت ليست أكبر من فاذهب إلى خطوة 5.

4. احسب W من المعادلة (1) ثم اذهب إلى الخطوة 8.

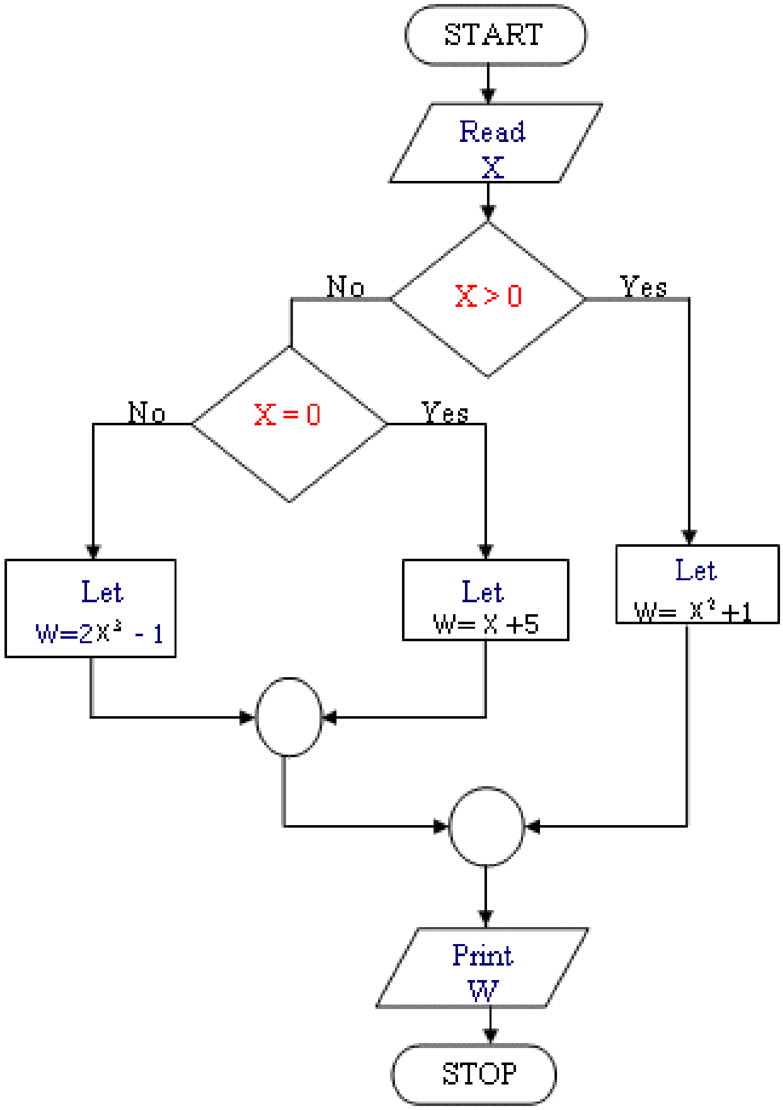
5. إذا كانت X تساوي صفر فاذهب إلى الخطوة 6 وإلا فاذهب إلى الخطوة 7.

6. احسب W من المعادلة (2) ثم اذهب إلى الخطوة 8.

7. احسب W من المعادلة (3) ثم اذهب إلى الخطوة 9.

8. اطبع قيمة W.

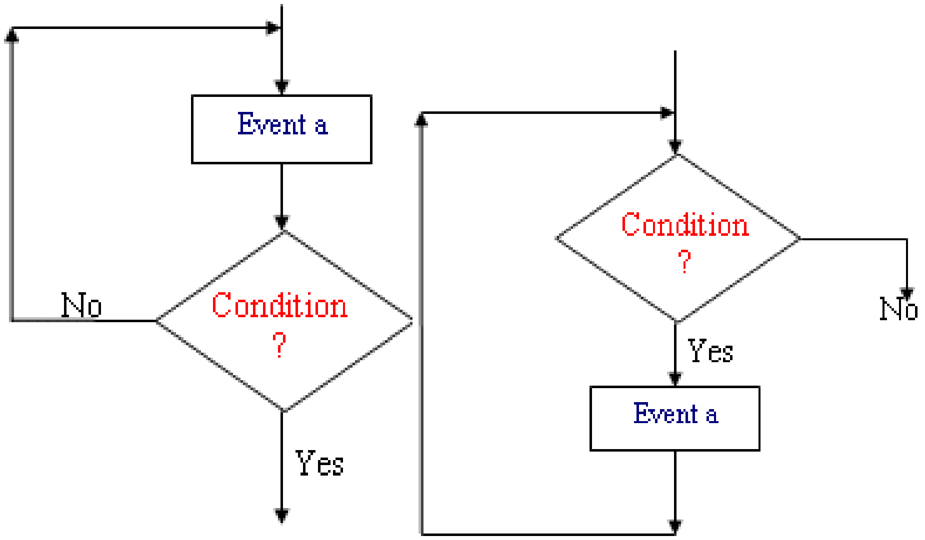
9. توقف.


START
Read
X
X > 0
No
Yes
X = 0
No
Yes
Let
W=2X³ - 1
Let
W= X +5
Let
W= X² +1
Print
W
STOP

الشكل(8-1)

3-3-1 خرائط الدوران (التكرار) البسيط:

وهذه الخرائط نحتاج إليها لإعادة عملية أو مجموعة من العمليات في البرنامج عددًا محدودًا أو غير محدود من المرات، ويكون الشكل العام لمثل هذه الخرائط كما يلي انظر الشكل(1-9)



حدث (a) يتكرر تنفيذه في كل دوره حتى يصبح جواب الشرط YES.

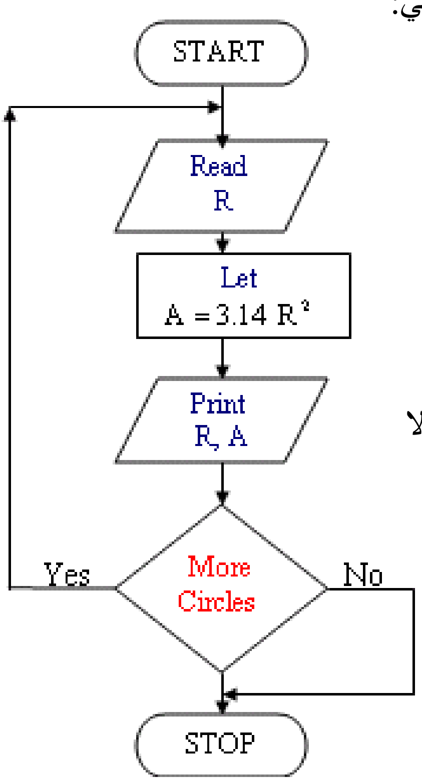
الحدث (a) يتكرر تنفيذه في كل دورة طالما كان جواب الشرط YES.

الشكل (1-9)

مثال: ارسم خريطة سير العمليات لإيجاد مساحة مجموعة من الدوائر أنصاف أقطارها معلومة:

الحل:

تكون خطوات الحل المبينة في الشكل (1-10) كما يلي:



الشكل (1-10)

1. ابدأ.
2. اقرأ نصف قطر الدائرة (R).
3. أوجد مساحة الدائرة (A).
4. اطبع قيم كل من A, R.
5. هل هناك مزيد من الدوائر؟

   فإن كان نعم فعد إلى الخطوة(2) وإن كان لا فعد إلى الخطوة (6).
6. توقف.

1-3-4 العداد Counter

في كثير من الأحيان نحتاج في برامج الحاسب الالكتروني إلى العد Counting، فقد نريد مثلاً أن نعد عدد كل من الطلاب والطالبات ضمن الشعبة, وقد تكون هذه العملية سهلة للإنسان لأنها أصبحت ضمن قدراته العقلية التي يكتسبها من الطفولة، إلا أن الحاسب يحتاج إلى تصميم خوارزمية للعد Counting Algorithm تتضمن خطوات معينة إذا اتبعتها استطاع أن يعد.

ويمكن تحديد الخطوات التي يتبعها الحاسب حتى يتمكن من العد في الخطوات الأساسية:

1.اجعل العداد مساويًا للصفر.

2.اجعل القيمة الجديدة للعداد تساوي القيمة القديمة لها زائد واحد, أي أن:
قيمة العداد (الجديدة)= قيمة العداد (القديمة)+1

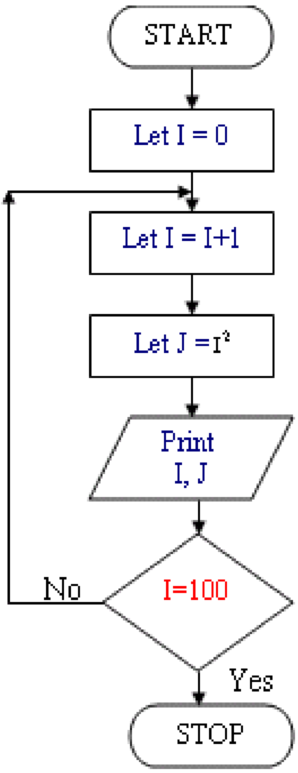
3.كرر الخطوات ابتداء من الخطوة 2.

مثال: ارسم خريطة سير العمليات التي يتبعها الحاسب لطباعة الأعداد الطبيعية من 1 إلى 100 ومربعاتها.

الحل:

خطوات الحل مبينة في الشكل (1-11)هي:

1. ابدأ.
2. اجعل I=0.
3. اجعل I=I+1.
4. اجعل $J = I^2$.
5. اطبع J, I.
6. إذا كانت I=100 اذهب إلى الخطوة 7 وإلا اذهب إلى الخطوة 3.
7. توقف.



الشكل (1-11)

5-3-1 المجاميع الإجمالية:

في كثير من الأحيان نحتاج في برامج الحاسب الإلكتروني إلى جمع مجموعة كبيرة من الأعداد التي تمثل معطيات ظاهرة معينة، فمثلاً قد نرغب في إيجاد الوسط الحسابي لأعمار طلاب الجامعة، ولتحقيق هذا أولاً يجب أن نحسب مجموع أعمار الطلاب، وطبعًا ليس عمليًا إعطاء رمز أبجدي لكل عمر طالب فقد تحتاج لأكثر من عشرة الآلاف رمز، في مثل هذه الحالات نصمم خوارزمية معينة للتجميع تسمى خوارزمية التجميع summers Algorithm تتضمن خطوات محددة إذا اتبعها الحاسب استطاع أن يجمع أي كمية من البيانات باستخدام متغيرين اثنين إحداهما هو المتغير الذي نجمعه والآخر هو الجمع الإجمالي (المجمع)، ويمكن تحديد الخطوات التي يجب أن يتبعها الحاسب لتحقيق ذلك في أربع خطوات هي:

1. اجعل المجمع مساويًا الصفر.
2. ادخل قيمة واحدة للمتغير.
3. اجعل القيمة الجديدة للمجمع تساوي القيمة القديمة له زائد القيمة المدخلة للمتغير، أي أن:

قيمة المجمع الجديدة = قيمة المجمع القديمة + آخر قيمة مدخلة للمتغير.

4. كرر ابتداءً من الخطوة الثانية.

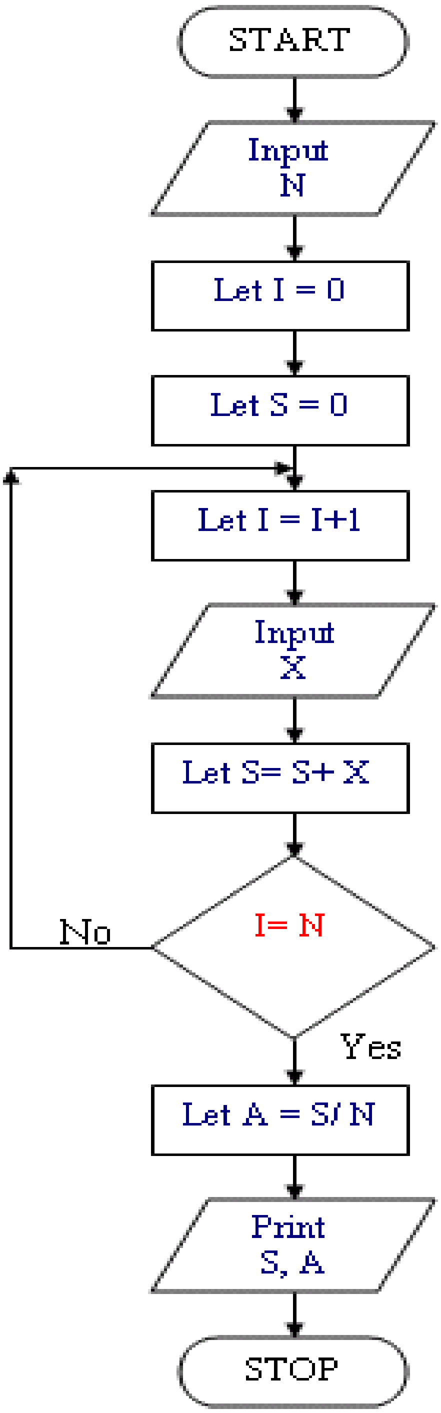
مثال:

ارسم خريطة سير العمليات لإيجاد الوسط الحسابي لأعمار طلاب شعبتك. الحل: نفترض أن إجمالي عدد الطلاب =N ونستخدم عددًا لرقم كل طالب ونرمز له بالرمز I ونرمز لعمر الطالب بـX ونستخدم مجمعًا لأعمار الطلبة ونرمز له بالرمز S ونستخدم الرمز A ليدل على معدل أعمار الطلبة.

الحل:

وتكون خطوات الحل كما هو مبين في الشكل (1-12)هي:

1. ابدأ.
2. ادخل إجمالي عدد الطلاب (N).
3. اجعل I=0.
4. اجعل S=0.
5. اجعل I=I+1.
6. ادخل X.
7. اجعل S=S+X.
8. إذا كانت I=N اذهب إلى الخطوة 9 وإلا اذهب إلى الخطوة 5.
9. اجعل A=S/N.
10. توقف.

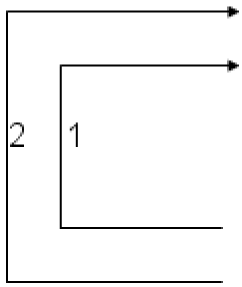


الشكل(1-12)

6-3-1 خرائط الدورانات المتدخلة:

في هذه الحالة تكون الدورانات داخل بعضها البعض بحيث لا تتقاطع فإذا كان لدينا مثلاً دورانان من هذا النوع (انظر شكل 1-13 فيسمى الدوران قم (1) دورانًا داخليًا (Inner Loop) بينما الدوران رقم (2) دورانًا خارجيًا Outer Loop) ويتم التناسق في عملي مثل هذين الدورانين بحيث:

تكون أولوية التنفيذ للدوران الداخلي.



الشكل (1-13)

مثال: يرغب نجار في تقطيع مجموعة من القطع الخشبية طول كل منها يزيد عن3 متر إلى قطع صغيرة طول الواحدة منها يساوي 3 متر.ارسم خريطة سير العمليات.

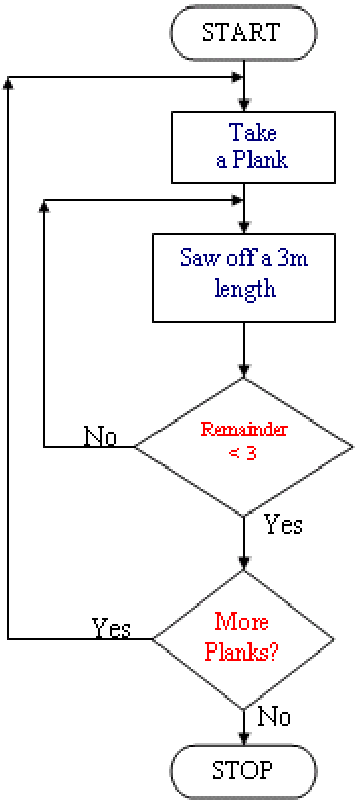
الحل:

خطوات الحل المبينة في شكل (1-14)هي:

1. ابدأ.
2. خذ قطعة.
3. اقطع منها قطعة طولها 3 متر.
4. هل المتبقي يزيد عن 3 متر؟
   إذا كان الجواب نعم فاذهب إلى الخطوة(3). وإذا كان الجواب لا فاذهب إلى الخطوة (5).

5. هل هناك مزيد من القطع المراد تقطيعها ؟ إن كان الجواب نعم فاذهب إلى الخطوة(2) وإن كان لا فاذهب إلى الخطوة(6).
6. توقف.

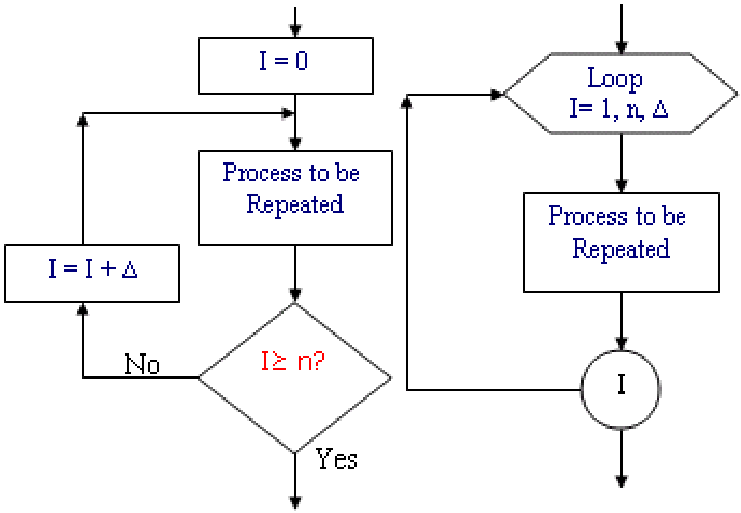
ملحوظة: يلاحظ من الشكل (1-14) أن الدوران الداخلي يتضمن تقطيع القطعة الواحدة إلى قطع متعددة طول كل منها 3 متر بينما يمثل الدوران الخارجي تناول قطعة واحدة جديدة لتنفذ عليها إجراءات الدوران الداخلي.



الشكل (1-14)

1-3-7 صيغة الدوران باستعمال الشكل الاصطلاحي:

لقد عرفنا في الفقرتين السابقتين مفهوم الدوران البسيط والدورانات الضمنية ويمكننا الآن استخدم الشكل الاصطلاحي للدوران والوارد على النحو التالي:



الشكل (1-15)

نلاحظ في الشكل (1-15) أننا نحتاج إلى العناصر الآتية:

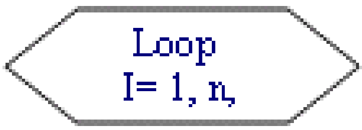
- القيمة الأولية للعداد I (هنا I=1).
- القيمة النهائية للعداد I (هنا I=1).
- القيمة النهائية للعداد I (هنا n).
- قيمة الزيادة عند نهاية كل دورة Δ.

نلاحظ في الشكل (1-15) إن إجراءات الدوران كانت تتم طبقاً للخطوات الآتية والمفصلة من قبل المبرمج:

1. أعط I قيمة أولية.
2. أتم الإجراءات المطلوب إعادتها.
3. (تقرير) إذا كانت قيمة العداد 1 وصلت إلى القيمة النهائية n اخرج إلى الخطوة التالية في البرنامج وإلا فاذهب إلى الخطوة (4).
4. زد I بمقدار الزيادة ∈ .
5. عد إلى الخطوة (2).

يمكننا استبدال الخطوات المفصلة (5,4,3,2,1)في الشكل(15-1) بخطوة مجملة واحدة مبينة في الشكل الاصطلاحي للدوران شكل15-1حيث تنفذ هذه الخطوات بصورة أوتوماتيكية من قبل الحاسب، وهذا من شأنه تسهيل عملية البرمجة واختصار عدد التعليمات في البرنامج وتجنب بعض الأخطاء.

ملاحظة: تعتبر قيمة ∆تساوي 1 دائمًا إذا لم تعط قيمة أخرى بخلاف ذلك، وفي حالة عدم ذكر قيمة ∆يصبح الشكل الاصطلاحي الوارد في الشكل 15-1 كما يلي حيث تكون قيمة ∆تساوي 1وبصورة أوتوماتيكية.

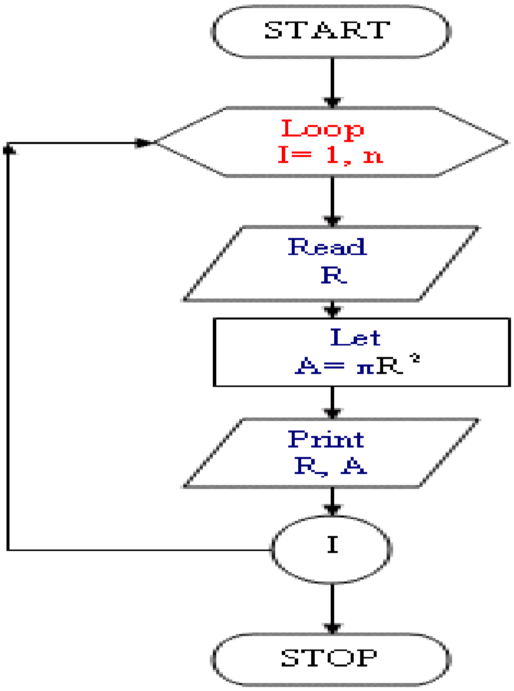


الشكل (16-1)

مثال: أعد حل مثال الموضح في الشكل (11-1) لإيجاد مساحة n من الدوائر باستخدام الشكل الاصطلاحي للدوران.

الحل:

خطوات الحل كما هي مبينة في الشكل (17-1).



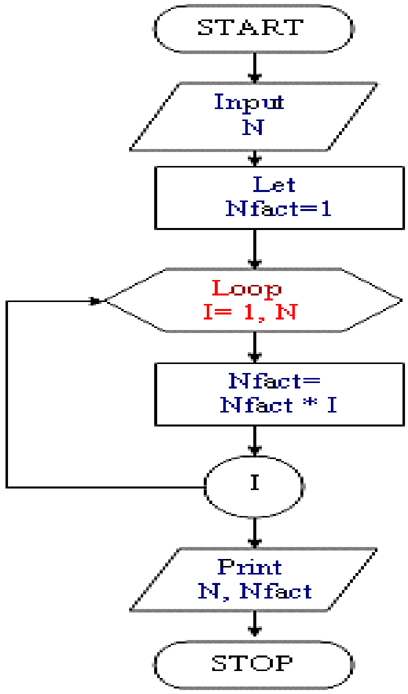
الشكل (17-1)

مثال: ارسم خريطة سير العمليات لإيجاد N!.

الحل:

.N!=N (N-1) (N-2) …3*2*1

فخطوات الحل كما يلي هي مبينة في الشكل (18-1) :


START
Input
N
Let
Nfact=1
Loop
I= 1, N
Nfact=
Nfact * I
I
Print
N, Nfact
STOP

الشكل (18-1)

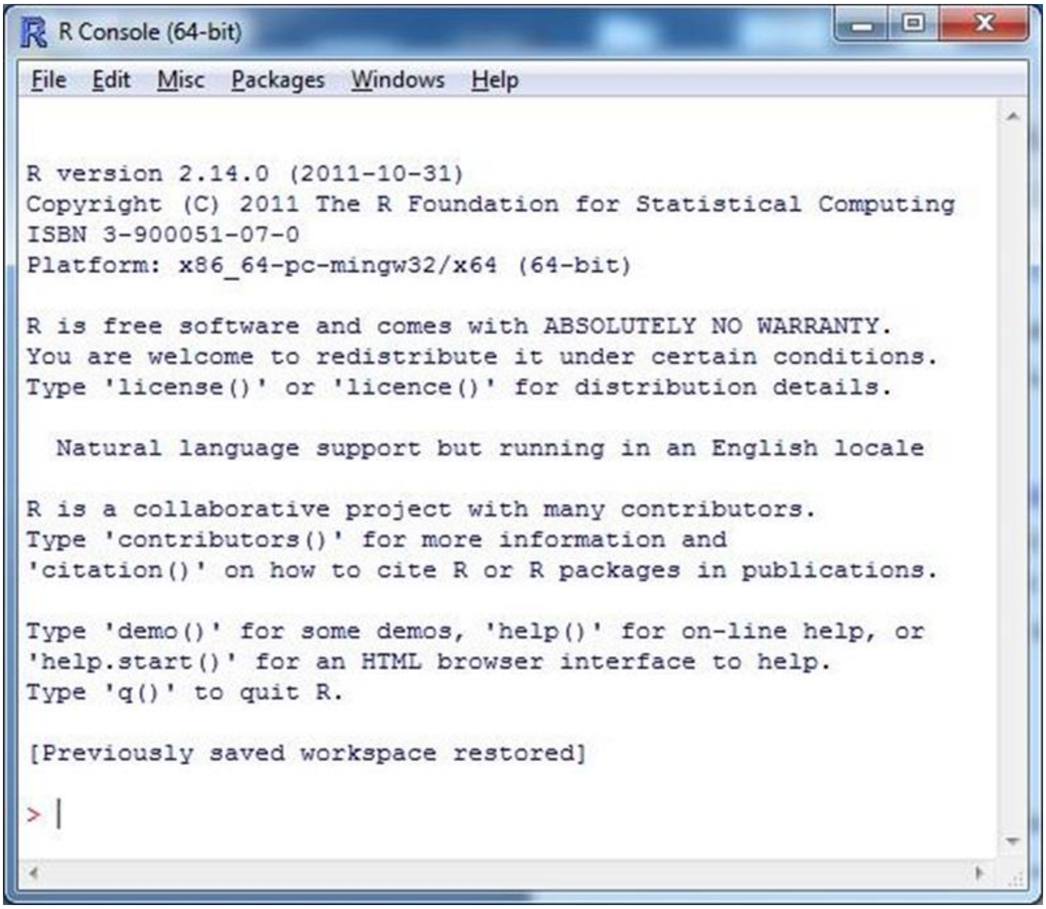
الفصل الثاني
اسس البرمجة Basic of programming R

1-2 المقدمة

تعد لغة **R** من اللغات التي صعد نجمها حديثا وبشكل سريع بمجال البرمجة العلمية في قطاعي الإحصاء والمعلوماتية الحيوية (bioinformatics) حيث باتت معتمدة على نطاق واسع في كثير من الجامعات ومراكز البحث العلمية، وأصبحنا نرى استخدامها والإشارة إليها في المقالات المنشورة بالمجلات العلمية المحكّمة يزداد بشكل طردي ومتسارع، هذا عدى عن حقيقة كونها لغة حرة مفتوحة المصدر يخضع توزيعها لترخيص GPL الشهير. كل ذلك أدى إلى تزايد ما هو متوفر ومتاح على الشبكة (الإنترنت) من مصادر لها على توزع طيف تلك المصادر، فهناك الكتب الإلكترونية والدروس التعليمية وحتى المناهج الأكاديمية والدورات التدريبية إضافة إلى البرامج الجاهزة والمكتوبة بلغة R لتنفيذ هذه المهمة أو تلك، حتى أنها باتت تحظى ببعض الامتياز مقارنة بالعديد من العمالقة في قطاع البرمجة الرياضيات العلمية والإحصائية مثل SAS و SPSS خصوصا في مجال توافر الجديد من الطرق والخوارزميات الحديثة، حيث يقاد هذا التوجه في معظمه من طرف الجامعات ممثلة بطلاب الدراسات العليا يحفّزهم على ذلك سهولة بناء الإضافات لهذه اللغة، ويعتبر هذا الأسلوب رغم ما قد يشوبه من نقاط ضعف تتعلق بموثوقية وجودة وغزارة تلك الإضافات الجديدة، والتي تتبع خبرة ومهارة مطوريها وناشريها، لكنها تبقى في القطاع العلمي والأكاديمي أفضل كثيرا من البدائل التجارية التي يعيبها ارتفاع ثمنها من جهة، ومن جهة أخرى بطئ إضافة التحديثات التي تعكس تطور القطاعات العلمية المختلفة، حيث أنها عادة ما تتبع دورة تجارية تتحكم بها الشركات المنتجة.

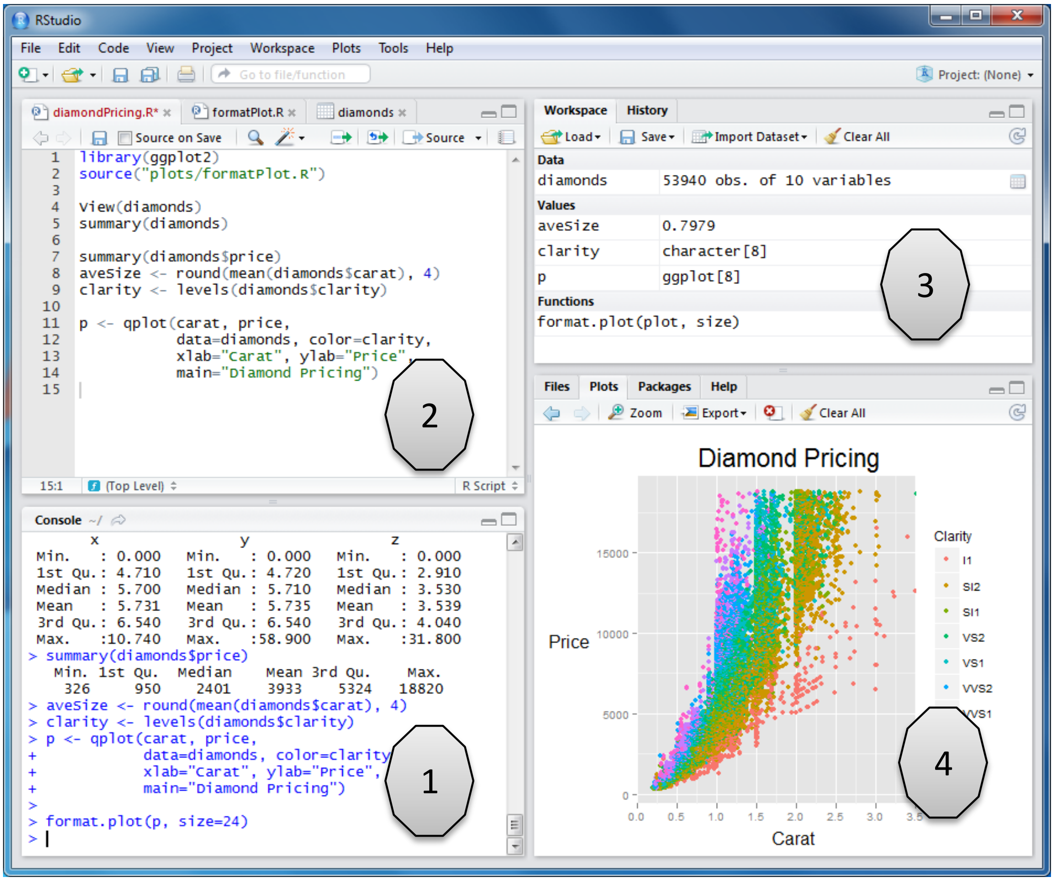
يمكنك تحميل لغة R من الموقع الرسمي لها على الشبكة http://www.r-project.org حيث توجد إصدارات منها لمعظم أنظمة التشغيل الشائعة ومنها Windows و Linux وحتى Apple. إن عملية التنصيب

سهلة وتخلو من التعقيدات، وعند الانتهاء منها يمكنك تشغيل بيئة عمل لغة R بالنقر على الأيقونة الخاصة بالبرنامج سواء تلك الموجودة على سطح المكتب أو من خلال قائمة البرامج، وحينها ستظهر لك شاشة سطر الأوامر الخاصة بلغة R وهو المكان المعتاد لكتابة الأوامر الخاصة بهذه اللغة كما هو ملاحظ في الشكل التالي:



حيث علامة الاكبر من (>) تدل على ان البرنامج جاهز للعمل ،وبعدها تنزيل بيئة العمل R-Studio وهي بيئة تطوير كثيرة الاستخدام والشيوع بين مستخدمي R بسبب تبسيطها واختصارها لكتابة الأوامر ولدعمها لميزات كثيرة وذلك من الرابط الاتي https://www.rstudio.com/products/rstudio/download/ . بعد تنصيبك

لبرنامج R-Studio وتشغيله ستجد أن بيئة العمل تنقسم إلى أربعة أقسام كما موضح بالشكل ادناه.



القسم الأول Console : وفيه يتم تنفيذ الأوامر، وبإمكانك كتابة الكود أو الأمر الذي تريد تنفيذه ثم الضغط على Enter ليتم التنفيذ، ولست بحاجة لحفظ التعليمات بشكل كامل في R Studio لأنه يمتلك ميزه إكمال العبارات التي يمكنك الاستفادة منها بالضغط على زر Tab ، فتنبثق قائمة لكل الأوامر القريبة من الأمر الذي بدأت بكتابته فتختار منها ما تشاء.

القسم الثاني محرر المصدر Source Editor : وفيه يمكن كتابة الأوامر، وتعديلها، وحفظها للاستفادة منها لاحقا، كما يمكنك تنفيذ السطر الذي تشاء منه بالضغط على Ctrl+Enter وتستطيع تنفيذ أي جزء من الكود بتحديده باستخدام الفأرة ثم الضغط أيضاً على Ctrl+Enter .

القسم الثالث ساحة العمل والحافظة والملفات Workspace, History and Files : في ساحة العمل يمكن مشاهد المتحولات التي تم تعريفها، وفي الحافظة تظهر الأوامر التي تم تنفيذها، كما يمكن إعادة تنفيذ أي تعليمة تريد بمجرد النقر عليها نقرتين متتاليتين، أو نقل التعليمة إلى محرر المصدر بالنقر على زر Shift مع نقرتين متتاليتين على التعليمة، أما الملفات وهي اختصار لمستعرض الملفات، ففيها يتم عرض الموقع من القرص الصلب والذي يتم العمل فيه، وبإمكانك تغيير الموقع إلى أي مسار تريده.

القسم الرابع الرسوم البيانية والحزم والمساعدة Plots, Packages and Help : يتم عرض جميع الرسوم التي قمت برسمها في Plots ويمكنك التنقل بين هذه الرسوم وحفظها، أما الحزمة، فهي مجموعة من التوابع المعرفة مسبقاً، ويحتوي R على الكثير من الحزم الجاهزة التي لم تترك أي جانب من الإحصاء إلا ودخلت فيه، وفي هذه اللائحة تستطيع تنزيل الحزم من الانترنت وإجراء التحديثات وغير ذلك، أما لائحة المساعدة فتقدم لك المساعدة عن أي أمر تقوم بكتابته في صندوق البحث

2-2 اهمية لغة البرمجة R

هي منصة برمجية مفتوحة المصدر من أجل تحليل البيانات الإحصائية. بدأ مشروع R في عام 1993 كمشروع أطلقه اثنين من الإحصائيين في نيوزيلندا، وهما روس إلهاكا و روبرت جينتلمان، وكان هدفهما إنشاء منصة بحث جديدة في الحوسبة الإحصائية. ومنذ ذلك الحين نما هذا المشروع الريادي ليشمل أكثر من عشرين

إحصائي وعالم كمبيوتر من جميع أنحاء العالم. وبسبب كونها منصة مفتوحة المصدر، تم اعتماد R بسرعة كبيرة من قبل أقسام الإحصاء من جامعات في مختلف أنحاء العالم، وقد جذبتهم الطبيعة التوسعية لها كمنصة للبحوث الأكاديمية، كما أن مجانية المنصة لعبت دوراً هاماً كذلك. وخلال فترة ليست بطويلة بدأ الباحثون الإحصائيون وعلماء البيانات والتعلم الآلي بنشر الأبحاث العلمية المحتوية على التعليمات البرمجية لـ R لتنفيذ مهام العمل الجديدة، ضمن أغلب المجلات الأكاديمية. جعلت المنصة R هذه العملية سهلة للغاية: يمكن لأي شخص أن ينشر حزمة عمل ضمن المنصة في "شبكة الأرشيف الكامل لـR" المسماة اختصاراً بـCRAN، وتصبح متاحةً للجميع. حتى كتابة هذه السطور، ساهم آلاف مستخدمو منصة R بأكثر من 6100 حزمة عمل، موسعين قدرات المنصة إلى مجالات متنوعة كالاقتصاد وتحليل التجارب السريرية والعلوم الاجتماعية وبيانات الويب. ويمكن لأي شخص أن يقوم بالبحث عن التطبيقات في MRAN عن الموضوع الذي يريده.

تقوم العديد من الشركات والمنظمات الأخرى بالعمل على توسيع نطاق مشروع R، مع الحفاظ على الجوهر الأصلي عن طريق مؤسسة R غير الربحية (مقرها في فيينا، النمسا). وقد قامت TheBioConductor بإنشاء أكثر من 900 حزمة عمل إضافية، جاعلة هذا المشروع رائداً برمجياً في تحليل البيانات الجينية والوراثية. كما أن RStudio أنشأت بيئة تطوير تفاعلية رائعة بلغة R، معززةً إنتاجية المستخدمين في جميع أنحاء العالم. وقد قامت Revolution Analytics بدعم مشروع R بثورة مفتوحة جعلت تضمينه ضمن أي تطبيقات أخرى أمراً سهلاً.

بالإضافة إلى استخدام R على نطاق واسع ضمن القطاع الأكاديمي، لم يمض وقت طويل حتى بدأ استخدامها ضمن القطاع التجاري كذلك. ففي يناير 2009، كان مشروع Rهوموضوع الصفحة الأولى لصحيفة نيوورك تايمز للإصدارالتقني، مولياً هذا

المشروع الكثير من الاهتمام، كما أن شركة Revolution Analytics كانت فعالة جداً وقدمت الدعم الفني وخدمات ضخمة للبيانات الكبيرة.

4-2 مميزات لغة البرمجة R

1. مجانية، مفتوحة المصدر، ومتاحة للجميع
2. متعددة المنصات يعمل على أنظمة لينوكس ويونكس وماك وويندوز.
3. مختصة في التحليل الإحصائي وبنائها "syntax " سهل ملائم جدا لهذه الغاية، مثلا لحساب المجموع والمعدل والتباين. أستعمل أوامر بديهية مثل:

mean, sum, var

4. تعتمد فلسفة البساطة والحد الأدنى، أي أنها تعطيك المخرجات التي تحتاجها فقط وتتفادى تكديس النتائج كما تفعل برمجيات إحصائية أخرى (كتقارير SPSS).
5. لغة مفسرة ولغة لكتابة السكربتات مثل بايثون.
6. ذات أداء عال وقابلة للموازاة (Parallel computing) وهو أمر هام لعمليات حوسبة معقدة مثل نمذجة ومحاكاة المناخ والنظم الأحيائية،
7. التمثيل بياني ذو جودة عالية مع إمكانية إنتاج مخططات ثلاثية الأبعاد

4-2 رموز لغة R : R Symbols

تتكون لغة R من العناصر الأساسية التالية:

أ- حروف أبجدية إنكليزية: وهي: A, B, … , Z, a, b, … , z

ب- أرقام حسابية: 0, 1, 2, …, 9

ج- رموز خاصة مثل:) , (,* , ; , < , > , = ,- , + ,{ , ... الخ.

1-4-2 الثوابت Constants:

يوجد في لغة R أنواع متعددة من الثوابت أهمها:-

(أ) الثوابت العددية Numerical Constants:

وتتكون من عدد من الأرقام ولها عدة أشكال هي:

(1) الثوابت الصحيحة: مثال: 0, +23, 472, -18

ملاحظة: أكبر عدد صحيح مستخدم.

(2) الثوابت الحقيقية: مثل: 0.0, 51.8, 472.5, -18.0

(3) الثوابت الحقيقية المدونة تدويناً يائياً: حيث تحول الصيغة الجبرية 10N إلى صيغة يائية EN فمثلا تصبح 2.0×103 في الجبر: 2.0E3 أو 2.0E+3 بالتدوين اليائي في R وكذلك تصبح 1.7×102- في الجبر: 1.7E2- في التدوين اليائي وكذلك تصبح 10-3×3.2E-3 : 3.2 :0.0032

(4) الثوابت العقدية: مثل: 1 - 2i , 6 - 9i , 6 + sin (0.5) * j , sqrt (- , 2)

مثال : c2 = 3 * (2 – sqrt (-1) * 3) ⟹ 6.000 – 9.000i

(ب) الثوابت الرمزية String Constants:

يسمى هذا النوع من "ثوابت" مجازاً لأن الثابت هذا يتكون من حروف وأرقام ورموز توضع بين علامتي اقتباس quotations مفردة أي ' ' ويستخدم عادة كعناوين توضح القيم الناتجة من الحسابات ووحداتها، تسمى العبارات التالية والموجودة بين الحاصرات العليا ثوابت رمزية.

'The speed of wind ='

'I love Basrah'

كل الثوابت الرمزية أعلاه، وان استخدمت أرقاما حسابية داخلها، فهي لا تحمل معنى حسابي، ومن الجدير بالذكر أثناء استعمال الثوابت الرمزية انه لا يجوز استخدام حاصرات علوية داخل حاصراتها، كما ينبغي التنبيه أي أن هناك قيماً رمزية للحروف يعتبر الحرف A اقل من الحرف B ويمكن كتابة ذلك بالصورة:

'A' < 'B'

(جـ) الثوابت المنطقية Boolean Constants:

وهي الثوابت التي قيمتها العددية (1) في حالة true و (0) في حالة false.

مثال:

3 > 2 ⇨ 1

0 > 5 ⇨ 0

2-4-2 المتغيرات Variables:

عندما نعمل في R فإننا سنتعامل مع كائنات تسمى الأشياء"objects" ، وهذه الأشياء ستكون في أغلب الأحيان متغيرات "variables" أو دوال"functions"

ان المتغير الافتراضي في R هو الجدول"vector" ، وهذا يعني أن أي متغير مفرد نقوم بإنشائه دون تحديد أي شيء آخر

مثال:

```
> x<-1
> x
[1] 1
```

سيكون العنصر الأول في جدول [1]. خلافا لبايثون ترقيم عناصر الجدول يبتدئ من 1 وليس 0. لا يمكن للجدول احتواء سوى نوع واحد من المتغيرات، ونسمي هذه الخاصية ذرية المتغير "atomicity". يمكن أن تكون المتغيرات بشكل أساسي عددية "numeric"، أو عددية مركبة "complex"، أو نصية "character"، أو منطقية "logical"، أو خاصة."special" هناك عدة أنواع من المتغيرات في لغة R وهي:-

(أ) المتغيرات العددية Numerical Variables:

تتكون من حرف واحد أو مجموعة من الحروف من A إلى Z و a إلى b ويمكن أن يحتوي على أرقام من 0 إلى 9 ويمكن أن تكون سلسلة من الأرقام والحروف بشرط أن

يبدأ بحرف (خليط من أرقام وحروف مبدوءة بحرف) ويمكن كذلك أن يحتوي المتغير على underscore حتى 63 رمزاً. وتكون قيمة المتغير عددية (صحيح، حقيقي، عقدي أو أسي).

مثال: Ali_Ahmed, X2, S2, ks, K

نأخذ في البداية بعض الأمثلة من متغيرات عددية:

مثال:

```
> x<-1
# keep in mind that ‘x’ is a vector
> y<-2.3
> z<-2+3i
```

سنقوم الآن بتفقد أنواع المتغيرات التي قمنا بإنشائها:

```
> class(x) # what is the class of variable x
[1] "numeric"
> typeof(x) # what is the type of variable x (more specific)
[1] "double"
```

كما ترى هنا كل الأعداد التي نقوم بإدخالها يعتبرها R، بشكل افتراضي، ثنائية double حتى نقوم نحن بتحديد ما إذا كنا نريدها صحيحة:-

```
> x<-as.integer(1)
> typeof(x)
[1] "integer"
> typeof(y) # "double"   is the default type of numerical var
iables
[1] "double"
```

والمتغير الثالث عبارة عن عدد مركب:-

```
> typeof(z)
[1] "complex"
```

ويمكننا العمل بالجزء الحقيقي أوالتخيّلي من العدد المركب:

```
> Re(z) # display the real part
[1] 2
> Im(z) # imaginary part
[1] 3
```

(ب) المتغيرات المنطقية

كون إما بقيمة"TRUE" أو "FALSE" قطعا. تصلح هذه المتغيرات كما سنرى لاحقا في عمل الاختبارات وفي التحكم في تدفق البيانات بالجمل الشرطية "conditional statements"، حيث الإجابة تكون بنعم أو لا:

مثال :

```
> x<-2
> x==2 # is 'x' equal to 2 ?
[1] TRUE
```

تعاملنا هنا مع علامات المقارنة "==" وهذا يعني أننا نريد أن نعرف هل قيمة إن المتغير "x" مساوية تماما ل2 أم لا. الإجابة ستكون بالطبع بنعم وهكذا قمنا بإنشاء متغير منطقي بقيمة "TRUE"

سنقوم الآن بتفقد نوع المتغيرالذي قمنا بأنشائه :-

```
> typeof(x==2)
[1] "logical"
```

لاحظ اعطى نوع المتغير هو منطقي

(ج) المتغيرات الرمزية String Variables:

تشبه في تركيبها المتغيرات العددية والفرق الوحيد بينهما هو أن قيمة المتغير الرمزي تكون رمزية (محصورة بين علامتي اقتباس).

مثال:

```
> s<-"Hello"
> class(s)
[1] "character
```

ملاحظة: التعابير في الطرف الأيمن لا يكون لها قيم حسابية لو استخدمت في عمليات حسابية لأنها موضوعة داخل " ".

هناك بعض القواعد الواجب مراعاتها عند كتابة اسم المتغير وهي:

1. لا يمكن استخدام الكلمات المفتاحية (الكلمات المحجوزة) أو الدوال التي توفرها اللغة كأسماء متغيرات، مثال:

   if, cos, for, break, else, return, function, sin, log, …

2. أسماء المتغيرات حساسة لحالة الحرف (COST, CoST, cost, Cost متغيرات مختلفة, وكذلك A و a).
3. يمكن لأسماء المتغيرات أن تحوي 63 رمزا وسيهمل أي رمز زائد عن 63.
5. يجب أن تبدأ أسماء المتغيرات بحرف متبوعا بأي عدد من الأرقام أو الأحرف أو النقطة او underscore .
6. جميع أوامر R تكتب بالحروف الصغيرة (if, while, for, …).

هناك عدة أنواع من المتغيرات في لغة R وهي:

5-2 التعبير الحسابي

يتكون التعبير الحسابي من مجموعة من الثوابت والمتغيرات تجمع بينهما عمليات حسابية ويستخدم فيها الرموز الحسابية مثل +، -، /، *، ^ والأمثلة آلاتية تعبر عن تعابير جبرية صيغت بلغة MATLAB.

التعبير الجبري	التعبير بلغة R
a – 3b	a – 3 ∗ b
$c^2 - 10$	c ^ 2 – 10
$a^2 + b^2$ / 12	a ^ 2 + b ∗∗ 2) / 12
m (7d – 8g)	m ∗ (7 ∗ d – 8 ∗ g)

ملاحظة: ان علامة الرفع لاس تتم بطريقتين في لغة R الاولى(^) والثانية(**)

امثلة :

```
> sqrt(2) # square root
 [1] 76
 > cos(pi) # cosine of pi, pi is the ‘π’ constant
 [1] -1
 > sin(20)^2+cos(20)^2
 [1] 1
 > log(1) # natural log
 [1] 0
 > log10(10) # decimal log
 [1] 1
 > exp(0) # exponential
 [1] 1
```

لاحظ هنا أمرين:

أولاً، أجريت العملية الحسابية ولكن وقع تجاهل كل ماهو مكتوب بعد العلامة "#" لأنه يعتبر تعليقا. كتابة التعليقات مهمة جداً في أي لغة برمجة .

ثانيًا، طبعت قبل النتيجة العلامة "[1]" وهذا لأن R يعتبر افتراضيا كل شيء بمثابة جدول "vector" والرقم واحد هو مؤشر عن العنصر الأول في الجدول.

1-5-2 قاعدة الأسبقية (الأولوية) Rule of Precedence

وهذه القاعدة مهمة في فهم وترتيب أولويات العمليات الحسابية في التعابير والمعاملات الحسابية، كما يجريها وينفذها الحاسب، وتنص القاعدة على أن الأولوية الأولى تعطى للعمليات الموجودة بين القوسين ومن اليسار إلى اليمين, وبالنسبة للعمليات الحسابية فالرفع إلى الأس أولا, والضرب (أو القسمة) ثانيا، والجمع (أو الطرح) أخيراً والمثال التالي يوضع هذه القاعدة:

مثال : التعبير:

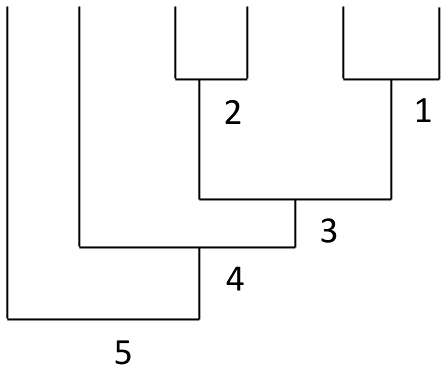
$\underbrace{\underbrace{A / B}_{1} + C}_{2}$ يكافئ في الجبر $\frac{A}{B} + C$

بينما يكافئ التعبير $\underbrace{A / (\underbrace{B + C}_{1})}_{2}$ يكافئ في الجبر $\frac{A}{B + C}$

لان الجمع داخل الأقواس يجري أولاً حسب الأولوية ثم يقسم A على نتيجة القوس.

مثال: التعبير

A - B / (K * F - X ^ M)



تنفيذ العمليات حسب الخطوات التالية:

تأخذ الأقواس الأولوية الأولى، وتنفذ العمليات داخلها حسب الأولوية أيضا.

العملية الأولى: رفع X إلى الأس M لتصبح كمية واحدة.

العملية الثانية: ضرب K في F لتصبح كمية واحدة.

العملية الثالثة: طرح نتيجة العملية الأولى من نتيجة العملية الثانية وتصبح النتيجة كمية واحدة.

العملية الرابعة: تقسم B على نتيجة العملية الثالثة وتصبح النتيجة كمية واحدة.

العملية الخامسة: تطرح نتيجة العملية الرابعة من A وتصبح النتيجة كمية واحدة.

2-5-2 الجملة الحسابية Arithmetic Statement

الجملة الحسابية في MATLAB تكافئ المعادلة الحسابية في الجبر إلا أن MATLAB تشترط أن يكون اسم المتغير المراد حساب قيمته في الطرف الأيسر وحده بدون أشاره بينما يكون التعبير الحسابي (بقية المعادلة) في الطرف الأيمن، كما في الأمثلة التالية:

1) y = A * X + B

2) A = 3.14 * R ^ 2

مثال:

أولوية العمليات الحسابية في الجمل الحسابية:

$$\underbrace{Z = \underbrace{A - \underbrace{B / C}_{1}}_{2}}_{3}$$

يمكن ملاحظة أن إشارة المساواة تمثل آخر أولوية حسابية بعد انتهاء جميع العمليات الحسابية في الطرف الأيمن.

6-2 الدوال المكتبية Library Functions:

يتوفر في معظم الحاسبات باستخدام لغة R اقترانات رياضية يكثر استعمالنا لها، مثل الدوال والاقترانات المثلثية واللوغارتيمية وغيرها ويمكن استدعائها في أي وقت، ومنها:

الدالة	الوصف
sqrt	الجذر التربيعي
abs	القيمة المطلقة
exp	المرفوع إلى قوة بأساس 10
log	اللوغاريتم الطبيعي
log 10	اللوغاريتم العشري
log 2	اللوغاريتم ذو الأساس 2
sin	جيب الزاوية
cos	جيب تمام الزاوية
tan	ظل الزاوية
atan	ظل معكوس الزاوية
floor	التدوير باتجاه اللانهاية السالبة
ceiling	التدوير باتجاه اللانهاية الموجبة
round	التدوير باتجاه أقرب عدد صحيح
sum	ايجاد المجموع لعدد من القيم

sign	إشارة العدد إذا كانت موجبة يعطي 1, سالبة يعطي -1, صفر
pi	النسبة الثابتة
re	الجزء الحقيقي
im	الجزء التخيلي
factorial	يعطي مضروب العدد

مثال:

```
> x <- 2.6;
>  y1 <- floor (x); y2 <- ceiling (x); y3 <- round (x);
y1
[2]
y2
[2]
y3
[3]
```

الدالة b<-ceiling(a) يدور عناصر a إلى أقرب عدد باتجاه اللانهاية أو بمعنى آخر يدور عناصر المصفوفة a إلى أقرب عدد صحيح أكبر أو يساوي إلى عناصر a و من أجل العناصر العقدية يتم تدوير القسم التخيلي و القسم الحقيقي كلاً على حدة.

مثال:

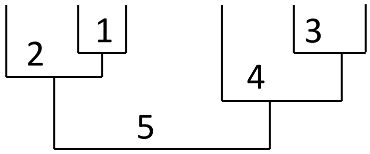
```
> a<-c(-1.9,-0.2,3.4,5);
> b<-ceiling(a)
>b
[1] -1  0  4  5
```

ملاحظة:

تأخذ الاقترانات المكتبية أولوية بعد الأقواس عند تنفيذ العمليات الحسابية.

مثال:

sin (a + b) – m / sqrt (d)



يكون تنفيذ العمليات الحسابية كما يلي:

العملية الأولى: إيجاد قيمة جمع a مع b.

العملية الثانية: إيجاد قيمة جيب الزاوية لناتج العملية (1).

العملية الثالثة: إيجاد قيمة الجذر التربيعي لــ d.

العملية الرابعة: إيجاد ناتج قيمة ناتج قسمة m على ناتج العملية (3).

العملية الخامسة: طرح ناتج العملية (4) من ناتج العملية (2) وتصبح النتيجة النهائية كمية واحدة (عدداً واحداً)

مثال: تمثل الجمل التالية إقترانات مكتبية في الجبر وإزائها قيمتها في R:

$b = \sqrt{a^2 + 10}$	⟸	b <- sqrt (a ^ 2 + 10)
z = ln (cx + ny)	⟸	z <- log (c * x + n * y)
y = sin 3 (x + nk)	⟸	y <- (sin (x + n * k)) ^ 3
s = tan-1 (y / x)	⟸	s <- atan (y / x)
$r = 2\sqrt{e^{x-5}}$	⟸	r<-2 * sqrt (exp (x – 5))
$t = \left\|x - \sqrt{y}\right\| / (a+m)$	⟸	t <- abs (x – sqrt (y)) / (a +m)

الفصل الثالث
ايعازات الادخال والاخراج (input and output)

1-3 المقدمة

لتغذية الحاسبة بالبيانات تستخدم ايعازات خاصة للإدخال وللحصول النتائج تستخدم ايعازات اخرى للإخراج , كما يمكن ان تتغذى الحاسبة بالبيانات و المعلومات مباشرة عن طريق الادخال المباشر بلوحة المفاتيح حيث تكون المتغيرات في الطرف الايسر من العبارات .

3- 2 ايعازات عرض وقراءة البيانات في لغة R

يتم استيراد البيانات وقراءتها من مصادرها وإن تعددت تنسيقات وصيغ تلك المصادر لدى لغة R أيضا المزيد من تعليمات الاستيراد التي تختص كل منها بتنسيق مختلف، فعلى سبيل المثال لا الحصر نذكر الأوامر التالية: read.spss و read.systat و read.mtp و read.xport .

1-2-3 الايعاز read

يعتبر هذا الايعاز احد ايعازات ادخال البيانات للحاسبة وتستخدم في البرنامج لإدخال البيانات المقابلة لقيم المتغيرات .وتستخدم لقراءة البيانات بالاشكال والامتدادات المختلفة وهي كالاتي:

2-2-3 الايعاز read.csv

يستعمل هذا الايعاز لقراءة البيانات ذات امتداد csv وهي بيانات بشكل جداول مستخرجة من برنامج الاكسل. والصيغة العامة للايعاز هي:

```
read.csv(file,choose(), header=TRUE, sep=",")
```

اذ ان :
read.csv: يمثل الايعاز
file,choose() : يظهر لنا نافذة يتم من خلالها اختيار الملف المطلوب للعرض.

حيث عند تنفيذ الايعاز تظهر نافذة يتم من خلالها اختيار الملف المخزون والذي نرغب بقراءته
ملاحظة: بعض الايعازات لم توجد في لغة R وتستطيع تحميلها من شبكة الانترنيت حيث توجد بداخل الحزم .

3-2-3 ايعاز read.spss

يستعمل هذا الايعاز لعرض بيانات ذات امتداد sav وهي بيانات بشكل جداول مستخرجة من البرنامج الاحصائي spss حيث يتضمن الجدول اسماء المتغيرات بشكل اعمدة ويتم قراءتها والصيغة العامة للايعاز هي:

```
library(foreign)

read.spss("filename.sav")
```

اذ ان :
read.spss : يمثل الايعاز

filename.sav : يمثل اسم الملف المطلوب عرض بياناته.

ملاحظة : يجب تنزيل الحزمة foreign لكي نستطيع العمل بهذا الايعاز.

4-2-3 ايعاز read_excel

يستعمل هذا الايعاز لعرض بيانات من نوع اكسل والشكل العام للايعاز هو :

```
library(readxl)

read_excel(path)
```

اذ ان:
read_excel : يمثل الايعاز

path : يمثل المسار المخزون فيه البيانات :

ملاحظة: لكي نستطيع العمل بهذا الايعاز يجب تنزل الحزمة readxl وان الايعاز library(readxl) يستعمل لاستدعاء جميع الاوامر التي تتضمنها هذه الحزمة واختيار الامر المطلوب منها.

مثال : اعرض جدول بيانات اكسل TEA ؟

الحل:

```
library(readxl)
TEA <- read_excel("C:/Users/Arshad/Desktop/TEA.xlsx")
```

3-3طريقة الحصول على المساعدة في لغة R

نتعرف على طريقة الحصول على المساعدة ، إذ يتدرج الأمر من طلب الحصول على المساعدة الخاصة بأمر محدد أو دالة بعينها، وذلك بذكر اسم الأمر أو الدالة عقب علامة الاستفهام ومن ثم النقر على زر الإدخال، فمثلا يقوم الأمر read.table بعرض الصفحة الخاصة بتوثيق التعليمة read.table ضمن ملفات المساعدة الخاصة بلغة R. أما إن أردت البحث عن مفهوم معين أو كلمة مفتاحية ما دون أن تعلم تماما أي الدوال هي التي تتعامل معها في لغة R، فيمكنك استخدام الأمر help.search("data ") input لتعرض عليك بعدها مجموعة من الأوامر ذات الصلة بهذا المفهوم، وتستطيع حينها الحصول على شرح أو مساعدة تفصيلية لأي من تلك الدوال بالطريقة التي أشرنا إليها سابقا. هناك وسيلة مساعدة أخرى متوفرة في لغة R موجهة إلى فئة المبرمجين الذين يفضلون رؤية الأمثلة وهي تعمل على أن يقرؤوا العشرات من أسطر ملفات المساعدة، وهؤلاء يمكنهم استخدام الأمر example بعد أن تمرر له اسم الدالة المراد الحصول على أمثلة عملية عن طريق استخدامها، فعلى سبيل المثال يمكنك تجربة الأمر example(mean)

4-3 التعليقات في لغة R

ففي لغة R التعليقات هي كل نص يتلو الرمز # سواء ظهر من بداية السطر أو جاء بعد تعليمة ما، 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).

5-3 عملية الإسناد في لغة R

يشير الى عملية الاسناد في لغة R بالرمز <- وهي الطريقة الأكثر شيوعا مقارنة برمز المساواة = والذي يصح استخدامه على الرغم من عدم شيوعه بين معشر المبرمجين بلغة R، إن البيانات المقروءة سيتم حفظها ضمن إطار بيانات (dataframe) أسميناه في حالة مثالنا السابق data، ويمكنك استعراض محتويات إطار البيانات ذلك بمجرد كتابة اسمه ومن ثم النقر على زر الإدخال ضمن سطر الأوامر، أما إن كانت كمية البيانات ضخمة فمن المفيد استخدام أي من الأمرين head(data) والذي يعرض مجموعة من الأسطر مقتطعة من بداية كتلة البيانات، أو الأمر tail(data) والذي يعرض مجموعة أخرى من الأسطر مقتطعة من نهاية كتلة البيانات ذاتها. كذلك تستطيع استخدام الأمر التالي:

```
data <- edit(data)
```

نستطيع الوصول بكل سهولة إلى أي جزئية في إطار البيانات الحالي من خلال المرونة التي تتيحها لنا لغة R، فلو كان لدينا إطار عمل يدعى data على سبيل المثال، فإن التعبير data[i,j] سيشير إلى العنصر أو القيمة الموجودة في السطر i والعمود j، أما التعبير data[i,] فيشير إلى كامل السطر i في حين أن التعبيرdata[,n:m] فيشير بدوره إلى مجموعة الأعمدة بدءا من n حتى m، من جهة أخرى فإن التعبير data[-,] i فيشير إلى كامل البيانات ضمن data فيما عدى السطر i، وأخيرا فإن التعبير data[c(n,m),] فهو يشير إلى السطرين n و m تحديدا دون غيرهما من أسطر البيانات في data.

6-3 ايعازات الاخراج

1-6-3 ايعاز print

يستخدم هذا الايعاز لاستخراج النتائج الخاصة بالبرنامج على الشاشة والشكل العام للايعاز هو

```
print(x)
```

حيث ان x يمثل المتغير الذي نرغب بظهور نتيجته على الشاشة

في لغة R تستخدم الفاصلة المنقوطة للفصل فيما بين كل أمر من أوامر اللغة الموجودة على سطر واحد (فيما لاحاجة لتلك الفواصل المنقوطة إن كانت كل تعليمة ترد ضمن سطر مستقل بها)، كما ترى فإن خرج تنفيذ أي أمر أو دالة بلغة R يظهر بعدها مباشرة، وهكذا تتكون جلسة العمل الاعتيادية من تنفيذ لتتالي من الأوامر والتعليمات وصولا إلى إنجاز العمل أو التحليل المطلوب.

ملاحظه:

تستخدم علامة الاقتباس (" ") في ايعاز print وذلك لطباعة النصوص المرغوب بها والتي تتناسب مع البيانات المطبوعة

مثال

```
> print("good")
[1] "good"
```

7-3 ايعاز تشغيل البرنامج Run

يستخدم هذا الايعاز لتنفيذ البرنامج او جزء من البرنامج فعندما نحدد البرنامج بكامله ثم الضغط على ايعاز Run الموجود في شريط الادوات للبرنامج سيتم تنفيذ جميع الخطوات للبرنامج واذا لم نحدد سيتم تنفيذ خطوة واحدة في كل مرة نضغط فيها على ايعاز Run.

8-3 ايعاز الخزن save وايعاز تحميل load

الايعازين حفظ (save) وتحميل (load) سوف تستخدم في لغة R ايعاز save لحفظ كائن R واستعادة هذا الكائن مرة أخرى باستخدام ايعاز load عند تحميلها يتم استعادة كائن بنفس الاسم كان عند حفظها.

والشكل العام لهما هو :-

```
save(x1,x2,x3,…. file = ("file' name"))
```

x1,x2,x3 تمثل اسماء المتغيرات

وان ("file' name") يمثل اسم الملف الذي نرغب بحفظه

مثال:

```
x <- stats::runif(20)
y <- list(a = 1, b = TRUE, c = "oops")
save(x, y, file = "xy.RData")
```

حيث x المتغير الاول و y المتغير الثاني و تم خزن الملف باسم xy.RData

اما شكل ايعاز load هو

```
load(file, envir = parent.frame(), verbose = FALSE)
```

مثال

اذا كانت لدينا الصيغة الخاصة لتباين عينة عشوائية $x_1.........x_n$ هي مساوية الى

$$s^2 = \frac{1}{1-n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

اكتب برنامج لحساب التباين ولعينة حجمها 11 مشاهدة؟

الحل :

البرنامج :

```
Untitled1*
Source on Save
## حساب تباين العينة#
x <- 1:11
 mean(x)
var (x)#" باستخدام الدالة الجاهزة"
V= sum((x - mean(x))^2)/ 10;#" باستخدام الصيغة الحسابية"
print(V)# "ايعاز طباعة قيمة المتغير V"
```

وتكون نتيجة البرنامج كالاتي:

```
> ## حساب تباين العينة#
> x <- 1:11
> mean(x)
[1] 6
> var (x)#" باستخدام الدالة الجاهزة"
[1] 11
> V= sum((x - mean(x))^2)/ 10;#" باستخدام الصيغة الحسابية"
> print(V)# "ايعاز طباعة قيمة المتغير V"
[1] 11
```

ان قيم x تم تعريفها وهي القيم من 1,2.......,11 حيث تم استخدام : (colon) لتحديد متجه x

نلاحظ من خلال البرنامج في البدء تم حساب المتوسط الحسابي بواسطة دالة خاصه به وهي *mean(x)* وكذلك تم حساب التباين بواسطة دالة خاصة وهي *var(x)* وكذلك تم استخدام الصيغة الخاصة بتباين العينة ونلاحظ استخراج نفس النتيجة في الطريقتين .

سؤال /لنفس المثال اعلاه اكتب برنامج لحساب تباين العينة بدون استخدام الدالة $var(x)$ بالاعتماد على الصيغة الاتية

$$s^2 = \frac{1}{n-1}(\sum_{i=1}^{n} x_i - n\bar{x})^2$$

مثال : عينة عشوائية بحجم (n= 100)مفردة فاذا كان الوسط الحسابي يساوي (50 =ME) ولانحراف المعياري (s=2) وتم حساب قيمة اختبار t ‏(t= 1.69).اكتب برنامج لحساب حدود الثقة والتي يتم حسابها وفق الصيغة الاتية :-

$$CU = ME - t.s/\sqrt{n}$$
$$CL = ME + t.s/\sqrt{n}$$

الحل:

```
Untitled1*
Source on Save
n=100#حجم العينة
s=2#الانحراف المعياري
t=1.69#قيمة اختبار t
ME=50
CL=ME-t*s/sqrt(n)#الحد الادنى
CU=ME+t*s/sqrt(n)#الحد الاعلى
print(CL)#اطبع الحد الادنى
print(CU)#اطبع الحد الاعلى
```

من خلال البرنامج نلاحظ ان قيمة الحد الاعلى CU=50.338 والحد الادنى CL=49.662

مثال : ارسم خارطة السير واكتب برنامجا لحساب جذور المعادلة التربيعية باستخدام صيغة الدستور

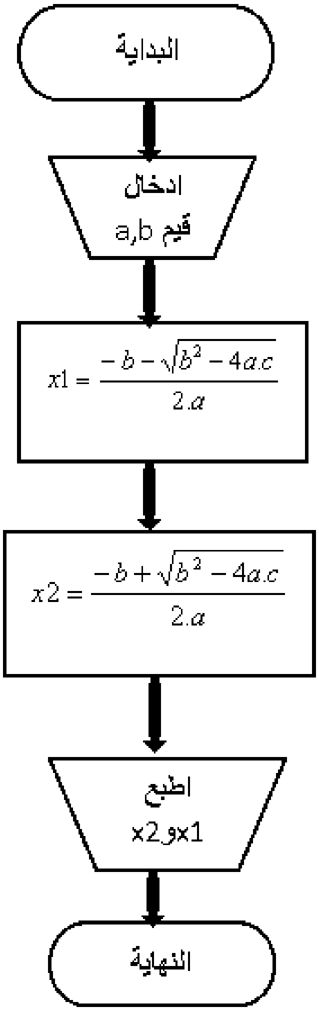
حيث :-

$$x = \frac{-b \pm \sqrt{b^2 - 4a.c}}{2.a}$$

الحل:

الخوارزمية:-

1- البداية
2- ادخل قيم a,b
3- احسب x1 و x2
4- اطبع x1 و x2
5- النهاية



البرنامج :- اذا كانت $a=2\ ,b=5\ ,c=3$

```
Untitled1*
Source on Save

#####ايجاد جذور المعادلة التربيعية باستخدام الدستور######
a<-2
 b<-5
 c<-3
 x1=(-b-sqrt(b**2-4*a*c))/(2*a)#قيمة الجذر الاول
 x2=(-b+sqrt(b**2-4*a*c))/(2*a)#قيمة الجذر الثاني
 print(x1)#اطبع المجذر الاول
 print(x2)#اطبع الجذر الثاني
```

وتكون نتيجة تنفيذ البرنامج كالاتي:

```
Console ~/
> #####ايجاد جذور المعادلة التربيعية باستخدام الدستور######
> a<-2
> b<-5
> c<-3
> x1=(-b-sqrt(b**2-4*a*c))/(2*a)#قيمة الجذر الاول
> x2=(-b+sqrt(b**2-4*a*c))/(2*a)#قيمة الجذر الثاني
> print(x1)#اطبع المجذر الاول
[1] -1.5
> print(x2)#اطبع الجذر الثاني
[1] -1
```

من البرنامج نلاحظ ان عملية المساوة مرة كتبت بالشكل (=) ومرة اخرى (->) ونلاحظ ان عمية الرفع للاس ايضا كتبت بدل ^ العلامة ** وفي كلا الحالتين يمكن استخدام اي رمز سواء في حالة المساواة او في حالة الرفع الى الاس.

$X2=-1$ $X1=-1.5$

مثال: اكتب الخوارزمية مع برنامجا كاملا لقراءة درجات الطالب في مرحلة السادس العلمي في دروس اللغة العربية A واللغة الانكليزية E والرياضيات M والفيزياء P والكيمياء Cوالاحياء B ثم ايجاد وطبع معدل ذلك الطالب ثم انشاء الدالة تمثل المعدل

الحل:

الخوارزمية :-

1- البداية
2- ادخال درجات الطالب اي اقرأ قيم A,E,M,P,C,B
3- جد حاصل جمع قيم الدرجات واجل الناتج مساوي الى قيمة s اي
S=A+E+M+P+C+B
4- اجعل av=s/6
5- اطبع معدل الطالب اي اطبع
قيمة (av)
6- النهاية

البرنامج:-

```
Untitled1*   Untitled2*
Source on Save
av<-function(A,E,M,P,C,B){
  s=A+E+M+P+C+B;
  avv=s/6
  return(avv)
}
```

وتكون نتيجة البرنامج كالاتي:

```
Console ~/
> av(90,67,89,86,90,67)
[1] 81.5
>
```

نلاحظ من خلال البرنامج اعلاه تم انشاء دالة جديدة باسم av وبالمدخلات التي تمثل درجات الطالب حيث ان 90 تمثل درجة اللغة العربية المتمثلة بالرمز A و 67 تمثل درجة اللغة الانكليزية المتمثلة بالرمز E وان 89 تمثل درجة الرياضيات المتمثلة بالرمز M وان 86 تمثل درجة الفيزياء و 90 الكيماء و67 الاحياء وكان المعدل لهذا

الطالب 81.5 من خلال المخرجات بدلالة الدالة return ولان بإمكانك ادخال اي درجات من خلال الدالة av لتحصل على معدل جديد.

مثال : اكتب الخوارزمية وبرنامج يقوم بحساب قيمة C من المعادلة $c = \frac{AB}{A+B}$ علما بان قيمة A=6 و قيمة B=3.

الحل:

الخوارزمية

1- البداية
2- ادخل قيمة A,B
3- اجعل C=(A*B)/(A+B)
4- اطبع قيمة C
5- النهاية

البرنامج:

```
A=6
B=3
C=(A*B)/(A+B)
print(C)
```

وعند تنفيذ البرنامج تكون النتيجة كالاتي:

```
> A=6
> B=3
> C=(A*B)/(A+B)
> print(C)
[1] 2
```

نلاحظ انه تم حساب قيمة C = 2 وتم طباعة الناتج كما في اعلاه.

مثال : اكتب خوارزمية و برنامج لإيجاد مساحة دائرة من المعادلة
$area = r^2\pi$ حيث r يمثل نصف القطر

الخوارزمية

1- البداية
2- ادخل قيمة r
3-
4- اجعل area= r^2*pi
5- اطبع area
6- النهاية

من البرنامج نلاحظ ان مساحة الدائرة مساوية الى 0.1963495

البرنامج:-

```
r=0.25
area=r^2*pi
print(area)
```

وعند التنفيذ تكون النتيجة كالاتي:

```
> r=0.25
> area=r^2*pi
> print(area)
[1] 0.1963495
>
```

مثال: اكتب برنامج مع الخوارزمية لإيجاد الحد الاخير للمتوالية العددية ومجموع الحدود اذا كانت A تمثل الحد الاول و D تمثل اساس المتوالية و N عدد الحدود احسب الحد الاخير L ومجموع الحدود C من الصيغ التالية :-

L=A+(N-1)D

C=N/2(A+L)

الحل:

الخوارزمية :-

1- البداية
2- ادخل قيم N,D,A
3- اجعل L=A+(N-1)D
4- اجعل C=N/2(A+L)
5-
6- اطبع C,L
7- النهاية

البرنامج :

```
Untitled1*
Source on Save
N=20
D=2
A=0.5
L=A+(N-1)*D
C=N/2*(A+L)
print(L)
print(C)
```

وعند التنفيذ يكون الناتج :

```
Console ~/
> N=20
> D=2
> A=0.5
> L=A+(N-1)*D
> C=N/2*(A+L)
> print(L)
[1] 38.5
> print(C)
[1] 390
>
```

الحد الاخير L=38.5 , مجموع الحدود C=390

الفصل الرابع
عبارات التحكم Control statement

1-4 المقدمة

ان جميع البرامج التي تكتب بلغة R هي برامج عباراتها متسلسلة وتنفذ وفق تسلسل ثابت اي تنفذ الحاسبة هذه العبارات الواحدة بعد الاخرى دون ان تتم اي عملية قفز الى عبارات تالية او العودة للعبارات السابقة وهذه الحالة تكون في البرامج الاولية والبسيطة ولكن في اغلب المسائل يقتضي حلها بالرجوع الى عبارات سابقة او التكرار لعملية معينة لأكثر من مرة او الاتجاه في مسار معين من بين عدد من المسارات عند نقطة معينة وفق شروط محددة. مما تقدم نجد ان التعبير عن الشروط بلغات البرمجة من الامر المهمة وفي لغة R يعبر عن الشرط بتعبير منطقي Logical expression وقبل التطرق الى التعابير المنطقية لابد من اعطاء فكرة عن العوامل المنطقية .

2-4 العوامل المنطقية Logical operator

لأجل القيام بعملية التفرع لابد من ايجاد وسيلة للتعبير عن الشرط أن يكون شرط المساواة او لامساواة او الاكبر من او الاصغر من او الاكبر والمساواة او الاصغر والمساواة ,يتم ذلك من خلال استخدام عدد من العوامل العلائقية وهي :

الرمز بلغة R	العامل العلائقي
<	أصغر من
<=	أصغر أو يساوي
>	أكبر من
>=	أكبر أو يساوي
==	إشارة المساواة
!=	اشارة عدم المساواة

ملاحظة:

لاحظ بان الإشارتين (=) و (==) تعنيان شيئاً مختلفاً, حيث تستخدم (==) للتعبير المنطقي الذي يمثل شرط المساواة , بينما تستخدم (=) لإسناد إخراج العملية إلى متغير.

1-2-4 التعبير المنطقي البسيط

هو عبارة عن تعبيرين حسابيين يفصل بينهما عامل منطقي واحد والشكل العام للتعبير المنطقي البسيط هو

Expression logical operator expression

وتكون نتيجة التعبير المنطقي اما صائبة True او خاطئة False

2-2-4 التعبير المنطقي المركب

هو عبارة عن تعبيرين منطقيين بسيطين يفصل بينهما احد العوامل المنطقية الاتية:-

العامل المنطقي	الوصف
&	**AND (و)**
|	**OR (أو)**
!	**NOT (النفي)**

وفي حالة استخدام OR تكون العلاقة صائبة او خاطئة وفق الترتيب الاتي :

T OR T = T	**EXAMPLE**	**A>B OR B>C**
T OR F= T	**EXAMPLE**	**D>B OR B<C**
F OR T=T	**EXAMPLE**	**I>D OR D=I*2**
F OR F=F	**EXAMPLE**	**D=I OR B<1**

حيث T تمثل True و F تمثل False

وفي حالة استخدام AND تكون العلاقة وفق الترتيب الاتي :

T AND T = T	**EXAMPLE**	**A>B AND B<D**
T AND F = F	**EXAMPLE**	**A=K+4 AND A>B+D**
F AND T = F	**EXAMPLE**	**C=D AND C<K**
F AND F = F	**EXAMPLE**	**K>A AND K>D**

3-4 أسـبقـية الـعوامل

يقوم برنامج R بإيجاد قيمة تعبير مستنداً إلى مجموعة من القواعد الناظمة لأسبقية المعامل, وتحسب المعاملات ذات الأسبقية العليا قبل المعاملات ذات الأسبقية الدنيا, وتقيم المعاملات ذات الأسبقية المتساوية من اليسار إلى اليمين. ويشرح الجدول التالي قواعد أسبقية المعامل التي يعتدها برامج R.

العامل	مستوى الأسبقية
الأقواس ( )	**الأعلى**
القوة (^)	
إشارة النفي (!)	
الضرب (*), القسمة (/)	
الجمع (+), والطرح (-)	
معامل النقطتين المتعامدتين (:)	
أصغر من (<), واصغر أو يساوي(<=), اكبر من (>), اكبر من أو يساوي(>=), المساواة (==),عدم المساواة (!=)	
الجمع المنطقي(&) AND	
المعامل المنطقي (|) OR	**الأدنى**

4-4 الايعاز if

يستعمل هذا الايعاز لتنفيذ عبارة او عدد من العبارات على ضوء تحقيق شرط معين وغالبا ما يستخدم لتكوين التفرعات الثنائية وفق شروط محددة . لذلك هناك اشكال مختلفة للايعاز حسب الهدف من استعمالها في البرنامج وسنحاول ان نعض كل حالة.

1-4-4 الحالة الاولى if

تنفيذ عدد من العبارات (الاوامر)على ضوء شرط معين في هذه الحالة يكون الشكل العام للايعاز كالاتي :

```
if (logical expression) {statements}
```

if : يمثل الايعاز

logical expression : تعبير منطقي يمثل شرط وهذا التعبير يكون تعبير منطقيا بسيطا او مركبا .

statements : العبارات المطلوب تنفيذها عندما يتحقق الشرط

عمل الايعاز if

اذا كان التعبير المنطقي صائب True تنفذ العبارات {statements} اما اذا كان التعبير المنطقي خاطئ False لا تنفذ العبارات {statements} و ينتقل التنفيذ الى ما بعد ايعاز if .

بعض الامثلة الخاصة بالحالة الاولى :

مثال: اكتب برنامج لحساب قيمة Y اذا كان

$$Y = 2x \qquad if\ x > 2$$

الحل:

البرنامج :

```
1  x <-3
2   if(x>2){y=2*x}
3  print(y)
```

وعند التنفيذ يكون الناتج كالاتي:

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

نلاحظ من خلال البرنامج ان قيمة x=3

والشرط هو ان قيمة x اكبر من 2 هذا يعني العبارة صائبة True لذلك نفذت العبارات

y=2*3=6

في الايعاز نلاحظ ان التعبير المنطقي بسيطا لأنه يحتوي على عامل منطقي واحد وهو اشارة >

مثال :اكتب برنامج يقوم بحساب كل من y و z حيث

$Y = x^2 + 2x \qquad if \ \ x > 0$

$Z = \ x \ + 4x + 1 - x$

$Y = x + \ x\ 2 \ + 3 \qquad if\ x < 0$

$Z = \ x^{\ 2} \ + 1 - 2x \qquad if\ x = 0$

اطبع error

الحل:

البرنامج :

```
x=0.68
if(x>0){y=sqrt(x)+2*x;
z=sqrt(x)+4*x+1-x;}
if(x<0){y=x+2+3;z=sqrt(x)+1-2*x}
if(x==0){print("error")}
print(y)
print(z)
```

وعند تنفيذ البرنامج يكون الناتج كالاتي:

```
> x=0.68
> if(x>0){y=sqrt(x)+2*x;
+ z=sqrt(x)+4*x+1-x;}
> if(x<0){y=x+2+3;z=sqrt(x)+1-2*x}
> if(x==0){print("error")}
> print(y)
[1] 2.184621
> print(z)
[1] 3.864621
```

من خلال البرنامج نلاحظ ان ايعاز if ينفذ الشرط ذات العبارة الصائبة ويترك بقية العبارات لاحظ انه تم ادخال قيمة x=0.68 قيمة اقل من الصفر لذلك حقق الشرط عندما يكون x<0 ونفذ العبارات الخاصة بها .

ملاحظة : نافذة مساحة العمل console يتم من خلالها عرض النتائج عند تنفيذ الاوامر الخاصة بالبرنامج من خلال ايعاز Run

2-4-4 الحالة الثانية:- if-else

ويستخدم هذا الايعاز في لبناء التفرعات الثنائية والشكل العام للإيعاز هو :

```
if(conditions) {
 Statements1
 } else {
 Statements2
}
```

عمل الايعاز

اذا كان التعبير المنطقي "TRUE " تنفذ العبارات "Statments1" فقط وينتقل التنفيذ الى ما بعد ايعاز if اما اذا كان التعبير المنطقي خاطئا "FALSE" تنفذ العبارات "Statments2" فقط .

مثال : اكتب برنامج لإيجاد قيمة y المعرف حسب المعادلة التالية:

$$y = \begin{cases} 10, & x > 3 \\ 0, & otherwise \end{cases}$$

الحل:

البرنامج:

```
## Generate a uniform random number
x <- runif(1, 0, 10)
if(x > 3) {#الطريقة الاولى
   y <- 10
} else {
   y <- 0
}
print(y)

ifelse(x>3,10,0)#الطريقة الثانية

```

وعند تنفيذ البرنامج تكون مخرجاته كالاتي:

```
Console ~/
[1] 0
> ## Generate a uniform random number
> x <- runif(1, 0, 10)
> if(x > 3) {
+ y <- 10
+ } else {
+ y <- 0
+ }
> print(y)
[1] 0
> ifelse(x>3,10,0)
[1] 0
```

3-4-4 الحالة الثالثة:- اذا المتداخلة **(Nested if)**

تستخدم هذه الحالة لانشاء ثلاث تفرعات او اكثر حسب شروط معينة .والشكل العام للايعاز هو:

```
If(logical expression 1)

{ statments1}

else if (logical expression 2)

{statments2}

else if(logical expression 3)

{statments3}
```

عمل الايعاز:

تنفذ العبارات التي تسبقها علاقة منطقية صائبة وترك بقية العبارات لان علاقاتها المنطقية خاطئة وتكتب بالشكل اعلاه فقط .

مثال: ارسم خريطة سير العمليات ثم اكتب برنامج لحساب قيمة W طبقًا للمعادلات الآتية علمًا بأن قيمة المتغير X معطاة معلومة ونلاحظ لا يعمل الايعاز الا بالطريقة المكتوبة .

$$w = \begin{cases} x^2 + 1 & if\ x > 0 \\ 5 + x & if\ x = 0 \\ 2x^3 - 1 & if\ x < 0 \end{cases}$$

الحل:

الخوارزمية :

1- البداية

2- اقرأ قيمة المتغير X.

3- إذا كانت X أكبر من صفر فاذهب إلى الخطوة 4 أما إذا كانت ليست أكبر من فاذهب إلى خطوة 5.

4- احسب W من المعادلة (1) ثم اذهب إلى الخطوة 8.

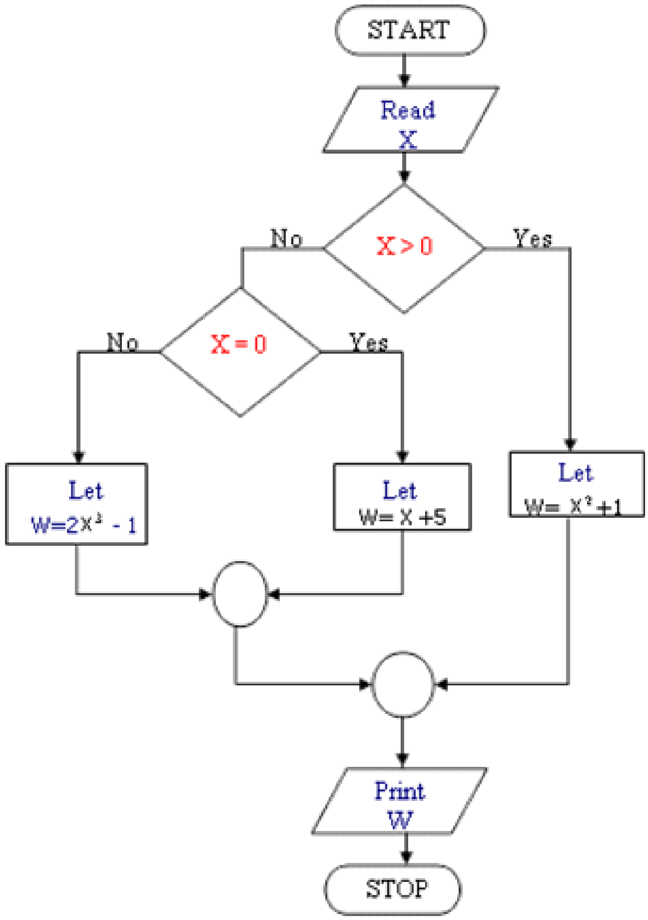
5- إذا كانت X تساوي صفر فاذهب إلى الخطوة 6 وإلا فاذهب إلى الخطوة 7.

6- احسب W من المعادلة (2) ثم اذهب إلى الخطوة 8.

7- احسب W من المعادلة (3) ثم اذهب إلى الخطوة 9.

8- اطبع قيمة w

9- النهاية



البرنامج:

Untitled1*

Source on Save

```
x<-4
if(x>0){
  w=x^2+1
}else if(x==0){
  w=5+x
}else if(x<0){
w=2*x^2-1
}
print(w)
```

وعند تنفيذ البرنامج تكون مخرجاته كالاتي:

```
Console ~/
> x<-4
> if(x>0){
+ w=x^2+1
+ }else if(x==0){
+ w=5+x
+ }else if(x<0){
+ w=2*x^2-1
+ }
> print(w)
[1] 17
```

نلاحظ ان قيمة x التي تم ادخالها في البرنامج x=4 الي اكبر من الصفر لذلك حقق شرط (x>0) والعبارات الخاصة بها لذلك كان الناتج مساويا الى 17.

مثال: اكتب برنامج يقوم بطباعة Negative إذا كان العدد المدخل سالباً، ويطبع Positive إذا كان العدد المدخل موجباً، ويطبع صفر فيما عدا ذلك.

الحل:

```
rm(list=ls())
 x<- scan()
 if (x<0) {
print("Negative")
} else if(x>0)
{
 print("Positive")
} else
  print("Zero")
```

وعند التنفيذ يكون الناتج كالاتي:

```
Console ~/
> x<- scan()
1: 5
2:
Read 1 item
> if (x<0) {
+ print("Negative")
+ } else if(x>0)
+ {
+  print("Positive")
+ } else
+ print("Zero")
[1] "Positive"
```

نلاحظ عند ادخال العدد 5 طبع العبارة Positive لدلالة ان العدد 5 عدد موجب

ملاحظة: الايعاز scan يتم من خلاله ادخال قيم مباشرة الى console أي يكافئ الايعاز input في اللغات الاخرى مثل لغة MATLAB .

ملاحظة مهمة : يمكن كتابة if المتداخلة بشكل مدمج (elseif) وبالشكل العام لها هو

ifelse(condition, statments1,statments2)

مثال: اكتب برنامج يفحص المتجه (5,7,2,8) اذا كان العدد فردي يطبع odd واذا كان العدد زوجي يطبع even .

الحل:

```
rm(list=ls())
a <-c(5,7,2,8)
ifelse(a %% 2 == 0,"even","odd")
```

وعند التفيذ تكون مخرجات البرنامج كالاتي:

```
Console ~/
> rm(list=ls())
> a <-c(5,7,2,8)
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd"  "odd"  "even" "even"
> 
```

الفصل الخامس
جمل الدوران وحلقات التكرار

1-5 المقدمة

في كثير من الاحيان يتطلب حل المسالة تكرار اجراءات معينة لعدد من المرات حسب شروط معينة .ولاجل تكوين حلقات تكرار في البرمجة بلغة R وغالبا ما تكون التكرارات معلومة او محددة ولكن في بعض المسائل يكون العداد غير محدد وغالبا ما يعتمد التكرار على تحقيق شرط معين وتتوقف عملية التكرار في حالة عدم تحقق الشرط .ولتنفيذ هذه العملية نستخدم ايعازات خاصة لتكوين حلقات التكرار في برمجة R هي:

1- ايعاز for
2- ايعاز while
3- ايعاز repeat

2-5 ايعـــاز for loop

يستخدم هذا الايعاز لتكوين حلقات تكرار ذات العدد المعلوم من التكرارات والشكل العام للايعاز هو:

```
for (name in vector) {statements}
```

حيث ان :

for : دليل الايعاز

Name : اسم العداد (المتغير) والذي يكون مساويا الى عناصر المتجة في السلسلة

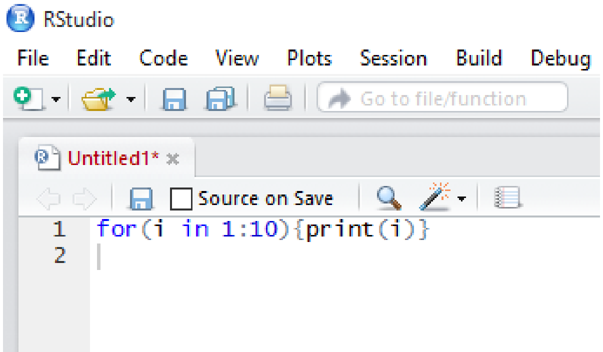
Vector : متجة يمثل سلسلة من القيم العددية التي يأخذها العداد

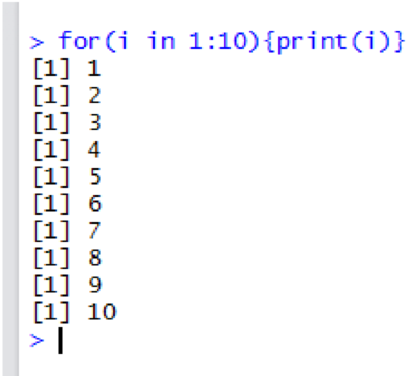
Statements : عبارات البرنامج المطلوب تكرارها في كل حلقة تكرار.

مثال : اكتب برنامج يطبع قيم المتغير (i) من 1 الى 10

الحل:

البرنامج :





نلاحظ ان (i) اسم المتغير العددي اي اسم العداد

1 القيمة الاولى للعملية

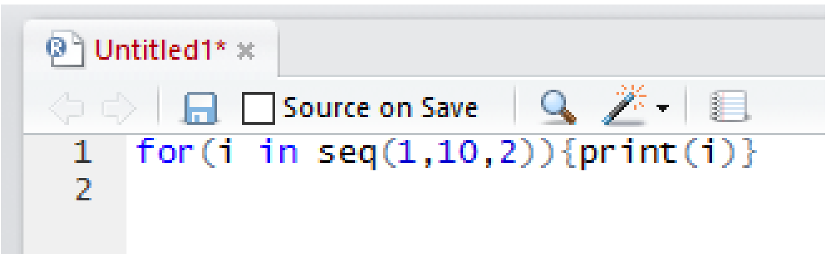
10 القيمة القصوى للعداد

Statements:- هي العبارات المطلوب تكرارها هي طباعة قيم المتغير (i) وان سلسلة المتجه هي القيم من 1 الى 10 .

ملاحظة : في بعض الاحيان نرغب بتكرار خطوات بسلسلة بحيث يتم القفز في العدد اي ليس بصورة متتالية وكما في المثال التالي

مثال : اكتب برنامج يطبع الاعداد الفردية المحصورة بين 1 و 10

الحل:



يكون ناتج التنفيذ كالاتي:

```
> for(i in seq(1,10,2)){print(i)}
[1] 1
[1] 3
[1] 5
[1] 7
[1] 9
```

من خلال البرنامج نلاحظ انه تم استخدام الدالة **seq** وذلك لتكوين سلسلة للمتغير i كعداد حيث يبدا من 1 وينتهي الى 10 بزيادة مقادرها 2 خطوة اي العداد يقفز بزيادة مقدارها 2

ملاحظة : لا يمكن عمل سلسلة عددية بزيادة معينة باستخدام colon اي (:) عكس بقية لغات البرمجة الاخرى .

مثال: ارسم خريطة سير العمليات ثم اكتب برنامج لإيجاد مساحة مجموعة من الدوائر أنصاف أقطارها معلومة:

الحل:
تكون خطوات الحل المبينة في الشكل الاتي:

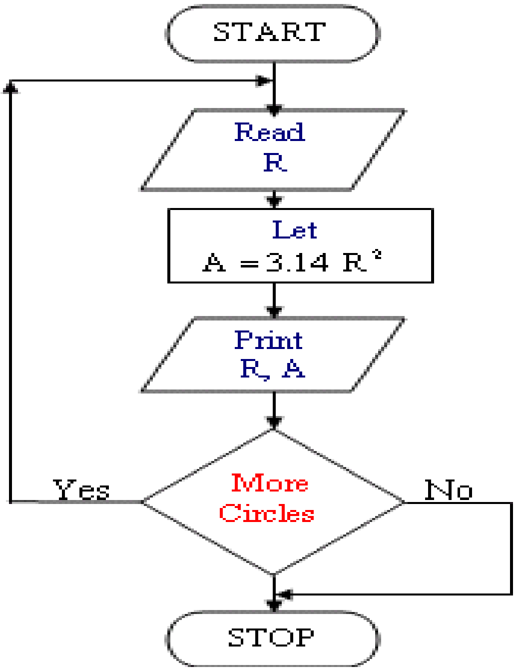
الخوارزمية وخارطة السير

1. ابدأ.
2. اقرأ نصف قطر الدائرة (R).
3. أوجد مساحة الدائرة (A).
4. اطبع قيم كل من A, R.
5. هل هناك مزيد من الدوائر؟

فإن كان نعم فعد إلى الخطوة(2) وإن كان لا فعد إلى الخطوة (6).

7. النهاية

خارطة سير العمليات



البرنامج:

```
rm(list=ls())
r=c(0.5,0.6,0.7,0.8,1);
n=5;
pi=22/7;
for(i in seq(1,5)){
  area<-pi*r^2
  }
print(area)
```

وعند تنفيذ البرنامج يكون الناتج هو :

```
Console ~/
> rm(list=ls())
> r=c(0.5,0.6,0.7,0.8,1);
> n=5;
> pi=22/7;
> for(i in seq(1,5)){
+   area<-pi*r^2
+   }
> print(area)
[1] 0.7857143 1.1314286 1.5400000 2.0114286 3.1428571
>
```

نلاحظ من البرنامج انه اخذ الدائرة الاولى وحسب لها المساحة بالنصف القطر الاول ثم اخذ الثانية وحسب لها المساحة واهكذا الى الدائرة الخامسة ثم توقف عن التكرار لانه حددنا في البداية عدد الدوائر 5 .

3-5 العداد : Counter

في كثير من الأحيان نحتاج في برامج الحاسب الالكتروني إلى العد Counting، فقد نريد مثلاً أن نعد عدد كل من الطلاب والطالبات ضمن الشعبة, وقد تكون هذه العملية سهلة للإنسان لأنها أصبحت ضمن قدراته العقلية التي يكتسبها من الطفولة، إلا أن الحاسب يحتاج إلى تصميم خوارزمية للعد Counting Algorithm تتضمن خطوات معينة إذا اتبعتها استطاع أن يعد.

ويمكن تحديد الخطوات التي يتبعها الحاسب حتى يتمكن من العد في الخطوات الأساسية:

1. اجعل العداد مساويًا للصفر.

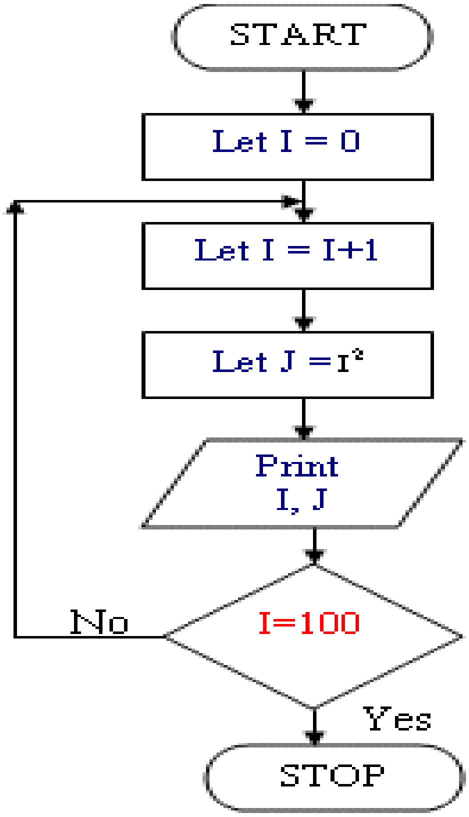
2. اجعل القيمة الجديدة للعداد تساوي القيمة القديمة لها زائد واحد, أي أن:

قيمة العداد (الجديدة)= قيمة العداد (القديمة)+1

3.كرر الخطوات ابتداء من الخطوة 2.

مثال: اكتب خوارزمية ثم ارسم خريطة سير العمليات ثم اكتب برنامج للعمليات التي يتبعها الحاسب لطباعة الأعداد الطبيعية من 1 إلى 100 ومربعاتها.

الحل: خطوات الحل مبينة في الشكل



الخوارزمية:-

8. ابدأ.
9. اجعل I=0.
10. اجعل I=I+1.
11. اجعل $J = I^2$.
12. اطبع J, I.
13. إذا كانت I=100 اذهب إلى الخطوة 7 وإلا اذهب إلى الخطوة 3.
14. النهاية

البرنامج :

```
Untitled1*
Source on Save
## STATRT ###
I=0
for (h in 1:100) {
I=I+1
J=I^2
print(I)
print(J)
}
### end ####
```

ويكون تنفيذ البرنامج كالاتي:

```
Console ~/
> ## STATRT ###
> I=0
> for (h in 1:100) {
+ I=I+1
+ J=I^2
+ print(I)
+ print(J)
+ }
[1] 1
[1] 1
[1] 2
[1] 4
[1] 3
[1] 9
[1] 4
[1] 16
```

من المثال اعلاه نلاحظ ان الفرق بين العداد Counter وايعاز for حيث ان ايعاز for قام بعملية التكرار والمتمثلة في الخوارزمية اعلاه خطوة 3 اي اخذ قيمة i ثم اجرا عليها عملية التربيع ثم طبع الناتج وهكذا اما counter فان عمله اقتصر على الانتقال الى العدد الثاني لكي يكرره ايعاز for اي في كل مرة يتقل الى العدد الذي يلي العدد السابق.

4-5 المجاميع الاجمالية ∑:

في كثير من الأحيان نحتاج في برامج الحاسب الإلكتروني إلى جمع مجموعة كبيرة من الأعداد التي تمثل معطيات ظاهرة معينة، فمثلاً قد نرغب في إيجاد الوسط الحسابي لأعمار طلاب الجامعة، ولتحقيق هذا أولاً يجب أن نحسب مجموع أعمار الطلاب، وطبعًا ليس عمليًا إعطاء رمز أبجدي لكل عمر طالب فقد تحتاج لأكثر من عشرة الآلاف رمز، في مثل هذه الحالات نصمم خوارزمية معينة للتجميع تسمى خوارزمية التجميع summers Algorithm تتضمن خطوات محددة إذا اتبعها الحاسب استطاع أن يجمع أي كمية من البيانات باستخدام متغيرين اثنين إحداهما هو المتغير الذي نجمعه والآخر هو الجمع الإجمالي (المجمع)، ويمكن تحديد الخطوات التي يجب أن يتبعها الحاسب لتحقيق ذلك في أربع خطوات هي:

1- اجعل المجمع مساويًا الصفر.

1- ادخل قيمة واحدة للمتغير.
2- اجعل القيمة الجديدة للمجمع تساوي القيمة القديمة له زائد القيمة المدخلة للمتغير، أي أن:
قيمة المجمع الجديدة=قيمة المجمع القديمة + آخر قيمة مدخلة للمتغير.
3- كرر ابتداءًا من الخطوة الثانية.

مثال: ارسم خريطة سير العمليات ثم اكتب برنامج لإيجاد الوسط الحسابي لأعمار طلاب شعبتك.

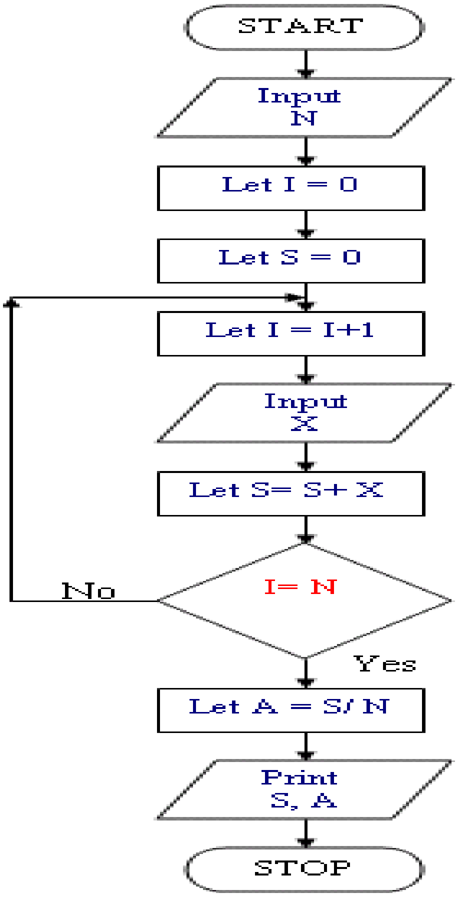
الحل:

نفترض أن إجمالي عدد الطلاب N ونستخدم عددًا لرقم كل طالب ونرمز له بالرمز I ونرمز لعمر الطالب بـX ونستخدم مجمعًا لأعمار الطلبة ونرمز له بالرمزS ونستخدم الرمز A ليدل على معدل أعمار الطلبة. وتكون خطوات الحل كما في ادناه:

الخوارزمية :

1. ابدأ.
2. ادخل إجمالي عدد الطلاب (N).
3. اجعل I=0.
4. اجعل S=0.
5. اجعل I=I+1.
6. ادخل X.
7. اجعل S=S+X.
8. إذا كانت I=N اذهب إلى الخطوة 9 وإلا اذهب إلى الخطوة 5.
9. اجعل A=S/N.
10. النهاية.

خارطة سير العمليات



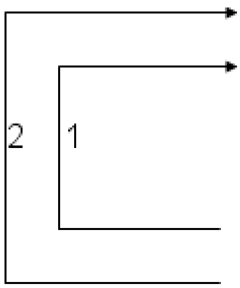
البرنامج :

```
rm(list=ls())
n=20
i=0
x=c(20,25,22,21,32,22,20,27,24,25,25,22,21,23,24,21,30,31,20,24)
  s=sum(x)
  i=i+1
a=s/n
print(a)
print(s)
```

يكون ناتج البرنامج هو 23.95 والمجموع 479

5-5 الدورانات المتدخلة

في هذه الحالة تكون الدورانات داخل بعضها البعض بحيث لا تتقاطع فإذا كان لدينا مثلاً دوران من هذا النوع انظر شكل ادناه فيسمى الدوران قم (1) دورانًا داخليًا (Inter Loop) بينما الدوران رقم (2) دورانًا خارجيًا (Outer Loop) ويتم التناسق في عملي مثل هذين الدورانين بحيث تكون أولوية التنفيذ للدوران الداخلي. ويجب ان يكون الدوران الداخلي اصغر من الدوران الخارجي كذلك يجب ان يكون المتغير العداد في الحلقات المتداخلة مختلف ولا يجوز استعمال نفس العداد لحلقتين كما يمكن استخدام اي عدد من حلقات التكرار المتداخلة .



مثال: الخوارزمية التالية كتبت لتوليد قيمة واحدة من متغير عشوائي x الذي يتبع توزيع بيتا بالمعالم n,m اي ان x~beta(n,m) نفذ البرنامج اذا n=2,m=4

الحل:

الخوارزمية

1. اجعل x1=0 , x2=0
2. اجعل i=1
3. ولد متغير عشوائي U~U(0,1)
4. اجعل i=i+1
5. اذا كان i≤ n+m ارجع للخطوة الثانية
6. $x = \frac{x1}{(x1+x2)}$

ملاحظة: لتوليد قيمة U في الخطوة الثانية استخدم الدالة unif(1)

البرنامج :-

```
Untitled1*   generatebeta.R*
Source on Save
rm(list=objects())
n<-3
m<-4
x1<-0
x2<-0
#for(j in 1:10){
for(i in 1:7){
   u<-runif(1)
if(i<=n){
    x1<-x1-log(u)
}else{x2<-x2-log(u)
}
}
x<-x1/(x1+x2)
print(x)
```

```
Console ~/
> rm(list=objects())
> n<-3
> m<-4
> x1<-0
> x2<-0
> #for(j in 1:10){
> for(i in 1:7){
+ u<-runif(1)
+ if(i<=n){
+ x1<-x1-log(u)
+ }else{x2<-x2-log(u)
+ }
+ }
> x<-x1/(x1+x2)
> print(x)
[1] 0.1475249
> 
```

اما لتوليد 10 قيم من المتغير اعلاه نتبع الاتي :

RStudio

File Edit Code View Plots Session Build D

Untitled1* generatebeta.R*

```
rm(list=objects())
n<-3
m<-4
x1<-0
x2<-0
for(j in 1:10){
for(i in 1:7){
  u<-runif(1)
if(i<=n){
   x1<-x1-log(u)
}else{x2<-x2-log(u)
}
}
x<-x1/(x1+x2)
print(x)
}
```

Console ~/

```
> rm(list=objects())
> n<-3
> m<-4
> x1<-0
> x2<-0
> for(j in 1:10){
+ for(i in 1:7){
+ u<-runif(1)
+ if(i<=n){
+ x1<-x1-log(u)
+ }else{x2<-x2-log(u)
+ }
+ }
+ x<-x1/(x1+x2)
+ print(x)
+ }
[1] 0.4307956
[1] 0.5797963
[1] 0.5250849
[1] 0.5596889
[1] 0.5006363
[1] 0.4768778
[1] 0.4780549
[1] 0.4744199
[1] 0.4659736
[1] 0.4750248
>
```

نلاحظ من البرنامج ان حلقة التكرار الداخلية لتوليد قيمة واحدة من x في حين الخارجية لتوليد 10 قيم من x

ملاحظة: يمكن اجراء البرنامج السابق بدون الحاجة الى حلقات تكرار من خلال الدالة u=unif(n) حيث n يمثل عدد القيم المراد توليدها وكما في البرنامج التالي :

```
rm(list=objects())
n<-3
m<-4
x1<-0
x2<-0
for(i in 1:7){
u<-runif(10)
if(i<=n){
 x1<-x1-log(u)
}else{x2<-x2-log(u)
}
}
  x<-x1/(x1+x2)
  print(x)
```

وعند تنفيذ البرنامج تكون مخرجاته كالاتي:

```
Console ~/
> m<-4
> x1<-0
> x2<-0
> for(i in 1:7){
+ u<-runif(10)
+ if(i<=n){
+ x1<-x1-log(u)
+ }else{x2<-x2-log(u)
+ }
+ }
> x<-x1/(x1+x2)
> print(x)
 [1] 0.3278665 0.1863702 0.3709701 0.1754717 0.4135046 0.3524890 0.5747749 0.6444096
 [9] 0.3842349 0.3125072
> 
```

تم توليد 10 قيم من المتغير x بناءا على الدالة الموضحة في البرنامج بدون الحاجة الى حلقة تكرارها . ونلاحظ في كلا الحالتين القيم محصورة بين 0 و 1

6-5. ايعاز while

يستخدم هذا الايعاز لتكوين حلقات تكرار عندما يكون هناك شرط محدد لاستمرارها خاصتا عندما يكون عددها غير معلوم, ويستخدم كبديل عن عبارة go to في لغات البرمجة الاخرى والشكل العام للايعاز هو:

```
while(conditions)
 {
 Statements
}
```

While : يمثل الايعاز

Condition: شرط التكرار

Statements :- العبارات المراد تكرارها

عمل الايعاز

اذا كان الشرط صحيح تنفذ العبارات statements عدا ذلك يتوقف الايعاز وينتقل التنفيذ الى ما بعد الايعاز.

مثال: اكتب برنامج لطبع الاعداد من 1 الى 10 باستعمال حلقة التكرار while ؟

الحل:

```
i<-1
while(i<11)
{
print(i)
i=i+1
}
```

ناتج البرنامج هو:

```
Console ~/
> i<-1
> while(i<11)
+ {
+ print(i)
+ i=i+1
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

7-5 الايعاز break

يستعمل الايعاز break ضمن حلقة تكرار لإيقافها عند تحقق شرط محدد .

مثال: اكتب برنامج يطبع الأعداد من 1 إلى 10 ويتوقف عن الطباعة عند أول عدد زوجي من مضاعفات العدد 3 :

الحل:

```
for (i in 1:10){
if (i%%2==0 & i%%3==0)break
print(i)
}
```

تنفيذ البرنامج :

```
Console ~/
> for (i in 1:10){
+ if (i%%2==0 & i%%3==0)break
+ print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

نلاحظ ان التكرار توقف عند تحقق الشرط وهو اول عدد زوجي من مضاعفات العدد 3 .

8-5 الايعاز next

يستعمل الايعاز next لتجاهل تنفيذ العبارات التي تليها في الحلقة بدون الخروج من حلقة التكرار .

مثال: اكتب برنامج يقوم بطباعة الأعداد الواقعة بين 1 و 20 والتي تقبل القسمة على 3 فقط.

الحل:

```
Source on Save
for (i in 1:20){
if (i%%3!=0)next
print(i)
}
```

نتيجة تنفيذ البرنامج كالاتي:

```
Console ~/
> for (i in 1:20){
+ if (i%%3!=0)next
+ print(i)
+ }
[1] 3
[1] 6
[1] 9
[1] 12
[1] 15
[1] 18
```

9-5 الايعاز repeat

يعتبر هذا الايعاز حلقة تكرار ويجب استعمال الايعاز break لايقافها أي لا يمكن ايقافها الا بشرط محدد.والشكل العام للايعاز هو:

```
repeat
 {
Statements
}
```

مثال: اكتب برنامج يطبع الاعداد من 0 الى 10 باستعمال الايعاز repeat .

الحل:

```r
i<-0
repeat
{
print(i)
if (i==10)break
i<-i+1
}
```

نتيجة تنفيذ البرنامج هي:

```
> i<-0
> repeat
+ {
+ print(i)
+ if (i==10)break
+ i<-i+1
+ }
[1] 0
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

الفصل السادس
المصفوفات والمتجهات

1-6 المقدمة

لقد كانت جميع الحسابات التي أجريناها حتى الآن مؤلفة من أعداد وحيدة البعد سنسميها أعداد مفردة. وتعتبر العمليات المجراة على الأعداد المفردة هي أساسيات علم الرياضيات. وبنفس الوقت، وعندما يريد الشخص إجراء نفس العملية على عدد مفرد أو أكثر، فسيحتاج إلى أكثر إعادة إجراء العملية عدة مرات، مما يعني هدر في الوقت والجهد. ولحل هذه المشكلة، عمد برنامج R Programming إلى إجراء العمليات الرياضية على مصفوفة من البيانات.

2-6 المتجهات The vectors

المتجهات هي عبارة عن عدة كائنات لها نفس النوع ومخزنة بترتيب محدد.ويمكن انشاء المتجهات باستعمال الايعازات الاتية :

1-2-6 الايعاز c

يستعمل هذا الايعاز لأنشاء متجه يتكون من قيم عددية نستخدم الايعاز c( ) حيث يرمز الحرف c إلى الكلمة concatenate والتي تعني "تسلسل" والشكل العام للإيعاز هو :

```
c(num1,num2,….,numn)
```

اذ ان :

c: يمثل الايعاز

(num1,num2,….,numn) : تمثل الاعداد من 1 الى n

مثال: كون متجه من الاعداد 0,6,7,8,9,7 باستعمال الايعاز c

الحل:

```
c(0,6,7,8,9,7)
[1] 0 6 7 8 9 7
```

ملاحظة : نستطيع تكوين متجه يتكون من سلسلة من القيم العددية باستخدام colon (:)

مثال: كون متجه يتالف من الاعداد من 5 الى 25 باستعمال colon

الحل:

```
numbers5to25<-5:25
numbers5to25
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

6-2-2 الايعاز seq

يستعمل هذا الايعاز لتكوين سلسلة من القيم العددية بشكل متوالية او متسلسلة تكون المسافة بين هذه القيم متساوية و الشكل العام للايعاز هو:

```
seq(initial value , end value , increment)
```

اذ ان:

initial value : هي القيمة الاولية اي بداية السلسلة

end value : القيمة الاخيرة اي نهاية السلسلة

Increment:- مقدار الزيادة

مثال: كون سلسلة القيم من 0 الى 1 بزيادة مقدارها 0.1 باستخدام ايعاز seq

الحل:

```
> seq(0,1,0.1)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

هنا استخدمنا الايعاز بمقدار للزيادة 0.1

مثال: كون سلسلة من القيم محصورة بين (-2,2) بزيادة مقدارها 0.5

الحل:

```
Console ~/
> seq(-2,2,0.5)
[1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0
```

3-2-6 ايعاز rep

يستخدم هذا الايعاز لتكون سلسلة من القيم العددية لمتغير او رقم معين او حرف

والشكل العام لايعاز هو:

```
rep(x , n)
```

اذ ان :

rep : يمثل الايعاز

x: يمثل العدد او الحرف او المتغير المطلوب تكراره

n: عدد مرات التكرار

اي تكوين سلسلة من رقم معين طولها هذه السلسلة بمقدار n

مثال : كون سلسلة مكونة من الرقم 4 طولها 8

الحل:

rep(4,8)

[1] 4 4 4 4 4 4 4 4

4-2-6 الايعاز length

يستعمل هذا الايعاز لمعرفة عدد القيم التي يتكون منها المتجه والشكل العام للايعاز هو

```
length(x)
```

اذ ان :

Length: يمثل الايعاز

x: تمثل المتجه المطلوب معرفة عدد قيمه.

مثال: جد عدد القيم للمتجه numbers5to25 في المثال السابق؟

الحل:

```
Console ~/
> numbers5to25<-5:25
> length(numbers5to25)
[1] 21
```

نلاحظ ان عدد القيم التي يتكون منها المتجه هي 21 قيم .

3-6 الفلترة Filtering

نقصد بالفلترة هي عملية الوصول لبيانات المتجه والتعامل بها وفق شرطاً أو عدة شروط .

مثال: كون متجه من الاعداد (1,3,4,6,2,5,8,10) باسم x ثم جد الاتي:

1. جد العنصر الثالث بالمتجه x
2. اوجد عدد القيم التي يتكون منها المتجه x
3. اضف الاعداد 12,14,20 ,10 الى المتجه x
4. اطبع القيمة الثانية والرابعة من المتجه x
5. استبدل القيمة الرابعة بالعدد 25
6. احذف العنصر الرابع

الحل:

كما في الشكل ادناه

```
Console ~/
> x<-c(1,3,4,6,2,5,8,10)
> #1.
> x[3]
[1] 4
> #2.
> length(x)
[1] 8
> #3.
> c(x,10,12,14,20)
 [1]  1  3  4  6  2  5  8 10 10 12 14 20
> #4.
> x[c(2,4)]
[1] 3 6
> #5.
> x[4]<-c(25)
> x
[1]  1  3  4 25  2  5  8 10
```

المطلب رقم 6.

```
Console ~/
> x<-c(1,3,4,6,2,5,8,10)
> x[-4]
[1]  1  3  4  2  5  8 10
```

1-3-6 الايعاز subset

يستعمل هذا الايعاز للوصول إلى بيانات المتجهات وفق تحقق شرطاً محدد والشكل العام للإيعاز هو :

```
subset(vector, condition)
```

اذ ان :

Subset: يمثل الايعاز

Vector: متجه من القيم

Condition: الشرط المطلوب تحقيقه

مثال: للمثال السابق جد القيم التي اكبر من 3 والزوجية منها ؟

الحل:

```
Console ~/
> x<-c(1,3,4,6,2,5,8,10)
> subset(x,x>3 & x%%2==0)
[1]  4  6  8 10
```

4-6 المصفوفات في R

يمكن انشاء مصفوفات باستعمال الايعازات الاتية :

1-4-6 الايعاز matrix

يستعمل هذا الايعاز لتكوين مصفوفة بأبعاد معينة عند توفر البيانات الخاصة بالمصفوفة بشكل متجهات اما صفية او عمودية من القيم والشكل العام للايعاز هو:

```
matrix(data, nrow, ncol)
```

اذ ان :

Matrix: يمثل الايعاز

data: البيانات بشكل متجه (vector)

nrow : يمثل عدد الصفوف المطلوبة

ncol : يمثل عدد الاعمدة المطلوبة

مثال: كون مصفوفة من العناصر $(\frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{2}, 1)$ باستعمال الايعاز matrix.

الحل:

```
> x <- matrix(c(1, 1/2, 1/3, 1/2, 1/3, 1/4, 1/3, 1/4, 1/5), nrow=3, ncol=3)
> x
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.5000000 0.3333333
[2,] 0.5000000 0.3333333 0.2500000
[3,] 0.3333333 0.2500000 0.2000000
>
```

نلاحظ ان عدد الصفوف هو 3 وعدد الاعمدة ايضا 3 والبيانات هي عبارة عن 9 قيم بشكل vector فتكون النتيجة هي مصفوفة 3*3 .

مثال: ولد مصفوفة تتكون من عامودين باستخدام المتجه c(1, 2, 3, 1, 4, 9).

الحل:

```
Console ~/
> X <- matrix(c(1, 2, 3, 1, 4, 9), ncol=2)
> X
     [,1] [,2]
[1,]    1    1
[2,]    2    4
[3,]    3    9
```

6-4-2 الايعاز cbind

يستعمل هذا الايعاز لتكوين مصفوفة مكونه من متجهات عمودية والشكل العام للايعاز هو:

cbind(d1, d2, ..., dm)

اذ ان:

cbind: يمثل الايعاز

d1, d2, ..., dm: متجهات عمودية

مثال : ولد مصفوفة من 3 صفوف و 3 اعمدة باستخدام ايعاز seq.

الحل:

```
Console ~/
> cbind(seq(1, 3), seq(2, 4), seq(3, 5))
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    3    4
[3,]    3    4    5
> 
```

وكذلك باستخدام ايعاز rep ايضا تتكون بشكل اعمدة.

```
> cbind(rep(4,4), rep(5,4), rep(6,4))
     [,1] [,2] [,3]
[1,]    4    5    6
[2,]    4    5    6
[3,]    4    5    6
[4,]    4    5    6
> 
```

مثال : ولد مصفوفة بالقيم (1,2,3) ومربعاتها تتكون من عامودين وثلاث صفوف باستعمال الايعاز cbind .

الحل:

```
Console ~/
> x <- seq(1,3)
> x2<- x^2
> y<-cbind(x,x2)
> y
     x x2
[1,] 1  1
[2,] 2  4
[3,] 3  9
```

6-4-3 الايعاز rbind

يستعمل هذا الايعاز لتكوين مصفوفة مكونه من متجهات صفية والشكل العام للايعاز هو:

```
rbind(d1, d2, ..., dn)
```

اذ ان:

rbind: يمثل الايعاز

d1, d2, ..., dm: متجهات صفية

مثال : ولد مصفوفة تتكون من 3 صفوف باستخدام ايعاز seq.

الحل:

```
Console ~/
> rbind(seq(1, 3), seq(2, 4), seq(3, 5))
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    3    4
[3,]    3    4    5
>
```

6-5 الايعاز apply

يعتبر هذا الايعاز من اهم الايعازات في لغة البرمجة R فيما يخص العمليات الحسابية على عناصر المصفوفات من صفوف واعمدة حيث يستعمل هذا الايعاز في اداء العمليات الحسابية على الصفوف او الاعمدة مثل حساب الوسط الحسابي للصفوف او الاعمدة او حساب المجموع للصفوف او الاعمدة او أي دالة اخرى يمكن تطبيقها على الصفوف او الاعمدة والشكل العام للإيعاز هو :

```
apply(A, row or column, function )
```

اذ ان :

apply: يمثل الايعاز

A: يمثل المصفوفة المطلوب اجراء العمليات الحسابية عليها

row or column: لاجراء العمليات على الصفوف نعطي 1 وللأعمدة نعطي 2.

function: الدالة المطلوب تطبيقها على الصفوف او الاعمدة.

مثال: اذا كانت لدينا المصفوفة A هي كالاتي:

$$A = \begin{pmatrix} 3 & 6 & 0 \\ 4 & 5 & 4 \\ 7 & 10 & 2 \end{pmatrix}$$

جد ما يلي:

1. جد مجموع عناصر الصفوف
2. جد الوسط الحسابي للأعمدة
3. جد الفروق بين الصفوف

الحل:

```
Console ~/
> A<-matrix(c(3,4,7,6,5,10,0,4,2),3,3)
> A
     [,1] [,2] [,3]
[1,]    3    6    0
[2,]    4    5    4
[3,]    7   10    2
> #1.
> apply(A,1,sum)
[1]  9 13 19
> apply(A,2,mean)
[1] 4.666667 7.000000 2.000000
> #3.
> apply(A,1,diff)
     [,1] [,2] [,3]
[1,]    3    1    3
[2,]   -6   -1   -8
```

6-6 التعامل مع المصفوفات (الفلترة)

يمكن التعامل مع المصفوفات من خلال الوصول الى عناصرها واجراء التغيرات عليها من اضافة وحذف واستبدال وغيرها من التغيرات وسنتناول اهم الايعازات للتعامل مع المصفوفات وهي كما في الجدول الاتي:

الايعاز	الوصف
A[r ,]	استدعاء الصف رقم r من المصفوفة A
A[, c]	استدعاء العمود رقم c من المصفوفة A
rbind(A,X)	ضافة المتجه x كصف جديد الى المصفوفة A
cbind(A,X)	ضافة المتجه x كعمود جديد الى المصفوفة A
A[,-c]	حذف العمود c من المصفوفة A
A[-r ,]	حذف الصف r من المصفوفة A
A[-r ,-c]	حذف الصف r والعمود c
diag(A)	لاستدعاء عناصر القطر الرئيسي

مثال: للمثال السابق للمصفوفة A جد ما يلي:

1. استدعي عناصر الصف الثاني
2. استدعي عناصر العمود الثالث
3. اضف المتجه x=(15,20,25) كصف جديد
4. اضف المتجه y=(20,22,24) كعمود جديد
5. استدعي عناصر القطر الرئيسي
6. احذف الصف الثاني
7. احذف العمود الثالث
8. احذف الصف الثاني العمود الثالث

الحل:

```
Console ~/
> A<-matrix(c(3,4,7,6,5,10,0,4,2),3,3)
> #1.
> A[2,]
[1] 4 5 4
> #2.
> A[,3]
[1] 0 4 2
> #3.
> X<-c(15,20,25)
> rbind(A,X)
  [,1] [,2] [,3]
     3    6    0
     4    5    4
     7   10    2
X   15   20   25
```

```
> cbind(A,c(20,22,24))
     [,1] [,2] [,3] [,4]
[1,]    3    6    0   20
[2,]    4    5    4   22
[3,]    7   10    2   24
> #5.
> diag(A)
[1] 3 5 2
```

تكملة المطاليب 6,7,8

```
Console ~/
> A<-matrix(c(3,4,7,6,5,10,0,4,2),3,3)
> #6.
> A[-2,]
     [,1] [,2] [,3]
[1,]    3    6    0
[2,]    7   10    2
> #7.
> A[,-3]
     [,1] [,2]
[1,]    3    6
[2,]    4    5
[3,]    7   10
> #8.
> A[-2,-3]
     [,1] [,2]
[1,]    3    6
[2,]    7   10
```

7-6 بعض الايعازات الخاصة بالمصفوفات

1-7-6 الايعاز dim

يستعمل هذا الايعاز لإيجاد ابعاد المصفوفة (عدد الصفوف وعدد الاعمدة) الشكل العام للإيعاز هو :

dim(A)[1 or 2]

Dim: يمثل الايعاز

A: المصفوفة المطلوب ايجاد ابعادها

[1 or 2]: يمثل البعد الذي نرغب بإيجاده فاذا نرغب بمعرفة عدد الصفوف يكون [1] واذا نرغب بإيجاده عدد الاعمدة يكون البعد [2]

مثال: اذا كانت لدينا المصفوفة B جد ابعاد المصفوفة

$$B = \begin{pmatrix} 2 & 3 & 5 \\ 3 & 6 & 8 \\ 5 & 8 & 9 \end{pmatrix}$$

الحل:

```
Console ~/
> B<-matrix(c(2,3,5,3,6,8,5,8,9),3,3)
> B
     [,1] [,2] [,3]
[1,]    2    3    5
[2,]    3    6    8
[3,]    5    8    9
> m<-dim(B)[1]
> m
[1] 3
> n<-dim(B)[2]
> n
[1] 3
```

نلاحظ ان عدد الصفوف m=3 وعدد الاعمدة n=3 أي ان ابعاد المصفوفة هي 3*3

2-7-6 الايعاز qr

يستعمل هذا الايعاز لمعرفة رتبة المصفوفة والشكل العام للايعاز هو

qr(A)$rank

qr: يمثل الايعاز

$: يمثل استدعاء الايعاز الثاني وهو rank المضمن في الايعاز qr

مثال: للمثال السابق اوجد رتبة المصفوفة B

الحل:

```
> qr(B)$rank
[1] 3
```

8-6 المصفوفات القياسية

يوجد في لغة البرمجة R العديد من المصفوفات الخاصة ومنها .

1-8-6 المصفوفة الصفرية

لأنشاء مصفوفة صفرية بأبعاد معينة نستعمل الايعاز matrix وكما يأتي:

matrix(0, nrow=n, ncol=m)

nrow: عدد الصفوف

ncol: عدد الاعمدة

مثال: كون مصفوفة صفرية من 4 صفوف و 4 اعمدة

الحل:

```
Console ~/
> matrix(0,nrow = 4,ncol = 4)
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
[4,]    0    0    0    0
```

2-8-6 انشاء مصفوفة الواحد

لإنشاء مصفوفة تتكون جميع عناصرها من العدد (1) من درجة معينة نستخدم الايعاز matrix وكما يأتي:

matrix(1, nrow=n, ncol=m)

مثال: كون مصفوفة الواحد من 4 صفوف و 4 اعمدة.

الحل:

```
Console ~/
> matrix(1,nrow = 4,ncol = 4)
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    1    1    1    1
[3,]    1    1    1    1
[4,]    1    1    1    1
```

3-8-6 مصفوفة الوحدة

لأنشاء مصفوفة الوحدة identity نستعمل الايعاز diag و كما يأتي:

daig(1,n,m)

مثال: ولد مصفوفة الوحدة مكونة من 4 صفوف و 4 اعمدة .

الحل:

```
Console ~/
> diag(1,4,4)
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
```

4-8-6 الايعاز t

يستعمل هذا الايعاز لإيجاد مبدلة المصفوفة و الشكل العام للإيعاز هو :

t(A)

t: يمثل الايعاز ، A: يمثل المصفوفة المطلوب ايجاد المبدلة لها.

مثال: المصفوفة C الاتية جد المبدلة لها .

$$C = \begin{pmatrix} 4 & 9 & 0 \\ 6 & 3 & 5 \\ 8 & 2 & 4 \end{pmatrix}$$

الحل:

```
Console ~/
> C<-matrix(c(4,6,8,9,3,2,0,5,4),3,3)
> C
     [,1] [,2] [,3]
[1,]    4    9    0
[2,]    6    3    5
[3,]    8    2    4
> t(C)
     [,1] [,2] [,3]
[1,]    4    6    8
[2,]    9    3    2
[3,]    0    5    4
```

9-6 توليد المصفوفات

بالإضافة الى الدوال اعلاه هناك مصفوفات لأيمكن توليدها بتلك الدوال فيتم تكوينها برمجياً باستعمال عبارات التكرار for loop.

مثال : اكتب برنامج باستخدام لغة R لتوليد المصفوفة الاتية :

$$a=\begin{bmatrix} 1 & 2 & 2 & 2 \\ 1 & 2 & 4 & 4 \\ 2 & 2 & 3 & 6 \\ 3 & 3 & 3 & 4 \end{bmatrix}$$

الحل:

البرنامج:

```r
a <- matrix(NA_real_, nrow = 4, ncol = 4)
for(i in 1:4){
  for(j in 1:4){
    if(i==j){
      a[i,j]=i
    }else if(i<j){
      a[i,j]=2*i
    }else{
      a[i,j]=i-1
    }
    }
  }
a
```

فيكون الناتج كما يأتي :

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    2    2
[2,]    1    2    4    4
[3,]    2    2    3    6
[4,]    3    3    3    4
```

مثال: اكتب برنامج باستخدام لغة R لتوليد المصفوفة الاتية :

$$x=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 4 & 4 \\ 3 & 3 & 3 & 9 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

الحل:

البرنامج:-

```
x <- matrix(NA_real_, nrow = 4, ncol = 4)
for(i in 1:4){
  for(j in 1:4){
    if(i==j){
      x[i,j]=i
    }else if(i<j){
      x[i,j]=i^2
    }else{
      x[i,j]=i
    }
  }
}
x
```

وعند تنفيذ البرنامج تكون مخرجاته كالاتي:

```
Console ~/
+ x[i,j]=i
+ }else if(i<j){
+ x[i,j]=i^2
+ }else{
+ x[i,j]=i
+ }
+ }
+ }
> x
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    2    2    4    4
[3,]    3    3    3    9
[4,]    4    4    4    4
>
```

مثال: لنفس المثال السابق اكتب برنامج الاستبدال عناصر القطر الرئيسي بعناصر القطر الثانوي ؟

الحل:

البرنامج:

```
#x <- matrix(NA_real_, nrow = 4, ncol = 4)
for(i in 1:4){
  for(j in 1:4){
    if(i==j){
      x[i,j]=i
    }else if(i<j){
      x[i,j]=i^2
    }else{
      x[i,j]=i
    }
    t=x[i,i]
    x[i,i]=x[i,j]
    x[i,j]=t
    j==j-1
  }
}
x
```

وعند تنفيذ البرنامج يكون الناتج كالاتي:

```
Console ~/
+ x[i,j]=i
+ }
+ t=x[i,i]
+ x[i,i]=x[i,j]
+ x[i,j]=t
+ j==j-1
+ }
+ }
> x
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    4    4    2    4
[3,]    9    3    9    3
[4,]    4    4    4    4
```

نلاحظ تم ابدال عناصر القطر الرئيسي مكان القطر الثانوي.

مثال : اجمع عناصر القطر الرئيسي؟

مثال: اجمع عناصر المثلث العلوي وعناصر المثلث السفلي على التوالي؟

مثال: اكتب برنامج لتوليد المصفوفة A ثم اجمع عناصر القطر الرئيسي وعناصر المثلثين ؟

$$A=\begin{bmatrix}1 & 3 & 3 & 3\\ 5 & 2 & 3 & 3\\ 5 & 5 & 3 & 3\\ 5 & 5 & 5 & 4\end{bmatrix}$$

الحل:

ويكون البرنامج بالشكل الاتي:

Source on Save

```
A <- matrix(NA_real_, nrow = 4, ncol = 4)
s1=0
s2=0
s3=0
for(i in 1:4){
  for(j in 1:4){
    if(i==j){
      A[i,j]=i
      s1=s1+A[i,j]
    }else if(i<j){
      A[i,j]=3
      s2=s2+A[i,j]
    }else{
      A[i,j]=5
      s3=s3+A[i,j]
    }
  }
}
A
s1
s2
s3
```

وعند تنفيذ البرنامج يكون الناتج كالاتي:

```
> rm(list=ls())
> A <- matrix(NA_real_, nrow = 4, ncol = 4)
> s1=0
> s2=0
> s3=0
> for(i in 1:4){
+ for(j in 1:4){
+ if(i==j){
+ A[i,j]=i
+ s1=s1+A[i,j]
+ }else if(i<j){
+ A[i,j]=3
+ s2=s2+A[i,j]
+ }else{
+ A[i,j]=5
+ s3=s3+A[i,j]
+ }
+ }
+ }
> A
     [,1] [,2] [,3] [,4]
[1,]    1    3    3    3
[2,]    5    2    3    3
[3,]    5    5    3    3
[4,]    5    5    5    4
> s1
[1] 10
> s2
[1] 18
> s3
[1] 30
```

نلاحظ من البرنامج اعلاه ان مجموع عناصر القطر الرئيسي يتمثل ب s1 ومجموع عناصر المثلث العلوي s2 و مجموع عناصر المثلث العلوي s3

سؤال: بدل عناصر القطر الرئيسي بالثانوي للمصفوفة A؟

الفصل السابع

العمليات الحسابية الخاصة بالمصفوفات والمتجهات

7-1المقدمة

يتضمن هذا الفصل العمليات الحسابية على المصفوفات والمتجهات وكذلك اهم الايعازات الخاصة بهذه العمليات اذ يتيح لنا برنامج R القيام بالعمليات الحسابية بسهولة لما يتميز به من مميزات .

7-2 العمليات الحسابية الاساسية

تتضمن هذه الفقرة عملية الضرب والجمع والطرح والقسمة والاسس وغيرها من العمليات الاساسية الحسابية.

7-2-1 عملية الجمع والطرح

لأجراء عملية الطرح والجمع على المصفوفات يجب تحقق الشرط الخاص بالجمع والطرح وهو ان تكون المصفوفتين المطلوب جمعهما من نفس الدرجة اما اذا لم يتحقق هذا الشرط فلا نستطيع اجراء هذه العملية .

مثال: اذا كانت لدينا المصفوفتين A,B الاتيتين :

$$A = \begin{pmatrix} 2 & 4 & 3 \\ 6 & 7 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 5 & -2 & 0 \\ -2 & 4 & 9 \end{pmatrix}$$

جد كلاً مما يلي :

1. حاصل مجموع المصفوفتين ($A + B$)
2. حاصل طرح المصفوفتين ($A - B$)
3. جد 2A+3B

الحل:

```
Console ~/
> A<-matrix(c(2,6,4,7,3,0),2,3)
> A
     [,1] [,2] [,3]
[1,]    2    4    3
[2,]    6    7    0
> B<-matrix(c(5,-2,-2,4,0,9),2,3)
> B
     [,1] [,2] [,3]
[1,]    5   -2    0
[2,]   -2    4    9
> A+B
     [,1] [,2] [,3]
[1,]    7    2    3
[2,]    4   11    9
> A-B
     [,1] [,2] [,3]
[1,]   -3    6    3
[2,]    8    3   -9
> 
```

```
> 2*A+3*B
     [,1] [,2] [,3]
[1,]   19    2    6
[2,]    6   26   27
```

2-2-7 عملية الضرب والقسمة والرفع لاس

لأجراء عملية ضرب المصفوفات لابد من تحقق شرط الضرب وهو ان تكون عدد اعمدة المصفوفة الاولى يساوي عدد صفوف المصفوفة الثانية فاذا تحقق الشرط فنستطيع ايجاد حاصل الضرب اما اذا لم يتحقق الشرط فلا نستطيع ايجاد حاصل الضرب ويتم ضرب المصفوفات الجبري في برنامج R باستعمال الايعاز (%*%) اما عملية القسمة فتتم اذا كانت المصفوفات تحتوي نفس العدد من العناصر أي من نفس الدرجة وعملية الرفع لاس في الجير الخطي تعني حاصل الضرب بنفس المصفوفة مثلا $A^3 = A.A.A$ وهكذا .

مثال: اذا كانت لدينا المصفوفتين A,B الاتيتين :

$$A = \begin{pmatrix} 2 & 5 & 7 \\ 4 & 5 & 1 \\ 2 & 3 & 9 \end{pmatrix} \quad , B = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix}$$

جد ما يلي:

1. حاصل ضرب AB
2. حاصل قسمة A/B
3. جد $A^2 \ , B^3$

الحل:

```
Console ~/
> A
     [,1] [,2] [,3]
[1,]    2    5    7
[2,]    4    5    1
[3,]    2    3    9
> B<-matrix(c(1,2,3,1,2,3,1,2,3),3,3)
> B
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    2    2
[3,]    3    3    3
```

```
> #1.
> A%*%B
     [,1] [,2] [,3]
[1,]   33   33   33
[2,]   17   17   17
[3,]   35   35   35
```

```
Console ~/
> #2.
> A/B
          [,1] [,2] [,3]
[1,] 2.0000000  5.0  7.0
[2,] 2.0000000  2.5  0.5
[3,] 0.6666667  1.0  3.0
> #3.
> A^2
     [,1] [,2] [,3]
[1,]    4   25   49
[2,]   16   25    1
[3,]    4    9   81
> B^3
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    8    8    8
[3,]   27   27   27
```

3-7 الايعازات الخاصة بالعمليات الحسابية الاخرى

يتضمن برنامج R العديد من الايعازات المهمة الخاصة بالعمليات الحسابية الخاصة بالمصفوفات والمتجهات ومنها ما يلي:

1-3-7 ايعاز det

يستعمل هذا الايعاز لايجاد محدد المصفوفة والشكل العام للايعاز هو :

det(A)

اذ ان :

Det: يمثل الايعاز

A: تمثل المصفوفة المطلوب ايجاد المحدد لها.

مثال : اذا كانت لدينا المصفوفة A الاتية :

$$A = \begin{pmatrix} 2 & 5 & 7 \\ 4 & 5 & 1 \\ 2 & 3 & 9 \end{pmatrix}$$

جد محدد المصفوفة A.

الحل:

```
Console ~/
> A<-matrix(c(2,4,2,5,5,3,7,1,9),3,3)
> A
     [,1] [,2] [,3]
[1,]    2    5    7
[2,]    4    5    1
[3,]    2    3    9
> det(A)
[1] -72
```

2-3-7 الايعاز solve

يستعمل هذا الايعاز لإيجاد معكوس المصفوفة A أي ايجاد A^{-1} والشكل العام للإيعاز هو :

solve(A)

اذ ان :

Solve: يمثل الايعاز

A: تمثل المصفوفة المطلوب ايجاد المعكوس لها.

مثال: اذ كانت لدينا المصفوفة E الاتية :

$$E = \begin{pmatrix} 4 & 2 \\ -1 & 7 \end{pmatrix}$$

جد معكوس المصفوفة E.

الحل :

```
> E<-matrix(c(4,-1,2,7),2,2)
> E
     [,1] [,2]
[1,]    4    2
[2,]   -1    7
> solve(E)
           [,1]        [,2]
[1,] 0.23333333 -0.06666667
[2,] 0.03333333  0.13333333
```

3-3-7 الايعاز sum

يمكن ايجاد مجموع عناصر المصفوفة او متجه من خلال الايعاز sum والشكل العام للايعاز هو :

sum(v)

اذ ان :

sum: يمثل الايعاز

v: يمثل المصفوفة المطلوب ايجاد مجموع عناصرها

مثال : للمصفوفة A الاتية :

$$A = \begin{pmatrix} 2 & 5 & 7 \\ 4 & 5 & 1 \\ 2 & 3 & 9 \end{pmatrix}$$

جد حاصل مجموع عناصرها .

الحل:

```
Console ~/
> A<-matrix(c(2,4,2,5,5,3,7,1,9),3,3)
> A
     [,1] [,2] [,3]
[1,]    2    5    7
[2,]    4    5    1
[3,]    2    3    9
> sum(A)
[1] 38
```

4-3-7 الايعاز colSums

يستعمل هذا الايعاز لإيجاد مجموع عناصر اعمدة المصفوفة والشكل العام للإيعاز هو :

colSums(V)

اذ ان :

colSums : يمثل الايعاز

V: يمثل المصفوفة المطلوب ايجاد مجموع اعمدتها.

مثال : للمصفوفة A في المثال لسابق جد مجموع عناصر اعمدتها.

الحل:

```
Console ~/
> A<-matrix(c(2,4,2,5,5,3,7,1,9),3,3)
> A
     [,1] [,2] [,3]
[1,]    2    5    7
[2,]    4    5    1
[3,]    2    3    9
> sum(A)
[1] 38
> colSums(A)
[1]  8 13 17
```

5-3-7 الايعاز rowSums

يستعمل هذا الايعاز لايجاد مجموع عناصر صفوف المصفوفة و الشكل العام للإيعاز هو :

rowSums(V)

اذ ان :

rowSums : يمثل الايعاز

V: يمثل المصفوفة المطلوب ايجاد مجموع صفوفها .

مثال: للمثال السابق جد مجموع عناصر صفوف المصفوفة A

الحل:

```
> rowSums(A)
[1] 14 10 14
```

6-3-7 الايعاز prod

يستعمل هذا الايعاز لايجاد حاصل ضرب جميع عناصر المصفوفة او المتجه والشكل العام للايعاز هو :

prod(v)

اذ ان :

prod: يمثل الايعاز

v: يمثل المصفوفة المطلوب ايجاد ضرب جميع عناصرها .

مثال: اذا كانت لدينا المصفوفة B الاتية :

$$B = \begin{pmatrix} 2 & 4 & 6 \\ 2 & -1 & 7 \\ 5 & -2 & 3 \end{pmatrix}$$

جد حاصل ضرب جميع عناصر المصفوفة B .

الحل:

```
Console ~/
> B<-matrix(c(2,2,5,4,-1,-2,6,7,3),3,3)
> B
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    2   -1    7
[3,]    5   -2    3
> prod(B)
[1] 20160
```

ملاحظة 1: لإيجاد حاصل ضرب اعمدة المصفوفة B نستعمل الايعاز الاتي:

apply(B,2,prod)

مثال : للمصفوفة B جد حاصل ضرب عناصر اعمدتها .

الحل:

```
Console ~/
> B<-matrix(c(2,2,5,4,-1,-2,6,7,3),3,3)
> B
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    2   -1    7
[3,]    5   -2    3
> prod(B)
[1] 20160
> apply(B,2,prod)
[1]  20    8 126
```

ملاحظة 2 : لإيجاد حاصل ضرب صفوف المصفوفة B نستعمل الايعاز الاتي:

apply(B , 1 , prod)

مثال: للمصفوفة B في المثال السابق جد حاصل ضرب صفوفها .

الحل:

```
Console ~/
> B<-matrix(c(2,2,5,4,-1,-2,6,7,3),3,3)
> B
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    2   -1    7
[3,]    5   -2    3
> prod(B)
[1] 20160
> apply(B,2,prod)
[1]  20    8 126
> apply(B,1,prod)
[1]  48 -14 -30
```

7-3-7 الايعاز exp

يستعمل هذا الايعاز لإيجاد الدالة الاسية للمصفوفة أي ايجاد $e^A = \sum_{k=0}^{\infty} A^k/_{k!}$

وعند ايجاد مصفوفة الدالة الاسية فان الايعاز يأخذ المبدلة تلقائياً فيجب اعادة تبدلها باستعمال الايعاز t(A) والشكل العام للايعاز هو :

exp(A)

اذ ان :

Exp: يمثل الايعاز

A: المصفوفة او المتجه المطلوب ايجاد قيمة الدالة الاسية له .

مثال : للمصفوفة B الاتية :

$$B = \begin{pmatrix} 2 & 4 & 6 \\ 2 & -1 & 7 \\ 5 & -2 & 3 \end{pmatrix}$$

جد مصفوفة الدالة الاسية للمصفوفة B .

الحل:

```
Console ~/
> B<-matrix(c(2,2,5,4,-1,-2,6,7,3),3,3)
> B
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    2   -1    7
[3,]    5   -2    3
> exp(A)
          [,1]      [,2]        [,3]
[1,]  7.389056 148.41316 1096.633158
[2,] 54.598150 148.41316    2.718282
[3,]  7.389056  20.08554 8103.083928
> t(exp(A))
            [,1]       [,2]        [,3]
[1,]    7.389056  54.598150    7.389056
[2,]  148.413159 148.413159   20.085537
[3,] 1096.633158   2.718282 8103.083928
```

8-3-7 الايعاز cumsum

يستعمل هذا الإيعاز لإيجاد حاصل الجمع التراكمي لقيم عناصر المصفوفة او متجه والشكل العام للإيعاز هو :

cumsum(A)

اذ ان :

cumsum : يمثل الايعاز

A: يمثل المصفوفة او متجه المطلوب ايجاد حاصل الجمع التراكمي لها.

مثال: للمصفوفة C الاتية :

$$C = \begin{pmatrix} 2 & 4 & 6 \\ 6 & 2 & 4 \\ 9 & 9 & 3 \end{pmatrix}$$

جد حاصل الجمع التراكمي لقيم المصفوفة C :

الحل:

```
Console ~/
> C<-matrix(c(2,6,9,4,2,9,6,4,3),3,3)
> C
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    6    2    4
[3,]    9    9    3
> cumsum(C)
[1]  2  8 17 21 23 32 38 42 45
```

ملاحظة 1: لإيجاد حاصل الجمع التراكمي لاعمدة المصفوفة C نستعمل الايعاز الاتي:

apply(C,2,cumsum)

مثال: للمصفوفة C في المثال السابق جد حاصل الجمع التراكمي لاعمدة المصفوفة .

الحل:

```
Console ~/
> C<-matrix(c(2,6,9,4,2,9,6,4,3),3,3)
> C
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    6    2    4
[3,]    9    9    3
> cumsum(C)
[1]  2  8 17 21 23 32 38 42 45
> apply(C,2,cumsum)
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    8    6   10
[3,]   17   15   13
```

ملاحظة 2: لإيجاد حاصل الجمع التراكمي لصفوف المصفوفة C نستعمل الايعاز الاتي:

apply(C,1,cumsum)

مثال: للمصفوفة C في المثال السابق جد حاصل الجمع التراكمي لصفوف المصفوفة .

الحل:

```
Console ~/
> C<-matrix(c(2,6,9,4,2,9,6,4,3),3,3)
> C
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    6    2    4
[3,]    9    9    3
> apply(C,1,cumsum)
     [,1] [,2] [,3]
[1,]    2    6    9
[2,]    6    8   18
[3,]   12   12   21
```

9-3-7 الايعاز cumprod

يستعمل هذا الايعاز لايجاد حاصل الضرب التراكمي لجميع قيم عناصر المصفوفة او متجه والشكل العام للايعاز هو:

cumprod (A)

اذ ان :

cumprod: يمثل الايعاز

A: المصفوفة او المتجه المطلوب ايجاد حاصل الضرب التراكمي له.

مثال: للمصفوفة D الاتية :

$$D = \begin{pmatrix} 4 & -2 \\ 3 & 7 \end{pmatrix}$$

جد حاصل الضرب التراكمي لعناصر المصفوفة D.

الحل:

```
Console ~/
> D<-matrix(c(4,3,-2,7),2,2)
> D
     [,1] [,2]
[1,]    4   -2
[2,]    3    7
> cumprod(D)
[1]    4   12  -24 -168
```

والجدول الاتي يوضح كيفية ايجاد حاصل الضرب التراكمي للصفوف او الاعمدة

الايعاز	الوصف
apply(D,1,cumprod)	يستعمل الايجاد حاصل الضرب التراكمي لعناصر الصفوف فقط
apply(D,2,cumprod)	يستعمل الايجاد حاصل الضرب التراكمي لعناصر الاعمدة فقط

مثال: للمصفوفة D في المثال السابق جد حاصل الضرب التراكمي للصفوف مرة وللأعمدة مرة اخرى.

الحل:

```
Console ~/
> D<-matrix(c(4,3,-2,7),2,2)
> D
     [,1] [,2]
[1,]    4   -2
[2,]    3    7
> cumprod(D)
[1]   4   12  -24 -168
> apply(D,1,cumprod)
     [,1] [,2]
[1,]    4    3
[2,]   -8   21
> apply(D,2,cumprod)
     [,1] [,2]
[1,]    4   -2
[2,]   12  -14
```

10-3-7 الايعاز diff

يستعمل هذا الايعاز لايجاد الفرق بين عناصر المصفوفة او المتجه والشكل العام للايعاز هو :

diff(A)

diff: يمثل الايعاز

A: المصفوفة او المتجه المطلوب ايجاد الفروق لعناصره.

مثال: اذا كان لدينا المتجه E الاتي:

$$E = (1,2,3,4,5,6,7,8,9,10)$$

جد حاصل الفرق بين قيم عناصر التجه .

الحل:

```
Console ~/
> E<-c(1,2,3,4,5,6,7,8,9,10)
> E
 [1]  1  2  3  4  5  6  7  8  9 10
> diff(E)
[1] 1 1 1 1 1 1 1 1 1
```

4-7 الايعازات الخاصة بالعمليات الحسابية الاحصائية

1-4-7 الايعاز eigen

يستعمل هذا الايعاز لايجاد القيم الذاتية و المتجهات الذاتية للمصفوفات و الشكل العام للايعاز هو :

eigen(A)$values

eigen(A)$vectors

اذ ان :

eigen: يمثل الايعاز

A : يمثل المصفوفة المطلوب ايجاد لها القيم الذاتية او المتجهات الذاتية.

$: يمثل استدعاء القيم الذاتية او المتجهات

Values: تمثل القيم الذاتية

Vectors : يمثل المتجهات الذاتية .

مثال: للمصفوفة X الاتية :

$$X = \begin{pmatrix} 4 & 2 & 4 \\ 5 & 2 & 5 \\ 3 & 2 & 1 \end{pmatrix}$$

جد القيم الذاتية والمتجهات الذاتية للمصفوفة X

الحل:

Console ~/

```
> X<-matrix(c(4,5,3,2,2,2,4,5,1),3,3)
> X
     [,1] [,2] [,3]
[1,]    4    2    4
[2,]    5    2    5
[3,]    3    2    1
> eigen(X)
eigen() decomposition
$values
[1]  9.0400789 -1.7933482 -0.2467307

$vectors
           [,1]       [,2]       [,3]
[1,] -0.5924596 -0.3268051 -0.6099480
[2,] -0.7017607 -0.5664323  0.7427512
[3,] -0.3956305  0.7565401  0.2761956
```

القيم الذاتية

المتجهات الذاتية

2-4-7 الايعاز chol

يستعمل هذا الايعاز لايجاد مصفوفة كلاوس كي التي تكون مصفوفة مربعة ، متماثلة ،وموجبة ومصفوفة مثلث علوي لمصفوفة معينة ، والشكل العام للايعاز هو :

chol(A)

اذ ان :

Chol:يمثل الايعاز

A : المصفوفة المطلوب ايجاد كلاوس كي لها.

مثال: للمصفوفة B الاتية :

$$B = \begin{pmatrix} 5 & 1 \\ 1 & 3 \end{pmatrix}$$

جد مصفوفة كلاوس كي لها.

الحل:

```
Console ~/
> B<-matrix(c(5,1,1,3),2,2)
> B
     [,1] [,2]
[1,]    5    1
[2,]    1    3
> chol(B)
         [,1]      [,2]
[1,] 2.236068 0.4472136
[2,] 0.000000 1.6733201
```

3-4-7 الايعاز mean

يستعمل هذا الايعاز لإيجاد الوسط الحسابي لقيم عناصر المصفوفات او المتجهات

و الشكل العام للإيعاز هو :

mean(A)

mean: يمثل الايعاز

A: تمثل المصفوفة او متجه المطلوب حساب متوسط قيم عناصره.

مثال : للمصفوفة Z الاتية :

$$Z = \begin{pmatrix} 2 & 3 & 6 \\ 3 & 1 & 3 \\ 6 & 7 & 0 \end{pmatrix}$$

جد الوسط الحسابي لقيم عناصر المصفوفة .

الحل:

```
Console ~/
> Z<-matrix(c(2,3,6,3,1,7,6,3,0),3,3)
> Z
     [,1] [,2] [,3]
[1,]    2    3    6
[2,]    3    1    3
[3,]    6    7    0
> mean(Z)
[1] 3.444444
```

والجدول الاتي يوضح كيفية ايجاد الوسط الحسابي للصفوف او الاعمدة

الايعاز	الوصف
apply(D,1,mean)	يستعمل الايجاد الوسط الحسابي لعناصر الصفوف فقط
apply(D,2,mean)	يستعمل الايجاد الوسط الحسابي لعناصر الاعمدة فقط

مثال: للمصفوفة z في المثال السابق جد الوسط الحسابي لقيم عناصر الصفوف مرة وللأعمدة مرة اخرى.

الحل:

```
Console ~/
> Z<-matrix(c(2,3,6,3,1,7,6,3,0),3,3)
> Z
     [,1] [,2] [,3]
[1,]    2    3    6
[2,]    3    1    3
[3,]    6    7    0
> mean(Z)
[1] 3.444444
> apply(Z,1,mean)
[1] 3.666667 2.333333 4.333333
> apply(Z,2,mean)
[1] 3.666667 3.666667 3.000000
```

5-4-7 الايعاز sd

يستعمل هذا الايعاز لإيجاد الانحراف المعياري لقيم عناصر المصفوفات او المتجهات والشكل العام للإيعاز هو :

sd(A)

sd: يمثل الايعاز

A: تمثل المصفوفة او متجه المطلوب حساب الانحراف المعياري لقيم عناصره.

والجدول الاتي يوضح كيفية ايجاد الوسط الحسابي للصفوف او الاعمدة

الايعاز	الوصف
apply(D,1,sd)	يستعمل الايجاد الانحراف المعياري لعناصر الصفوف فقط
apply(D,2,sd)	يستعمل الايجاد الانحراف المعياري لعناصر الاعمدة فقط

مثال : للمصفوفة A الاتية :

$$A = \begin{pmatrix} -1 & 2 \\ 4 & 6 \end{pmatrix}$$

جد ما يلي:

1. الانحراف المعياري لجميع عناصر المصفوفة
2. الانحراف المعياري لصفوف المصفوفة
3. الانحراف المعياري لأعمدة المصفوفة

الحل:

```
Console ~/
> A<-matrix(c(-1,4,2,6),2,2)
> A
     [,1] [,2]
[1,]   -1    2
[2,]    4    6
> #1.
> sd(A)
[1] 2.986079
> #2.
> apply(A,1,sd)
[1] 2.121320 1.414214
> #3.
> apply(A,2,sd)
[1] 3.535534 2.828427
```

6-4-7 الايعاز **var**

يستعمل هذا الايعاز لإيجاد التباين لقيم عناصر المصفوفات او المتجهات و الشكل العام للإيعاز هو :

var(A)

var: يمثل الايعاز

A: تمثل المصفوفة او متجه المطلوب حساب التباين لقيم عناصره.

والجدول الاتي يوضح كيفية ايجاد الوسط الحسابي للصفوف او الاعمدة

الايعاز	الوصف
apply(D,1,var)	يستعمل الايجاد التباين لعناصر الصفوف فقط
apply(D,2,var)	يستعمل الايجاد التباين لعناصر الاعمدة فقط

مثال : للمصفوفة A الاتية :

$$A = \begin{pmatrix} 2 & 4 \\ 4 & -1 \end{pmatrix}$$

جد ما يلي:

1. التباين لجميع عناصر المصفوفة
2. التباين لصفوف المصفوفة
3. التباين لأعمدة المصفوفة

الحل:

```
Console ~/
> A<-matrix(c(2,4,4,-1),2,2)
> A
     [,1] [,2]
[1,]    2    4
[2,]    4   -1
> var(A)
     [,1] [,2]
[1,]    2 -5.0
[2,]   -5 12.5
> apply(A,1,var)
[1]  2.0 12.5
> apply(A,2,var)
[1]  2.0 12.5
```

7-4-7 الايعاز summary

يستعمل هذا الايعاز لايجاد اقل قيمة ،الربيع الاول ، الوسيط ، الوسط الحسابي، الربيع الثالث ، اكبر قيمة للمتجه او لكل عمود في المصفوفة باعتباره متجه والشكل العام للايعاز هو :

Summary(A)

Summary: الايعاز

A: المصفوفة المطلوب ملخص عنها .

مثال: للمصفوفة A في المثال السابق جد ملخص للمصفوفة .

الحل:

```
> summary(A)
       V1             V2
 Min.   :2.0   Min.   :-1.00
 1st Qu.:2.5   1st Qu.: 0.25
 Median :3.0   Median : 1.50
 Mean   :3.0   Mean   : 1.50
 3rd Qu.:3.5   3rd Qu.: 2.75
 Max.   :4.0   Max.   : 4.00
```

8-4-7 الايعاز cor

يستعمل هذا الايعاز لايجاد قيمة معامل ارتباط بيرسون بين متجهين من القيم العددية والشكل العام للايعاز هو :

cor(v,w)

اذ ان :

cor: الايعاز

v ,w : متجهين عددين

مثال: اذا كان لدينا المتجهين الآتيين v,w يمثلان درجات حرارة اسبوعين على التوالي وهي كالاتي:

v	25 ,30,27,35,44,41,34
w	33,23,44,56,23,33,44

جد معامل ارتباط بيرسون بين المتجهين .

الحل:

```
Console ~/
> v<-c(25 ,30,27,35,44,41,34)
> v
[1] 25 30 27 35 44 41 34
> w<-c(33,23,44,56,23,33,44)
> w
[1] 33 23 44 56 23 33 44
> cor(v,w)
[1] -0.2093695
```

نلاحظ ان قيمة معامل الارتباط 0.2093695- مما يدل على انه ارتباط عكسي ضعيف .

الفصل الثامن

بعض طرائق التحليل العددي

1-8 المقدمة

في البداية يجب علينا ان نفهم ما هو التحليل العددي ،سوف نوضح ذلك بمثال بسيط .لنفرض لدينا المعادلة الاتية والمطلوب هو الحصول على جذور x (roots of x) اي قيم x التي تحقق المعادلة

$$x^2 + 2x - 14.6025 = 0$$

ونفرض اننا لا نريد الحل بأستخدام الدستور فنلجأ الى الطرق العددية للوصول الى الحل السريع وبشكل تقريبي وان الفرق بين الحل التقريبي والحل الحقيقي يسمى بالخطأ.

2-8 ايجاد الجذور للمعادلات بالشكل التقريبي

لكي نبدا اي طريقة لايجاد حل تقريبي يجب ان نحدد بشكل تقريبي ان الجذر يقع في الفترة مثلا [1,2] او غيرها من الفترات ولكي نحدد موقع الجذر بشكل تقريبي يجب علينا ان نرسم منحنى الدالة ونحدد الموقع التقريبي. ويجب ان نعلم بان الدالة تتغير قيمتها قبل وبعد الجذر من الموجب الى السالب او بالعكس ومن هذه الخاصية ننطلق لتحديد الجذور .لذلك نقوم بخطوات معينه ونحدد قيمة الدالة في كل خطوة فمتى ما تغيرت اشارة الدالة دل ذلك على وجود جذر في تلك المنطقة.ان لغة R تسهل لنا هذه الخطوات وتعطينا جذورا للمعادلات عن طريق برمجة الخطوات والحصول على الجذور

مثال : اكتب برنامج لايجاد الجذور للمعادلة الاتية $f(x) = cox(2x)^3$ على الفترة [0,10] ثم حدد منطقة الجذور على الرسم ؟

الحل:

من خلال تحميل الحزمة rootsolve نستخدم الدالة uniroot.all لايجاد الجذور الممكنة وكما يأتي

البرنامج :-

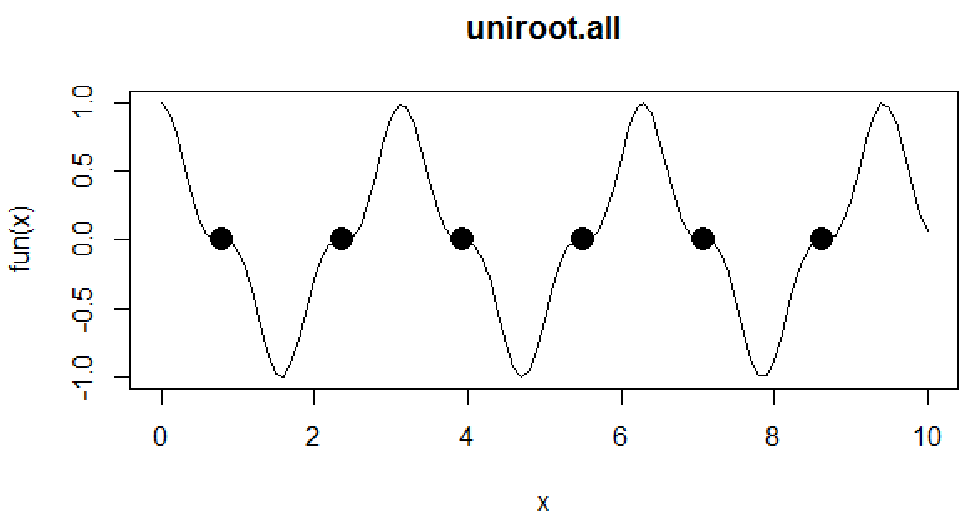
```
# roots of equation ####
fun <- function (x) cos(2*x)^3

curve(fun(x), 0, 10,main = "uniroot.all")

All <- rootSolve::uniroot.all(fun, c(0, 10))
points(All, y = rep(0, length(All)), pch = 16, cex = 2)
f(All)

```

اما بالنسبة للمنحى الدالة يكون بالشكل الاتي :



نلاحظ ان النقاط اعلاه هي منطقة جذور اما الجذور للمعادلة هي مساوية الى :

```
f(All)
[1] -21.703407    1.037199   -1.322566    1.018333
1.306515    1.029673
```

مثال: اكتب برنامج لإيجاد الجذور للمعادلة الاتية $f(x) = \frac{4}{25}x^2 - \frac{16}{25}x - \frac{9}{25}$ على الفترة [-5,5] ثم حدد منطقة الجذور على الرسم ؟

الحل:

البرنامج:-

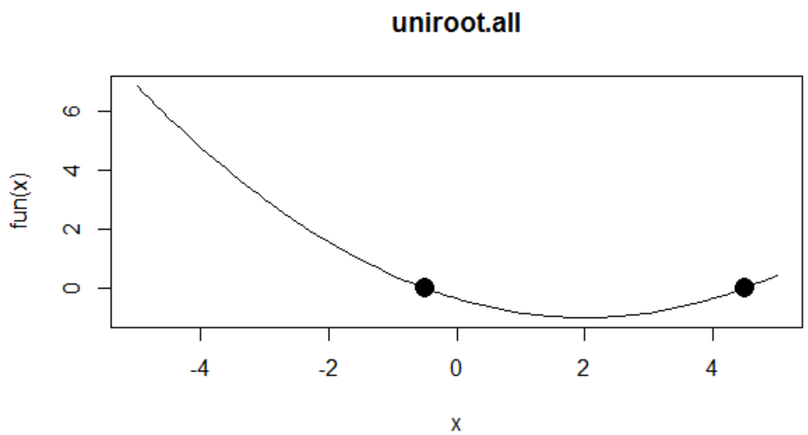
```
# roots of equation ####
fun <- function (x) 4/25*x^2-16/25*x-9/25

curve(fun(x), -5, 5,main = "uniroot.all")

All <- rootSolve::uniroot.all(fun, c(-5, 5))
points(All, y = rep(0, length(All)), pch = 16, cex = 2)
f(All)
```

```
> # roots of equation ####
> fun <- function (x) 4/25*x^2-16/25*x-9/25
> curve(fun(x), -5, 5,main = "uniroot.all")
> All <- rootSolve::uniroot.all(fun, c(-5, 5))
> points(All, y = rep(0, length(All)), pch = 16, cex = 2)
> f(All)
[1]  3.171358 -1.355576
```

من البرنامج نلاحظ وجود جذرين للمعادلة و هذا يعني وجود منطقتين ويبين ذلك من خلال الرسم .



اذا ان منطقة الجذور مبينة في الرسم في النقطة (1,0-) و النقطة (4.5,1)

مثال: اكتب برنامج لايجاد الجذور للمعادلة الاتية $f(x) = \sin(x) - \cos(x)$

على الفترة $[0,2\pi]$ ثم حدد منطقة الجذور على الرسم ؟

الحل:

البرنامج:-

diff simulation.R   Untitled1*

Source on Save

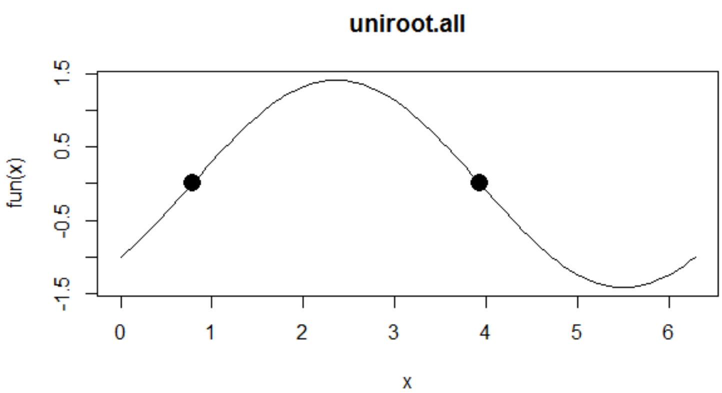
```
# roots of equation ####
fun <- function (x) sin(x)-cos(x)

curve(fun(x), 0, 2*pi,main = "uniroot.all")

All <- rootSolve::uniroot.all(fun, c(0, 2*pi))
points(All, y = rep(0, length(All)), pch = 16, cex = 2)
f(All)
```

```
> # roots of equation ####
> fun <- function (x) sin(x)-cos(x)
> curve(fun(x), 0, 2*pi,main = "uniroot.all")
> All <- rootSolve::uniroot.all(fun, c(0, 2*pi))
> points(All, y = rep(0, length(All)), pch = 16, cex = 2)
> f(All)
[1] -21.721342  -1.322382
```

نلاحظ وجود جذرين للمعادلة هما [1.322382- 21.721342-] ونبين ذلك من خلال الرسم .



8-2-1 طريقة نيوتن – رافسون

The Newton – Raphson method

إذا استطعنا عزل جذر واحد للمعادلة $f(x) = 0$ في الفترة $[a,b]$ وَ بفرض أن الدالة $y = f(x)$ دالة متصلة ضمن هذه الفترة, ثم إذا رسمنا المنحنى البياني للدالة $y=f(x)$ عندئذ إذا أخذنا نقطة ما x_0 في الفترة $[a,b]$ و أوجدنا $f(x_0)$ ورسمنا المماس للدالة $y = f(x)$ في النقطة $(x_0, f(x_0))$

حيث أن معادلة المستقيم المماس المار من النقطة $(x_0, f(x_0))$ و ميله $m = f'(x_0)$ تعطى بالعلاقة التالية :

$$y - y_0 = f(x) - f(x_0) = m(x - x_0) = f'(x_0)(x - x_0)$$

وَ بفرض أن هذا المماس يقطع محور السينات في نقطة و لتكن x_1

بما أن $(x_1,0)$ نقطة تقاطع هذا المماس مع محور السينات فإذا بدّلنا كل x بــ x_1 وكل y بــ 0 نجد :

$$0 - f(x_0) = f'(x_0)(x_1 - x_0)$$

و منه

$$x_1 - x_0 = \frac{-f(x_0)}{f'(x_0)}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

ثم نوجد $f(x_1)$ ونرسم المماس للدالة $y = f(x)$ في النقطة $(x_1, f(x_1))$ و ميله $m = f'(x_1)$

فيقطع المماس الجديد محور السينات في نقطة ولتكن x_2 بكتابة معادلة هذا المماس المار من النقطة $(x_1, f(x_1))$ و ميله $m = f'(x_1)$ نجد :

$$f(x) - f(x_1) = f'(x_1)(x - x_1)$$

بما أن $(x_2, 0)$ نقطة تقاطع هذا المماس مع محور السينات فإذا بدّلنا كل x بـ x_2 و $y = f(x)$ بـ 0 نجد :

$$0 - f(x_1) = f'(x_1)(x_2 - x_1)$$

و منه .

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

وبتكرار هذه العملية عدداً من المرّات نحصل على الصيغة العامة التالية :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0,1,2,....$$

و هذه الصيغة تسمى بصيغة نيوتن – رافسون - The Newton – Raphson method أو اختصاراً تسمى طريقة نيوتن (Newton) .!

وَ يمكن اعتبار النقطة x_{n+1} جذر تقريبي للمعادلة $f(x) = 0$ إذا تحقق الشرط :

$$\left| x_{n+1} - x_n \right| < \varepsilon$$

حيث ε عدد صغير جداً .

ملاحظة :

في بعض الحالات فإن رسم المماس عند نقطة ما من منحنى الدالة يؤدي الى ان نقطة تقاطع هذا المماس مع محور السينات تقع خارج الفترة [a,b] في هذه الحالات فإن طريقة نيوتن تكون غير عملية (المعنى الهندسي ان التفاضل الاول لا يمكن ايجاده).

ملاحظة:

ان الحل النظري بطريقة نيوتن رافسون يأخذ وقت وجهد لذلك توفر لك لغة R الوقت والجهد لتبرمج هذه الطريقة بخطوات بسيطة لايجاد الحل .

مثال: اكتب برنامج لايجاد حل المعادلة $f(x) = e^{-x} + x^4$ بطريقة نيوتن – رافسون بعدد دورات 7 اذا علمت ان $x_0 = 0.5$.مع رسم منحنى الدالة.

الحل:

البرنامج:-

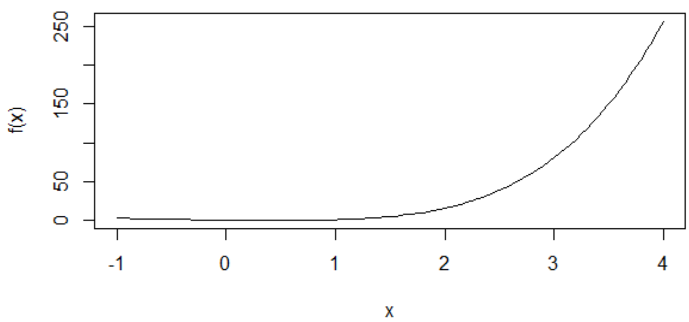
```
Untitled1*
Source on Save
f <- function(x) exp(-x) + x^4
fprime <- function(x) -exp(-x) + 4 * x^3
fprimeprime <- function(x) exp(-x) + 12 * x^2
x <- c(0.5, rep(NA, 6))
fval <- rep(NA, 7)
fprimeval <- rep(NA, 7)
fprimeprimeval <- rep(NA, 7)
for (i in 1:6) {
  fval[i] <- f(x[i])
  fprimeval[i] <- fprime(x[i])
  fprimeprimeval[i] <- fprimeprime(x[i])
  x[i + 1] <- x[i] - fprimeval[i] / fprimeprimeval[i]
  }
data.frame(x, fval, fprimeval, fprimeprimeval)
curve(f,-1,4)
```

يكون الناتج كما يأتي:-

```
> f <- function(x) exp(-x) + x^4
> fprime <- function(x) -exp(-x) + 4 * x^3
> fprimeprime <- function(x) exp(-x) + 12 * x^2
> x <- c(0.5, rep(NA, 6))
> fval <- rep(NA, 7)
> fprimeval <- rep(NA, 7)
> fprimeprimeval <- rep(NA, 7)
> for (i in 1:6) {
+ fval[i] <- f(x[i])
+ fprimeval[i] <- fprime(x[i])
+ fprimeprimeval[i] <- fprimeprime(x[i])
+ x[i + 1] <- x[i] - fprimeval[i] / fprimeprimeval[i]
+ }
> data.frame(x, fval, fprimeval, fprimeprimeval)
          x      fval     fprimeval fprimeprimeval
1 0.5000000 0.6690307 -1.065307e-01       3.606531
2 0.5295383 0.6675070  5.076129e-03       3.953806
3 0.5282544 0.6675038  9.980020e-06       3.938266
4 0.5282519 0.6675038  3.881429e-11       3.938235
5 0.5282519 0.6675038  0.000000e+00       3.938235
6 0.5282519 0.6675038  0.000000e+00       3.938235
7 0.5282519        NA            NA             NA
> curve(f,-1,4)
> 
```

نلاحظ من البرنامج تم عدد دورات x هي 7 وقد ثبتت قيمة x من الدورة الرابعة لذلك فان الحل لهذه المعادلة يكون مساويا الى $x7=0.5282519$

رسم الدالة :



لان سوف نوضح المثال المحلول في الجانب النظري اعلاه

$f(x)=x^3-2x^2+x-3=0$ بطريقة نيوتن رافسون . بعدد دورات 25 دورة .علما ان $x_0 = 2.2$

البرنامج:-

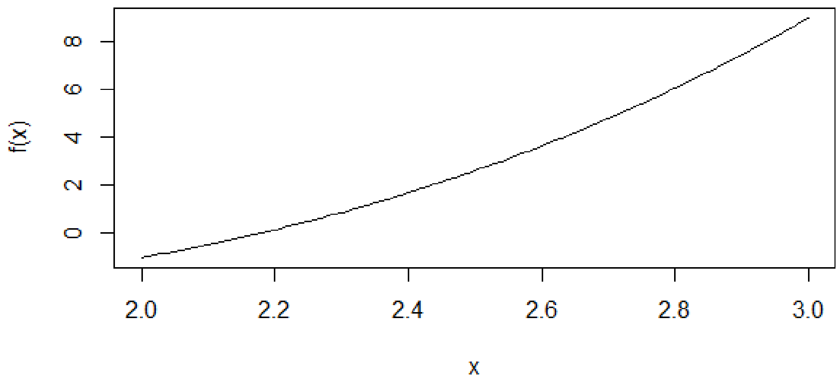
```
Untitled1*   Untitled2*
Source on Save
f <- function(x) x^3-2*x^2+x-3
fprime <- function(x) 3*x^3-4*x+1
fprimeprime <- function(x) 6*x-4
x <- c(2.2, rep(NA, 25))
fval <- rep(NA, 26)
fprimeval <- rep(NA, 26)
fprimeprimeval <- rep(NA, 26)
for (i in 1:25) {
  fval[i] <- f(x[i])
  fprimeval[i] <- fprime(x[i])
  fprimeprimeval[i] <- fprimeprime(x[i])
  x[i + 1] <- x[i] - fprimeval[i] / fprimeprimeval[i]
}
data.frame(x, fval, fprimeval, fprimeprimeval)
curve(f,2,3)
```

وعند تنفيذ البرنامج يكون الناتج كالاتي:

```
Console ~/
> fval <- rep(NA, 26)
> fprimeval <- rep(NA, 26)
> fprimeprimeval <- rep(NA, 26)
> for (i in 1:25) {
+ fval[i] <- f(x[i])
+ fprimeval[i] <- fprime(x[i])
+ fprimeprimeval[i] <- fprimeprime(x[i])
+ x[i + 1] <- x[i] - fprimeval[i] / fprimeprimeval[i]
+ }
> data.frame(x, fval, fprimeval, fprimeprimeval)
            x      fval     fprimeval fprimeprimeval
1   2.20000000  0.168000  2.414400e+01       9.200000
2  -0.42434783 -3.860903  2.468153e+00      -6.546087
3  -0.04730533 -3.051887  1.188904e+00      -4.283832
4   0.23022744 -2.863579  1.156996e-01      -2.618635
5   0.27441062 -2.855528 -3.565214e-02      -2.353536
6   0.25926229 -2.857745  1.523128e-02      -2.444426
7   0.26549332 -2.856766 -5.832026e-03      -2.407040
8   0.26307041 -2.857136  2.336530e-03      -2.421578
9   0.26403529 -2.856987 -9.198011e-04      -2.415788
10  0.26365455 -2.857045  3.646343e-04      -2.418073
11  0.26380534 -2.857022 -1.441523e-04      -2.417168
12  0.26374571 -2.857031  5.705065e-05      -2.417526
13  0.26376931 -2.857028 -2.256897e-05      -2.417384
14  0.26375997 -2.857029  8.929710e-06      -2.417440
15  0.26376366 -2.857029 -3.532918e-06      -2.417418
16  0.26376220 -2.857029  1.397788e-06      -2.417427
17  0.26376278 -2.857029 -5.530247e-07      -2.417423
18  0.26376255 -2.857029  2.188011e-07      -2.417425
19  0.26376264 -2.857029 -8.656729e-08      -2.417424
20  0.26376261 -2.857029  3.424983e-08      -2.417424
21  0.26376262 -2.857029 -1.355074e-08      -2.417424
22  0.26376261 -2.857029  5.361267e-09      -2.417424
23  0.26376262 -2.857029 -2.121152e-09      -2.417424
24  0.26376262 -2.857029  8.392209e-10      -2.417424
25  0.26376262 -2.857029 -3.320326e-10      -2.417424
26  0.26376262        NA            NA             NA
> curve(f,2,3)
>
```

من البرنامج السابق نلاحظ ان قيمة x ثبتت عند الدورة 23 لذلك فان الحل النظري لمثل هذه المسألة ياخذ وقت وجهد طويل وهنا تكمن اهمية البرمجة في توفير الوقت والجهد . اذا ان قيمة حل المعادلة السابقة هي x23=0.26376262

رسم منحنى الدالة :



3-8 التكامل العددي

هناك عدة طرق عددية لإيجاد القيمة التقريبية للتكاملات المحددة نعتمد عليها في الحالات عند استحالة إيجاد القيمة المضبوطة للتكامل كالتكامل المحدد عندما تكون عملية إيجاد القيمة المضبوطة للتكامل ممكنة ولكن بمشقة لانّ الدالة قد تكون معقدة وتحتاج إلى زمن طويل لإيجاد قيمة التكامل كالتكامل المحدد

قد تكون مسألتنا هي إيجاد مساحة تحت منحن معرّف بجدول قيم (أي أنّ الدالة معرّفة في نقاط معدودة في فترة التكامل) كما هو الحال عند تحليل نتائج التجارب .

ومن هذه الطرق :

1-3-8 قاعدة شبه المنحرف Trapezoidal rule

لحساب قيمة التكامل المحدد نقسم الفترة [a,b] إلى? n من الفترات الجزئية المتساوية

الطول فيكون طول كل فترة فتكون قاعدة شبه المنحرف كالآتي :

تعمل قاعدة شبه المنحرف بتقريب المنطقة تحت منحنى الدالة

$$\int_a^b f(x)\,dx \approx (b-a)\frac{f(a)+f(b)}{2}.$$

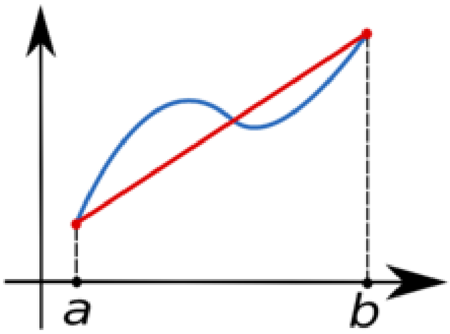
لحساب التكامل بدقة أفضل, يمكن فصل فترة التكامل $[a, b]$ أولا إلى n فترات أصغر, ومن ثم تطبيق قاعدة شبه المنحرف على كل فترة. يمكن تحصيل قاعدة شبه المنحرف المركب

$$\int_a^b f(x)\,dx \approx \frac{b-a}{n}\left[\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1} f\left(a+k\frac{b-a}{n}\right)\right].$$

ويمكن صياغة هذا بشكل اخر

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2n}\left(f(x_0)+2f(x_1)+2f(x_2)+\cdots+2f(x_{n-1})+f(x_n)\right)$$

$$x_k = a+k\frac{b-a}{n},\ \text{for } k=0,1,\ldots,n$$



قاعدة شبه المنحرف

ان الاستخدام النظري لطريقة شبه المنحرف لايجاد التكامل وكذلك طريقة شبه المنحرف المركب يأخذ وقت وجهد لذلك لذلك سنستخدم لغة البرمجة لحل هذه الطرق.

** تعتمد هذه الطريق على الدالة الموجودة ضمن الحزمة pracma . يجب تنزيل الحزمة ثم العمل بها .

مثال: اكتب برنامج لحساب التكامل العددي اي المساحة تحت منحنى الدالة $\int_0^{\pi} \sin(x)\, dx$ بطريقة شبه المنحرف وشبه المنحرف المركب باستخدام n=101

الحل:

البرنامج:-

```
Untitled1*
Source on Save
# Calculate the area under the sine curve from 0 to pi:
n <- 101
x <- seq(0, pi, len = n)
y <- sin(x)
pracma::trapz(x, y)                      #=> 1.999835504

# Use a correction term at the boundary: -h^2/12*(f'(b)-f'(a))
h  <- x[2] - x[1]
ca <- (y[2]-y[1]) / h
cb <- (y[n]-y[n-1]) / h
pracma::trapz(x, y) - h^2/12 * (cb - ca)  #=> 1.999999969
# Use two complex inputs
z  <- exp(1i*pi*(0:100)/100)
ct <- pracma::cumtrapz(z, 1/z)
ct[101]                              #=> 0+3.14107591i
```

مثال: اكتب برنامج لحساب التكامل العددي اي المساحة تحت منحنى الدالة $\int_0^{1.5} \sqrt{tan} + x\, dx$ بطريقة شبه المنحرف باستخدام n=6

الحل:

البرنامج:-

```
Untitled1*   Untitled2*
Source on Save
# باستخدام الدالة الجاهزة #######
n <- 6
x <- seq(0, 1.5, len = n)
y <- sqrt(tan(x))+x
pracma::trapz(x, y)

# باستخدام القانون اي الخوارزمية #### -h^2/12*(f'(b)-f'(a))
h  <- x[2] - x[1]
ca <- (y[2]-y[1]) / h
cb <- (y[n]-y[n-1]) / h
pracma::trapz(x, y) - h^2/12 * (cb - ca)
```

وعند تنفيذ البرنامج تكون مخرجاته كالاتي:

```
Console ~/
> # باستخدام الدالة الجاهزة #######
> n <- 6
> x <- seq(0, 1.5, len = n)
> y <- sqrt(tan(x))+x
> pracma::trapz(x, y)
[1] 2.921178
> # باستخدام القانون اي الخوارزمية #### -h^2/12*(f'(b)-f'(a))
> h  <- x[2] - x[1]
> ca <- (y[2]-y[1]) / h
> cb <- (y[n]-y[n-1]) / h
> pracma::trapz(x, y) - h^2/12 * (cb - ca)
[1] 2.881297
```

نلاحظ ان قيمة المساحة تحت منحنى الدالة مساوية الى 2.881297

مثال: اكتب برنامج لحساب التكامل العددي اي المساحة تحت منحنى الدالة $\int_0^{\frac{\pi}{2}} \sqrt{4+\sin(x)}\, dx$ بطريقة شبه المنحرف باستخدام n=4

الحل:

البرنامج:-

```
Untitled1* × Untitled2* ×
Source on Save
# باستخدام الدالة الجاهزة #######
n <- 4
x <- seq(0, pi/2, len = n)
y <- sqrt(4+sin(x))
pracma::trapz(x, y)


# باستخدام القانون اي الخوارزمية #### -h^2/12*(f'(b)-f'(a))
h  <- x[2] - x[1]
ca <- (y[2]-y[1]) / h
cb <- (y[n]-y[n-1]) / h
pracma::trapz(x, y) - h^2/12 * (cb - ca)
```

وعند تشغيل البرنامج يكون الناتج كالاتي:

```
Console ~/
> pracma::trapz(x, y) - h^2/12 * (cb - ca)
[1] 2.881297
> # باستخدام الدالة الجاهزة #######
> n <- 4
> x <- seq(0, pi/2, len = n)
> y <- sqrt(4+sin(x))
> pracma::trapz(x, y)
[1] 3.374731
> # باستخدام القانون اي الخوارزمية #### -h^2/12*(f'(b)-f'(a))
> h  <- x[2] - x[1]
> ca <- (y[2]-y[1]) / h
> cb <- (y[n]-y[n-1]) / h
> pracma::trapz(x, y) - h^2/12 * (cb - ca)
[1] 3.378708
>
```

ان قيمة التكامل مساوية الى 3.378708

4-8 الحل العددي للمعادلات التفاضلية

تعتبر المعادلات التفاضلية من الأدوات الرياضية الهامة في فهم العديد من المسائل الفيزيائية والهندسية والإجتماعية وقد إمتدت اهميتها موخراً الي حقول العلوم الإقتصادية وظهر ما يسمي بالنمذجة الرياضية،وهناك نوعان من المعادلات التفاضلية:

1) المعادلات التفاضلية العادية **Ordinary Differential Equations (ODEs)**

2) المعادلات التفاضلية الجزئية **Partial Differential Equations (PDEs)**

فالمعادلة التفاضلية العادية تحتوي علي متغير مستقل واحد أما المعادلة التفاضلية الجزئية تحتوي علي عدد من المتغيرات المستقلة (مثل درجة الحرارة $u(x,t)$ حيث تعتمد علي الموضع x والزمن t).

نرمز للمعادلة التفاضلية بالرمز $y^{(1)}$ y' $\frac{dy}{dx}$ وحلها يكون بالصيغة $y = g(x, c_1, c_2, ..., c_n)$.

حيث c_n عبارة عن ثوابت، ويشمل هذا النوع من المعادلات التفاضلية علي صنفين:

1) **مسائل القيم الابتدائية Initial Value Problem:**

في هذا النوع من المسائل تكون للمعادلة التفاضلية شرط إبتدائي (**initial Condition**) للمتغيرات والشرط الإبتدائي يمثل النقطة الإبتدائية التي تمر بها الدالة التي تمثل حل المعادلة التفاضلية.

2) **مسائل القيم الحدية Boundary Value Problem**

في هذه النوع من المسائل يكون للمعادلة التفاضلية شرط إبتدائي وشرط معين عند نهاية الفترة للمتغير المستقل وتمثل هذه الشروط نقطتين يجب أن تمر بهما الدالة التي تمثل حل المعادلة التفاضلية.

في هذا الجزء نتناول بعض الطرق العددية المستخدمة لحل المعادلة التفاضلية العادية مقتصرين علي معادلة الدرجة الأولي التي تكون علي الصيغة:

$$\frac{dy}{dx} = f(x,y)$$

ذات الشرط الابتدائي $y(x_0) = y_0$

تعتمد الطرق العددية علي معرفة المتغير التابع y في لحظة البدء x_0 ثم ننطلق من هذه النقطة خطوة خطوة إذ نحسب y_1 من أجل $x_0 + h$ و y_2 من أجل $x_1 + 2h$ حيث h تمثل التزائد الذي تأخذه x ويعرف بالصيغة $h = \frac{b-a}{M}$. وسنستخدم طريقة من طرق مسائل القيم الابتدائية وهي طريقة اويلر.

8–4–1 طريقة أويلر Euler's Method

تعتمد هذه الطريقة علي إعطاء الثابت h قيماً صغيرة بحيث يمكن حذف حدود سلسلة تايلر إبتداءاً من الحد الذي يحوي $\frac{h^2}{2!}y''(x)$ من سلسلة تايلر للنقطة $(x+h)$ والذي يعرف كالأتي :

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2!}y''(x) + \ldots\ldots\ldots\ldots + \frac{h^n}{n!}y^{(n)}.$$

وبحذف الحدود إعتباراً من الحد $\frac{h^2}{2!}y''(x)$ ينتج أن:

$$y(x+h) = y(x) + hy'(x) = y(x) + hf(x,y)$$

نبدأ من النقطة (x_0, y_0) وبالتعويض في العلاقة أعلاه ينتج أن:

$$y_1 = y(x_0 + h)$$
$$= y(x_0) + hy'(x_0)$$
$$y_2 = y(x_0 + 2h)$$
$$= y(x_0) + 2hy'(x_0)$$
$$\vdots$$
$$\vdots$$
$$y_{n+1} = y_n + hf(x_n, y_n) \qquad n = 0,1,2,3,\ldots\ldots, M-1$$

أي أن الصيغة العامة لقانون أويلر هي:

$$\boxed{\begin{aligned} & y_{n+1} = y_n + hf(x_n, y_n) \\ & \quad when \quad x_n = x_0 + nh \quad n = 0,1,2,3,\ldots\ldots, M-1 \end{aligned}}$$

مثال:

أوجد الحل التقريبي للمعادلة التفاضلية $y' = x + y$ حيث $y(0) = 0$ خذ $h = 0.2$ في الفترة $[0,1]$.

الحل:

بإستخدام القانون (1) ومن الشرط $y(0) = 0$ ينتج $y_0 = 0$ $x_0 = 0$ وبالتعويض في القانون نجد أن

$$f(x_n, y_n) = x_n + y_n \qquad x_n = x_0 + nh = nh$$

$$y_1 = y_0 + 0.2(x_0 + y_0) = 0$$
$$y_2 = y_1 + 0.2(x_1 + y_1) = 0 + 0.2(0.2 + 0) = 0.04$$
$$y_3 = y_2 + 0.2(x_2 + y_2) = 0.4 + 0.2(0.4 + 0.04) = 0.128$$
$$y_4 = y_3 + 0.2(x_3 + y_3) = 0.128 + 0.2(0.6 + 0.128) = 0.488$$
$$y_5 = y_4 + 0.2(x_4 + y_4) = 0.48 + 0.2(0.8 + 0.489) = 1.747$$

$$\boxed{\begin{aligned} & y_{n+1} = y_n + hf(x_n, y_n) \\ & \quad when \quad x_n = x_0 + nh \quad n = 0,1,2,3,\ldots.., M-1 \end{aligned}}$$

n	$x_n = 0.2n$	$y_{n+1} = y_n + 0.2(x_n + y_n)$
0	0	0
1	0.2	0.04
2	0.4	0.128
3	0.6	0.271
4	0.8	0.489
5	1	1.747

اما باستخدام البرنامج فيتم عمل دالة للخوارزمية ثم حفظ الدالة وتطبيقها لايجاد الحل المطلوب.

المثال النظري اعلاه

البرنامج:-

```
euler <- function(dy.dx=function(x,y){}, h=1E-7, y0=1, start=0, end=1) {
  nsteps <- (end-start)/h
  ys <- numeric(nsteps+1)
  ys[1] <- y0
  for (i in 1:nsteps) {
    x <- start + (i-1)*h
    ys[i+1] <- ys[i] + h*dy.dx(x,ys[i])
  }
  ys
}
dy.dx <- function(x,y) { x+y}
euler(dy.dx, start=0, end=1, h=0.2, y0=0)
```

لذلك يكون الناتج كأتي:-

```
> euler <- function(dy.dx=function(x,y){}, h=1E-7, y0=1, start=0, end=1) {
+ nsteps <- (end-start)/h
+ ys <- numeric(nsteps+1)
+ ys[1] <- y0
+ for (i in 1:nsteps) {
+ x <- start + (i-1)*h
+ ys[i+1] <- ys[i] + h*dy.dx(x,ys[i])
+ }
+ ys
+ }
> dy.dx <- function(x,y) { x+y}
> euler(dy.dx, start=0, end=1, h=0.2, y0=0)
[1] 0.00000 0.00000 0.04000 0.12800 0.27360 0.48832
> 
```

نلاحظ ان الحل نفس الناتج في الجدول السابق .

مثال : اكتب برنامج لايجاد الحل التقريبي للمعادلة التفاضلية $y' = 3x - y + 8$ حيث $y(0) = 3$ خذ $h = 0.1$ في الفترة $[0,0.5]$.

الحل:

البرنامج :-

```r
euler <- function(dy.dx=function(x,y){}, h=1E-7, y0=1, start=0, end=1) {
  nsteps <- (end-start)/h
  ys <- numeric(nsteps+1)
  ys[1] <- y0
  for (i in 1:nsteps) {
    x <- start + (i-1)*h
    ys[i+1] <- ys[i] + h*dy.dx(x,ys[i])
  }
  ys
}
dy.dx <- function(x,y) { 3*x - y + 8 }
euler(dy.dx, start=0, end=0.5, h=0.1, y0=3)
```

وعند التنفيذ يكون الناتج كالاتي:

```
> euler <- function(dy.dx=function(x,y){}, h=1E-7, y0=1, start=0, end=1) {
+ nsteps <- (end-start)/h
+ ys <- numeric(nsteps+1)
+ ys[1] <- y0
+ for (i in 1:nsteps) {
+ x <- start + (i-1)*h
+ ys[i+1] <- ys[i] + h*dy.dx(x,ys[i])
+ }
+ ys
+ }
> dy.dx <- function(x,y) { 3*x - y + 8 }
> euler(dy.dx, start=0, end=0.5, h=0.1, y0=3)
[1] 3.00000 3.50000 3.98000 4.44200 4.88780 5.31902
>
```

الفصل التاسع

المحاكاة simulation

1-9 المقدمة

إن إستخدام اسلوب المحاكاة في تمثيل الجوانب العملية يؤدي دوراً مهماً في معالجة المشكلات والمعضلات وتنفيذها وخاصة بعد التطور الواسع والكبير في مجال الحاسبات الإلكترونية مما دفع بالكثير من الباحثين الى إعتماد اسلوب المحاكاة في الكثير من البحوث التي تهدف الى دراسة سلوك أية مقدرات أو احصاءات اختبار أو أنموذج أو توزيع احصائي نظراً لصعوبة معرفة ذلك نظرياً.

وتعرف المحاكاة بأنها تقليد أو تمثيل للواقع من خلال استخدام أو تصميم نماذج معينة للنظام الحقيقي ومتابعة تنفيذ التجربة للتعرف على مخرجات هذا النظام.

ومن أهم طرائق المحاكاة وأكثرها شيوعاً في التحليل هي طريقة مونت كارلو (Monte Carlo)، التي تستعمل في توليد مشاهدات لمعظم التوزيعات الاحتمالية المعروفة.

2-9 محاكاة (Monte Carlo)

ان طرق مونت كارلو (Monte Carlo methods) هي مجموعة من الخوارزميات الحسابية اللائي تتضمن تكرار التجربة بقيم بدائية عشوائية. تستخدم هذه الطريقة عادة في أنظمة المحاكاة الرياضية والهندسية. تتضمن هذه الطريقة خمسة مراحل:

1- تحديد المجال الممكن لقيم الإدخال
2- توليد قيم عشوائية لقيم الإدخال ضمن الحدود المعروفة
3- تطبيق العمليات الحسابية المطلوبة على تلك القيم
4- مراكمة النتائج الحالية مع النتائج السابقة

5- تكرار العملية عدد محدد من المرات (تزداد دقة النتائج مع زيادة عدد التكرارات)

تمثل طريقة مونت كارلو عاملا مهما لمحاكاة الأنظمة ذات أزواج من درجات المرونة (many coupled degrees of freedom) كالسوائل، و المواد غير نظامية التركيب و الصلبة ذات قوة الربط الكبيرة والأبنية الخلوية.

من الأمثلة الأخرى على الظواهر التي يصعب التنبؤ بها هي بعض الحسابات التجارية، التي تكون نماذج محاكاتها مشوبة بنقص الدقة (uncertainty) . ومن الأمثلة في الرياضيات، تقييم التكاملات الثلاثية الأبعاد بالعوامل الحدودية المركبة. وفي علم الأبحاث الزيتية، تساعد محاكاة مونت كارلو بالتنبؤ بالأخطاء، كتغير الأسعار أو تغير الجدول الزمني وتؤدي إلى نتائج أفضل مما ينتج عن طريق الحدس أو الطرق البسيطة الأخرى.

مبدأيا، يمكن تطبيق طريقة مونت كارلو على أي مشكلة يتخللها تعدد الإحتمالات. عبر قانون الأعداد الكبيرة يمكن تقريب حسابات التكامل لمتغير عشوائي عبر أخذ متوسط القيمة التجريبية (empirical mean) للقيم. عندما يكون التوزيع الاحتمالي مركب جدا، فيتم اللجوء الى طريقة ماركوف شين مونت كارلو (Markov Chain Monte Carlo MCMC). الفكرة الأساسية هي تصميم نظام دقيق ذكي وفق ماركوف شين عبر قيم توزيع احتمالي ثابت. وفقا لنظرية ارجوديك، التوزيع الاحتمالي الساكن (stationary) يجري تقريبه عبر قياسات تجريبية للمخرجات العشوائية المأخوذة عبر عينات كاركوف شين.

3-9 تولد الارقام العشوائية

تفيد الأرقام العشوائية في مجموعة متنوعة الأغراض والأهداف، مثل توليد مفاتيح لتشفير البيانات، والنمذجة، ومحاكاة الظواهر المعقدة، واختيار عينات عشوائية من مجموعات البيانات الضخمة. كما تم استخدامها من الناحية الجمالية في الأدب والموسيقى. وبالطبع هي أيضاً مشهورة الاستخدام في الألعاب . الرقم العشوائي هو أحد الأرقام التي يتم الحصول عليها من مجموعة من القيم الممكنة، كل رقم من المجموعة له احتمال متساوٍ مع البقية للحصول عليه، أي تتوزع احتمالاتها توزيعاً منتظماً. عند مناقشة سلسلة من الأرقام العشوائية، يجب أن يكون كل عدد مستخرج مستقل إحصائياً عن الآخرين، أي أن الحصول على عدد ما لا يؤثر على احتمال الحصول على عدد آخر.

ومع ظهور الحواسيب، احتاج المبرمجون إلى وسيلة لاستحداث وتوليد العشوائية في برامج الحاسوب. وعلى رغم من ذلك، من الصعب إيجاد حاسوب يقوم بشيء مصادفة أو عشوائياً. لأن الحاسوب ينفذ التعليمات بصورة عمياء محددة له مسبقاً، وبالتالي يمكن التنبؤ بها تماماً. هناك نوعان من الطرق الرئيسية لتوليد الأرقام العشوائية باستخدام الحاسوب:

1- مولدات أرقام شبه عشوائية (Generators Numbers Random Pseudo)

2- مولدات أرقام عشوائية حقيقية (Generators Numbers Random True) لكل طريقة خصائص مختلفة تماماً عن الأخرى ولكل منها إيجابياتها وسلبياتها. سنتحدث قليلاً عن كلا النوعين.

4-9 توليد أرقام شبه عشوائية (Generators Of Pseudorandom (Numbers

تتضمن هذه الفقرة استكشاف ميكانكية توليد الارقام شبه العشوائية على وجه الخصوص ،سوف نصف طريقة من ابسط الطرق وهي طريقة multiplicative (congruential generator) لتوليد ارقام عشوائية تقع ضمن الفترة [0,1] ، حيث تمثل الارقام $u0, u1, u2, \ldots,$ ارقام شبه عشوائية منتظمة تقع ضمن الفترة المحددة نفرض ان m عدد صحيح كبير و نفرض b ايضا عددا صحيحا والذي يجب ان يكون اصغر من m .وغالبا ما نختار قيمة b بحيث تكون قريبة من الجذر التربيعي لقيمة m ، وان اعداد مختلفة من m ومن b تولد لنا ارقام شبه عشوائية بنوعية مختلفة . هناك معايير مختلفة متاحة لاختيار قيم جيدة من m , وb ولكن المهم هو اختبار نتائج التوليد للتأكد من أنه تم تقديم نتائج معقولة.

الطريقة تبدا بختيار قيمة اولية تمثل عدد صحيح لتكن x_0 تقع بين 1و m لذلك سيتم المتغيرات العشوائية بالاعتماد الكلي عل قيم x_0,m,b, وكما يأتي:

$$x_1 = bx_0 (mod\ m)$$

$$u_1 = x_1/m$$

حيث ان u_1 يمثل الرقم شبه العشوائي الاول ولتوليد عدد شبه عشوائي ثاني سيعتمد على قيمة x_1 وكما يأتي:-

$$x_2 = bx_1 (mod\ m)$$

$$u_2 = x_2/m$$

يمثل u_2 الرقم شبه العشوائي الثاني

حيث إرجاع عامل التشغيل *'mod'* في الصيغة أعلاه باقي عدد صحيح بعد قسمة عدد صحيح.

ونستمر بالخطوات الى الن نحصل على القاعدة العامة وهي

$$x_n = b\, x_{n-1}\ (\mathrm{mod}\ m)$$

$u_n = x_n/m.$

هذه الطريقة تنتج الأرقام التي هي تحديدا بشكل كامل، ولكن الملاحظ الذي لا يعرف الصيغة أعلاه، تظهر الأرقام، أن تكون عشوائية ولا يمكن التنبؤ بها، على الأقل خلال المدى القصير.

مثال : ولد 6 ارقام عشوائية بحيث ان قيمة $m=7$ وقيمة $b=3$ وبقيمة اولية $x0 = 2$؟

الحل :

$$x1 = 3 \times 2 \pmod 7 = 6,\ u1 = 0.857$$
$$x2 = 3 \times 6 \pmod 7 = 4,\ u2 = 0.571$$
$$x3 = 3 \times 4 \pmod 7 = 5,\ u3 = 0.714$$
$$x4 = 3 \times 5 \pmod 7 = 1,\ u4 = 0.143$$
$$x5 = 3 \times 1 \pmod 7 = 3,\ u5 = 0.429$$
$$x6 = 3 \times 3 \pmod 7 = 2,\ u6 = 0.286$$

ويمكن التحقق من هذه الاقام باستخدام حاسبة العلمية بحيث تحتوي على الدالة MOD .

ويمكن توليد 50 رقم بهذه الطريقة باستخدام لغة R بحث ان قيمة $m=30269$ وان $b=171$ وبقيمة اولية $x0 = 27\,218$.

```
random.number <- numeric(50) # 3   ليتم اخراجها
random.seed <- 27218
for (j in 1:50) {
random.seed <- (171 * random.seed) %% 30269
random.number[j] <- random.seed / 30269
}
random.number
```

لاحظ تم برمجة الخطوات طريقة (multiplicative congruential generator)

لاحظ ان عامل التشغيل*'mod'* في لغة R يشار له ب (%%) ويكون الناتج كألاتي:-

```
Console ~/
> random.number <- numeric(50) # 3  خزن 50 قيمة عددية ليتم اخراجها####
> random.seed <- 27218
> for (j in 1:50) {
+ random.seed <- (171 * random.seed) %% 30269
+ random.number[j] <- random.seed / 30269
+ }
> random.number
 [1] 0.76385080 0.61848756 0.76137302 0.19478675 0.30853348 0.75922561 0.82757937
 [8] 0.51607255 0.24840596 0.47741914 0.63867323 0.21312234 0.44391952 0.91023820
[15] 0.65073177 0.27513297 0.04773861 0.16330239 0.92470845 0.12514454 0.39971588
[22] 0.35141564 0.09207440 0.74472232 0.34751726 0.42545178 0.75225478 0.63556774
[29] 0.68208398 0.63636063 0.81766824 0.82126929 0.43704780 0.73517460 0.71485678
[36] 0.24051009 0.12722587 0.75562457 0.21180085 0.21794575 0.26872378 0.95176583
[43] 0.75195745 0.58472364 0.98774324 0.90409330 0.59995375 0.59209092 0.24754700
[50] 0.33053619
```

نلاحظ تم توليد 50 قيمة شبه عشوائية بالاعتماد على طريقة (MCG)

ويمثل هذه العملية من خلال استخدام طرق مختلفة وباستخدام دورات اطول من ذلك بكثير وبالاستخدام الداخلي بواسطة لغة R يتم توليد ارقام شبه عشوائية بشكل تلقائي من خلال الدالة .runif() . ودوال اخرى تخص توزيعات احصائية اخرى .

9-5 اهم الايعازات الخاصة بتوليد الارقام شبه العشوائية الخاصة بالتوزيعات الاحتمالية

9-5-1الايعاز runif

يستعمل هذا الايعاز لتوليد ارقام شبه عشوائية تتبع التوزيع المنتظم بالمعلمتين (a,b) والشكل العام للايعاز هو :

runif (n, min = a, max = b)

n:- عدد الارقام المراد توليدها

a: الحد الادنى للفترة

b : الحد الاعلى للفترة

بمجرد تنفيذ هذا الامر ينتج ارقام منتظمة شبه عشوائية ضمن الفترة [a,b] .

مثال: ولد 5 ارقام شبه عشوائية منتظمة ضمن الفترة [0,1] ؟

الحل : نستخدم الدالة اعلاه لتوليد هذه القيم وكما يأتي :

```
Console ~/
> runif(5)
[1] 0.85330462 0.54579517 0.38967584 0.93587475
[5] 0.05189955
```

نلاحظ الارقام التي تم توليدها تقع بين 0 و 1 .

مثال: ولد 10 ارقام شبه عشوائية منتظمة تقع ضمن الفترة [-3,-1] ؟

الحل : باستخدام الدالة

```
Console ~/
> runif(10, min = -3, max = -1)
 [1] -2.608186 -1.641711 -2.367757 -2.378421 -2.129321
 [6] -1.571959 -1.161482 -2.276787 -2.412028 -1.429710
```

نلاحظ القيم التي تم توليدها تقع ضمن الفترة المحددة .

2-5-9 الايعاز rbinom

يستعمل هذا الايعاز لتوليد ارقام عشوائية تسلك وفق التوزيع المتقطع ثنائي الحدين **binomial** بالمعلمة p.

وصف التوزيع

إذا كانت هناك تجربة عشوائية لها نتيجتان فقط هما ظهور حدث معين نجاح او عدم ظهور فشل مثل نجاح الطالب او فشلة، المصباح الكهربائي جيد او تالف، وصول طائرة في موعدها او عدم وصولها، ظهور الصورة عند الغاء قطعة نقود او عدم ظهورها. و إذا أجريت هذه التجربة m من المرات باحتمال نجاح p . و احتمال فشل q =1-p حيث p + q = 1 ونفرض أن x هو التغير العشوائي المعرف على هذه

التجربة لذلك تعطى دالة كثافة الاحتمال للمتغير العشوائي X و التي نرمز لها بالرمز f (x) بالمعادلة الاتي:

$$p\left(X=x\right)= f(x)=\begin{Bmatrix} \binom{m}{x} P^{x}(1-p)^{m-x} & x=0,1,2,....,n \\ 0 & else\ where. \end{Bmatrix}$$

حيث n عدد صحيح موجب، وتحت هذه الشروط واضح أن:

$0 \langle\, p \,\langle 1$

$f(x) \geq 0$

وبعد وصف التوزيع يكون الشكل العام للإيعاز هو

rbinom(n, size, prob)

اذ ان :

n : يمثل عدد الارقام المراد توليدها .(حجم العينة)

Size : تمثل عدد المحاولات للتجربة.

Prob : تمثل احتمال النجاح .

مثال: لنفترض أن 10٪ من الأنابيب التي تنتجها آلة معيبة، وافترض يتم إنتاج 15 أنابيب كل ساعة. كل أنبوب مستقل عن جميع الأنابيب الأخرى. ويحكم هذه العملية أن تكون خارج نطاق السيطرة عندما يكون الانتاج اكثر من أربعة أنابيب معيبة في أي ساعة. اجري محاكاة عدد أنابيب المعيبة التي تنتجها آلة لكل ساعة على مدار 24 ساعة؟

الحل :

يتم إنتاج 15 أنابيب كل ساعة وكل أنبوب يكون احتماله 0.1 كونه معيب ومستقل عن انتاج الأنابيب الاخرى فان عدد الانابيب المعيبة المنتجة في ساعة واحدة هو متغير عشوائي يتبع توزيع متقطع ثنائي الحدين بالمعلمات عدد المحاولات 15 محاولة وباحتمال 0.1 لمحاكاة عدد الانابيب المعيبة في كل ساعة خلال 24 الساعة .لذلك بحاجة الى توليد 24 رقم عشوائي يسلك وفق ذي الحدين ثم التعرف على جميع الحالات التي تتجاوز فيها عدد المعيب فيها 5 انابيب . نلاحظ التطبيق.

نلاحظ ان في خلال الساعة الاولى ولا انبوب معيب وفي الساعة الثانية ظهر انبوب معيب واحد وفي الساعة الثالثة ولا معيب والرابعة 4 معيب وهكذا وقمنا باختبار هل يوجد اكثر من 5 معيب خلال 24 ساعة نلاحظ النتيجة false يعني ولا ساعة من 24 ساعة فيها 5 معيب وهذا واضح من خلال المحاكاة.

```
Console ~/
>  defectives <- rbinom(24, 15, 0.1)
>  defectives
 [1] 0 1 0 4 4 1 1 2 1 1 1 1 1 1 2 1 0 2 2 1 2 4 0 1
>
>
> any(defectives > 5)
[1] FALSE
>
```

3-5-9 الايعاز rpois

يستعمل هذا الايعاز لتوليد ارقام عشوائية تتوزع وفق التوزيع بواسون بالمعلمة λ .

وصف التوزيع

في الحياة العملية أحيانا ما نقابل بعض الظواهر التي ينطبق عليها شروط توزيع ذي الحدين و لكن هذه الحوادث تكون نادرة الوقوع و هذا يعنى أن احتمال النجاح يكون صغير جدا يقترب من الصفر و علية فأنه يمكن القول أن $np=\lambda$ حيث λ هى مقدار ثابت و بذلك يكون احتمال الفشل كبير أي أنه يقترب من الواحد. و لكى نراقب بعض حالات النجاح فأننا سنجد أن n سوف تكون كبيرة جدا فمثلا لو اردنا حساب

احتمال خروج القطار من على الشريط " القضبان" فأننا سنقوم بمراقبة القطارات او عدد كبير جدا منها و نحسب عدد مرات خروج القطار من على الشريط أي حالات النجاح (التي حققت فيها الحادثة) حتى نستطيع أن نحسب الاحتمال.

و بذلك تكون شروط هذا التوزيع كالآتي:

1ـ أن تكون احتمال النجاح ثابت و كذلك احتمال الفشل في كل محاولة و يرمز لهما بالرمز q, p على التوالي.
2ـ أن يكون احتمال النجاح صغيرا و يقترب من الصفر و احتمال الفشل يقترب من الواحد الصحيح.
3ـ أن تكون عدد المحاولات كبيرا جدا حيث أن $= n\,p = \lambda$ مقدار ثابت.

و يعتبر توزيع بواسون من التوزيعات الاحتمالية المتقطعة وسمي هذا التوزيع بهذا الاسم نسبة الى أحد مكتشفة و هو بواسون و يعتبر من اهم التوزيعات في المسائل المتعلقة بالمكالمات التليفونية و حركة المرور، بعض الظواهر النادرة مثل الزلزال، و الحرائق، الحوادث على أحدى الطرق، عدد الاخطاء المطبعية في صفحة ما من كتاب و غير ذلك. و دالة كثافة الاحتمال لتوزيع بواسون هي :

$$f(x) = \begin{cases} \dfrac{\lambda^x \; e^{-\lambda}}{x\,!} & , \qquad x = 0, 1, 2, 3, \dots\dots\dots\dots \\ 0 & , \end{cases}$$

حيث e = 2.718 تمثل مقدار ثابت.

$$\lambda = n \ p$$

و نأخذ X قيماً صحيحة موجة اعتباراً من الصفر الى مالانهاية.

وبعد وصف التوزيع نوضح الشكل العام للإيعاز :

rpois(n, lambda)

اذ ان :

n: تمثل عدد الاقام شبه العشوائية المراد توليدها (حجم العينة)

Lambda : تمثل معلمة التوزيع

مثال: نفرض انه لدينا حوادث المرور التي تحدث في التقاطعات بمتوسط 3.7 لكل سنة .اجري تجربة محاكاة بالحوادث المرورية لمدة 10 سنوات .

الحل :

```
> rpois(10,3.7)
 [1] 4 7 0 1 3 2 1 6 7 4
>
```

لاحظ في السنة الاولى معدل الحوادث 4 والثانية 7 الى العاشرة 4

9-5-4 الايعاز rexp

يستعمل هذا الايعاز لتوليد ارقام شبه عشوائية تتوزع وفق التوزيع الاسي بالمعلمة λ

وصف التوزيع

عادة ما يستخدم التوزيع الأسي في مسائل متعلقة بقياس الزمن. من ذلك مدة خدمة شباك البريد، مدة مكالمة هاتفية، مدة تفريغ باخرة شحن، مدة تصليح آلة، مدة انتظار زبون قبل الحصول على الخدمة...في العلوم الدقيقة يستخدم التوزيع الأسي لتمثيل مدة

حياة الذرات المشعة (atomes radioactives) قبل أن تتفكك، حيث يعبر الوسيط عن اللحظة التي يبقى فيها نصف المجتمع الأصلي[1].

مثلا قد نستبعد استخدام التوزيع الأسي لتمثيل مدة حياة آلة عاملة قبل تعطلها لأن احتمال تعطلها في لحظة ليس مستقلا عن المدة التي عملتها الآلة من قبل، كذلك الأمر بالنسبة لمدة حياة الإنسان.

نشير أخيرا إلى أن للتوزيع الأسي علاقة بالتوزيع بواسون، فإذا كان وقوع أحداث ما يتبع هذا التوزيع، فإن المدة بين وقوع حدثين تتبع التوزيع الأسي؛ كمثال على ذلك، إذا كان وصول الزبائن إلى مركز خدمة ما يتبع التوزيع بواسون فإن المدة الزمنية بين وصول زبون "أ" والزبون الموالي تتبع التوزيع الأسي. وان دالة الكثافة الاحتمالية للتوزيع الاسي تأخذ الصيغة الاتية :

$$f(X = x) = \lambda e^{-\lambda x} \quad , x, \lambda > 0$$

اذ ان :

x: يمثل المتغير العشوائي

λ : معلمة التوزيع .

وبعد وصف التوزيع نوضح الشكل العام للإيعاز :

rexp(n, rate)

rate : تمثل معلمة التوزيع

مثال: في البنك راوي واحد الذي يوجه طابور من 10 زبائن. الوقت لكل زبون في الحصول على الخدمة يتوزع وفق التوزيع الاسي ، مع معدل 3 لكل دقيقة .اجري محاكاة أوقات الخدمة (بالدقائق) الى 10 زبائن.

الحل:

```
Console ~/
> servicetimes <- rexp(10, rate = 3)
> servicetimes
 [1] 0.30100945 0.03824615 0.02939367 0.47188319 0.38072813 0.44267687 0.32697236
 [8] 0.48148227 0.23484651 0.01389987
>
```

الوقت الكلي حتى الحصول على لخدمة لكل الزبائن العشرة سيكون حوالي 3 دقايق و16 ثانية

```
> sum(servicetimes)
[1] 3.273096
```

5-5-9 الايعاز rnorm

يستعمل هذا الايعاز في لغة R لتوليد ارقام شبه عشوائية تتوزع وفق التوزيع الطبيعي والطبيعي القياسي

وصف التوزيع الطبيعي

يعد التوزيع الطبيعي من أهم التوزيعات الاحتمالية شائعة الاستخدام لما له من خصائص تنطبق على نسبة كبيرة من الظواهر الطبيعية والاجتماعية والاقتصادية. فلو اخترنا بالصدفة مئة أو ألفا من المارين في شار ع ما وقسنا أطوالهم لوجدنا نسبة كبيرة منها قريبة من متوسط ما، ونسبة قليلة من طوال القامة ونسبة مقاربة لها من قصار القامة. ومثل هذا بالنسبة للأوزان. ولو مثلنا هذه البيانات في معلم متعامد متجانس لكان المنحنى الذي يمثل النسبة، أو ما يمكن أن نسميه الاحتمال، اذ شكل جرسي متماثل حول المتوسط وهي صفات التوزيع الطبيعي كما موضح في الشكل ادناه:

وتكتب دالة الكثافة لمنحنى للتوزيع الطبيعي كما يلي:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(1/2)\left(\frac{x-\mu}{\sigma}\right)^2} \qquad -\infty < x < \infty$$

حيث μ و σ هما على التوالي التوقع والانحراف المعياري

التوزيع الطبيعي القياسي

يعتبر التوزيع الطبيعي القياسي حالة خاصة من التوزيع الطبيعي عندما يكون المتوسط يساوي صفر والانحراف المعياري يساوي واحد ويمتلك دالة كثافة احتمالية هي كالاتي:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \qquad -\infty < z < \infty$$

وبعد وصف التوزيع نوضح الايعاز :

rnorm(n, mean, sd)

اذ ان :

n: يمثل حجم العينة

mean :- تمثل الوسط الحسابي

Sd :- يمثل الانحراف المعياري .

تعتبر هذه الدالة من اهم الدوال واكثرها استخداما من قبل الباحثين لأنها تتميز بتوليد ارقام شبه عشوائية تتبع توزيع الطبيعي والتوزيع الطبيعي القياسي .

مثال: ولد 10 ارقام عشوائية تتبع وفق التوزيع الطبيعي بمتوسط 3- وانحراف معياري 0.5؟

الحل:

```
Source

Console ~/
> rnorm(10, -3, 0.5)
 [1] -3.104346 -3.323958 -3.165461 -3.326440 -2.563451 -3.762103 -2.901598 -2.572396
 [9] -3.750027 -2.761158
> |
```

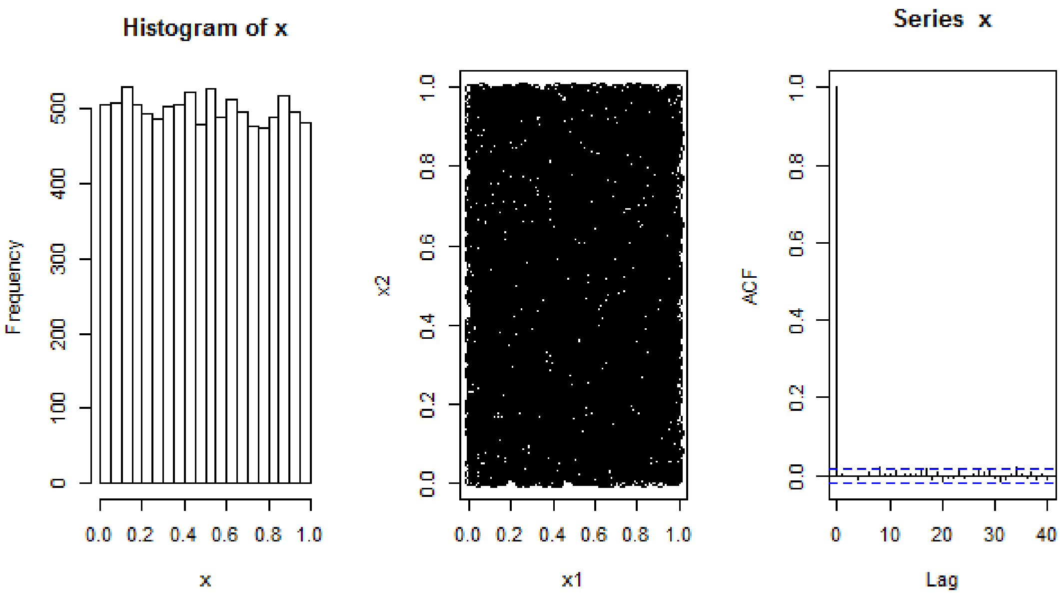
6-9 الطرق النظرية لتوليد المتغيرات العشوائية

1-6-9 التوزيع المنتظم

ذكرنا انه يتم توليد ارقام عشوائية تسلك وفق توزيع منتظم في الفترة [a,b] بأستخدام الدالة runif ولغرض معرفة خصائص التوليد لهذا التوزيع نعمل الاتي:

1- نولد المتغيرات Xi's باستخدام الدالة الخاصة بالتوزيع المنتظم
2- نرسم الزوج المرتب (Xi,Xi+1)
3- لاحظ تقدير دالة الارتباط

يكون الناتج كما يأتي:



```
Nsim=10^4          #عدد ارقام العشوائية
x=runif(Nsim)
x1=x[-Nsim]          #متجهات الرسم
x2=x[-1]
par(mfrow=c(1,3))
hist(x)
plot(x1,x2)
acf(x)
```

نلاحظ ان رسم المدرج التكراري على اليسار وفي المركز رسم يمثل الازواج من x1 وx2 والرسم الذي على اليمين يمثل تقدير دالة الارتباط بين المتغيرات التي تم توليدها باستخدام الدالة runif

2-6-9 طريقة التحويل المعكوس:

وهي الطريقة الأكثر استخداما وخاصة للتوزيعات الاحتمالية التي يمكن إيجاد $F^{-1}(x)$ لها وكذلك للتوزيعات التجريبية Empirical Distributions

وتقوم على التطابق $U = F(X)$

$$F(x) = \int_{-\infty}^{x} f(u)du$$

ثم نجد الحل بالنسبة ل x

مثال: ولد متغير عشوائي يتبع التوزيع الاسي بالمعلمة $\lambda = 1$ باستعمال طريقة التحويل المعكوس ثم ارسم مدرج تكراري يوضح التوزيع.

الحل: اذا كانت لدينا دالة cdf للتوزيع هي كما في المعادلة الاتية :

$$F(X) = 1 - e^{-x}$$

$$U = F(X)$$

وبحل المعادلة الاتية بالنسبة الى x وكما يلي:

$$U = 1 - e^{-x}$$

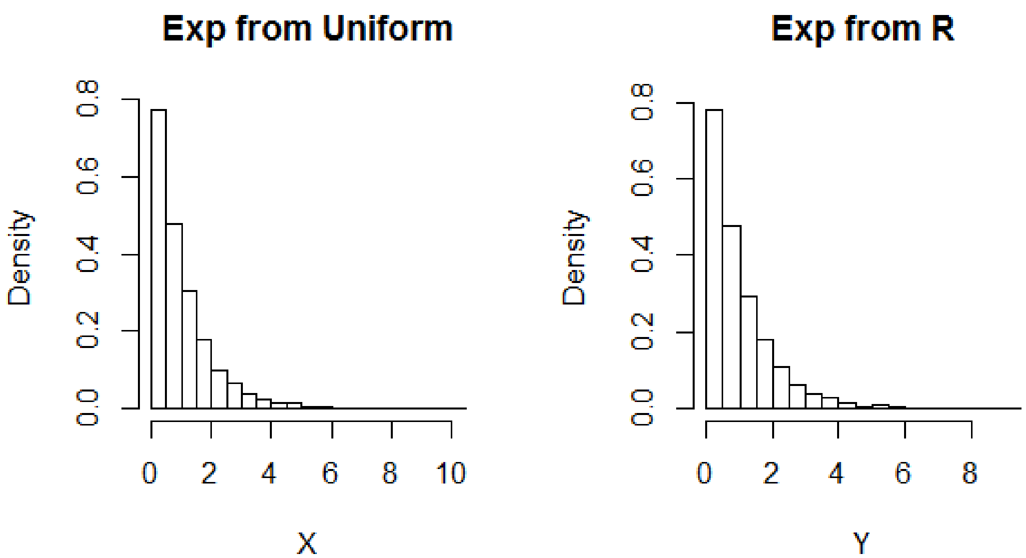
$$x = -\log(u)$$

لان سنقوم برسم طريقتي التوليد باستخدام البرنامج الاتي:

```
Untitled1*
Source on Save

Nsim=10^4          #number of random variables
 U=runif(Nsim)
 X=-log(U)         #transforms of uniforms
 Y=rexp(Nsim)      #exponentials from R
 par(mfrow=c(1,2))   #plots
 hist(X,freq=F,main="Exp from Uniform")
 hist(Y,freq=F,main="Exp from R")
```

يكون الناتج كما يأتي:



نلاحظ ان الطرقتين متكافئة في توليد الارقام العشوائية .

3-6-9 طرق التحويل العامة :

عندما تكون هناك علاقة بين توزيعين من التوزيعات العشوائية من خلال دالة الكثافة الاحتمالية يمكن الاستفادة من هذه العلاقة في بناء خوارزمية لمحاكاة من خلال دالة الكثافة الاحتمالية وهذا يعني محاكاة توزيع معين بدلالة توزيع اخر.

فاذا كان لدينا Xi's متغيرات مستقلة ومتماثلة التوزيع تتوزع توزيع اسي Exp(1) فانه يمكن من خلالها اشتقاق ثلاث توزيعات معيارية وكما ياتي:

$$Y = 2\sum_{j=1}^{\nu} X_j \sim \chi^2_{2\nu}\,, \qquad \nu \in \mathbb{N}^*\,,$$

$$Y = \beta\sum_{j=1}^{a} X_j \sim \mathcal{G}(a,\beta)\,, \qquad a \in \mathbb{N}^*\,,$$

$$Y = \frac{\sum_{j=1}^{a} X_j}{\sum_{j=1}^{a+b} X_j} \sim \mathcal{B}e(a,b)\,, \qquad a,b \in \mathbb{N}^*\,,$$

where $\mathbb{N}^* = \{1,2,\ldots\}$.

وسنأخذ التوزيع الاول والذي يمثل مربع كاي :

الخوارزمية

1- توليد U وفق التوزيع المنتظم بالفترة [0,1]
2- توليد مصفوفة الجمع ل U
3- توليد X يسلك وفق التوزيع الاسي من العلاقة X=-log(U)

البرنامج:-

```
U=runif(3*10^4)
U=matrix(data=U,nrow=3) #matrix for sums
X=-log(U) #uniform to exponential
X=2* apply(X,2,sum) #sum up to get chi squares
X1=rchisq(10^4,df=6)
X
X1
```

حيث ان X يمثل توليد مربع كاي باستخدام الخوارزمية، X1 يمثل توليد مربع كاي باستخدام دالة داخلية في لغة البرمجة R

4-6-9 طريقة (Box-Muller)

تستعمل طريقة Box-Muller لتوليد متغيرات تسلك وفق التوزيع الطبيعي بالمتوسط والانحراف المعياري وتتلخص الطريقة بالخوارزمية الاتية:

الخوارزمية:

الخطوة الاولى:- ولد متغيرين مستقلين u_1 و u_2 يتوزعان U(0,1)

الخطوة الثانية :- اجعل X_1 و X_2 كالآتي:

$$X_1 = \sqrt{-2\log(U_1)}\ \cos(2\pi U_2)\ , \qquad X_2 = \sqrt{-2\log(U_1)}\ \sin(2\pi U_2)\ ,$$

مثال: اكتب برنامج لتوليد 1000 قيمة للمتغير x تتبع توزيع طبيعي قياسي باستخدام طريقة Box-Muller؟

الحل:

البرنامج:

```
rm(list=ls())
n<-1000
for(i in 1:n){
   u1<-runif(1)
u2<-runif(1)
z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)
u3<-runif(1)
if(u3<0.5){
   x<-z1}
else{x<-z2}
print(x)
}
```

5-6-9 طريقة الرفض والقبول

ان ما نريد عمله هو ايجاد صيغة واضحة لدالة الكثافة التجميعية للمتغير العشوائي x ونرغب في توليد $F(x) = P(X \leq x)$ ولكن ليس دائما نسطيع توليدها لكن يتم ذلك بطرق بديلة لتوليد متغيرات عشوائية تتوزع بالاعتماد على F ومن هذه الطرق هي طريقة الرفض والقبول التي هي اكثر كفاءة من التحويل المعكوس والطرق الاخرى.وان هذه الطريقة تستخدم لتوليد متغيرات تسلك وفق التوزيع الطبيعي القياسي.

خوارزمية الطريقة

الخطوة الاولى :- ولد u1 و u2 يتوزعان وفق التوزيع المنتظم U(0,1)

الخطوة الثانية :- ولد y1 وy2 وفق التوزيع لاسي حسب الصيغة الاتية :-

Y1=-log(u1)   y2=-log(u2)

الخطوة الثالثة:-اذا كان $y2-(y1-1)^2 \leq 0$ اذهب الى الخطوة الاولى

الخطوة الرابعة :- ولد المتغير العشوائي $u3$ الذي يتوزع حسب التوزيع المنتظم $U(0,1)$

الخطوة الخامسة :- اجعل

$$Z=\begin{cases}-y1 & if\ u3>0.5\\ y1 & if\ \ u3\leq 0.5\end{cases}$$

الخطوة السادسة :- اجعل $x=\mu+\sigma z$

البرنامج:

```
av=45
std=2
n<-100
h=0
while(h<n){
  u1=runif(1)
  u2=runif(1)
  y1=-log(u1)
  y2=-log(u2)
  if(y2-(y1-1)^2>0){
    u=runif(1)}
  if(u>0.5){z=y1}else{z=-y1}
  x=av+std*z
  h=h+1
  print(x)
}
m=mean(x)
m
```

9-6-6 طريقة polar

قام الباحث POLAR بالتعديل على طريقة Box & Muller وذلك بالابتعاد عن الدوال المثلثية التي استخدمت في طريقة ***(Box-Muller)*** والتي تستخدم لتوليد بيانات تسلك وفق التوزيع الطبيعي القياسي وحسب الخوارزمية الاتية:-

الخوارزمية

الخطوة الاولى :- ولد v1 ,v2 متغيرين عشوائيين مستقلين يتوزعان حسب التوزيع المنتظم U(-1,1) .

الخطوة الثانية :- اجعل $R^2 = V1^2 + V2^2$

الخطوة الثالثة :- اذا كان $R^2 \geq 1$ اذهب الى الخطوة الاولى

عدا ذلك

$$Z1 = V1\sqrt{-2\log(R^2)/}R^2$$

$$Z2 = \sqrt{-2\log(R^2)/R^2}$$

مثال:- اكتب برنامجا لتوليد 100 قيمة تتبع التوزيع الطبيعي بمتوسط 27 وانحراف معياري 0.25 باستعمال طريقة polar ثم احسب الوسط الحسابي؟

البرنامج:-

Untitled1*

Source on Save

```r
av=27
std=0.25
n=100
for(i in 1:n){
  k=0
  while(k==0){
  v1=runif(1,-1,1)
  v2=runif(1,-1,1)
  r=sqrt(v1^2+v2^2)
    if(r<1){
      z1=v1*sqrt(-2*log(r^2)/r^2)
      z2=v2*sqrt(-2*log(r^2)/r^2)
    }else{k=k+1}
  }
  x1=av+std*z1
  x2=av+std*z2
  print(x1)
  print(x2)
}
me=mean(x1)
me2=mean(x2)
me
me2
```

يكون الناتج كالاتي:

```
> me=mean(x1)
> me2=mean(x2)
> me
[1] 27.12669
> me2
[1] 27.11637
>
```

وللتأكد من صحة الطريقة نلاحظ ان الوسط الحسابي الذي ادخلناه هو 27 والوسط الحسابي المولد هو 27.1 وهذا هو دليل على صحة الطريقة .

7-6-9 توليد بيانات وفق توزيع كاما gamma

يعتبر توزيع كاما هو احد التوزيعات العشوائية المستمرة بمعلمتي $\alpha>0$ و $\beta>0$ وله دالة كثافة احتمالية بالصيغة الاتية :-

$$p(x)=\frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}\mathrm{e}^{-x/\beta}\quad \text{for } 0\leq x\leq\infty,$$

حيث ان $\Gamma(\alpha)$ تمثل دالة كاما وان α تمثل معلمة الشكل وان β تمثل معلمة القياس لهذا التوزيع ويمكن توليد بيانات تتبع توزيع كاما بالاعتماد على معلمة الشكل α وهناك طريقتين لذلك وهي:

1- في حالة α اكبر من 1
2- في حالة α اصغر من 1

لنأخذ **الطريقة الاولى** في حالة α اكبر من 1 حيث ان هذه خوارزمية كتبت من قبل الباحث Feast عام 1979 لتوليد بيانات تسلك وفق توزيع كاما بمعلمة الشكل اكبر من الواحد .

الخوارزمية

1. Generate u_1 and u_2 independently from U(0, 1), and set
$$v=\frac{\left(\alpha-\frac{1}{6\alpha}\right)u_1}{(\alpha-1)u_2}.$$
2. If
$$\frac{2(u_2-1)}{\alpha-1}+v+\frac{1}{v}\leq 2,$$
then deliver $x=(\alpha-1)v$;
otherwise,
if
$$\frac{2\log u_2}{\alpha-1}-\log v+v\leq 1,$$
then deliver $x=(\alpha-1)v$.
3. Go to step 1.

مثال: اكتب برنامج لتوليد 100 قيمة تسلك وفق توزيع كاما بمعلمة شكل $\alpha = 2$ ثم احسب الوسط الحسابي؟

الحل:

البرنامج:

```
alpha=2
n=100
for(i in 1:n){
  k=0
  while(k==0){
    u1=runif(1)
    u2=runif(1)
    v=((alpha-1/6*alpha)*u1)/((alpha-1)*u2)
    s=(2*(u2-1)/alpha-1)+v+1/v
    d=(2*log(u2)/alpha-1)-log(v)
    if(s<=2){
      x=(alpha-1)*v
    }else if(d<=1){
      x=(alpha-1)*v
      k=k+1
    }
  }
  print(x)
}
me=mean(x)
me
```

اما في حالة **معلمة الشكل اقل من الواحد** فتكون الخوارزمية بالشكل الاتي:

الخوارزمية :

الخطوة الاولى:- اجعل $t = 0.07 + 0.75\sqrt{1-\alpha}$ and وان

الخطوة الثانية:- الخطوة الاولى :- ولد u1 و u2 يتوزعان وفق التوزيع المنتظم U(0,1) واجعل

V=bu1

الخطوة الثالثة :- اذا كانت $v \le 1$ فان $x = tv^{\frac{1}{\alpha}}$ و اذا كانت $u2 \le \frac{2-x}{2+x}$ اطبع قيمة x

عدا ذلك اذا كانت $u2 \le e^{-x}$ اطبع x عدا ذلك اجعل

$$x = -\log\left(\frac{t(b-v)}{\alpha}\right) \text{ and } y = \frac{x}{t};$$

اذ كانت $u2(\alpha + y(1-\alpha)) \le 1$ اطبع قيمة x

عدا ذلك اذا كانت $u2 \le y^{\alpha-1}$ اطبع قيمة x

الخطوة الرابعة : اذهب الخطوة الاولى .

مثال: اكتب برنامج لتوليد 40 قيمة تسلك وفق توزيع كاما بمعلمة شكل $\alpha = 0.8$ ثم احسب الوسط الحسابي؟

الحل:

البرنامج:-

```
Untitled1*
Source on Save
alpha=0.8
t=0.07+0.75*sqrt(1-alpha)
b=1+(exp(-t)*alpha)/t
n=40
for(i in 1:n){
  h=0
  while(h==0){
    u1=runif(1)
    u2=runif(1)
    v=b*u1
    x=t*v^(1/alpha)
    if(v<=1){
      h=h+1
    }else if(u2<=(2-x)/(2+x)){
      h=h+1
    }else if(u2<=exp(-alpha)){
      h=h+1
    }else{x=-log(t*(b-v)/alpha)
    y=x/t
    }
    if(u2*(alpha+y*(1-alpha))<=1){
      h=h+1
    }else if(u2<=y^(alpha-1)){
      h=h+1
    }
  }
  print(x)
}
m=mean(x)
m
```

8-6-9 توليد بيانات تسلك وفق توزيع بيتا

يعتبر توزيع بيتا هو احد التوزيعات العشوائية المستمرة بمعلمتي $\alpha>0$ و $\beta>0$ وله دالة كثافة احتمالية بالصيغة الاتية :-

$$p(x)=\frac{1}{\mathrm{B}(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1} \quad \text{for } 0\le x\le 1,$$

حيث ان $B(\alpha,\beta)$ تمثل دالة تسمى دالة Beta

ان طرق الكفاية لتوليد بيانات تسلك وفق توزيع بيتا تختلف باختلاف قيم المعلمات لهذا التوزيع فاذا كانت المعلمات مساوية الى الواحد فمن السهل توليد قيم بالاعتماد على طريقة التحويل المعكوس والتي في هذه الحالة ستكون تساوي جذر uniform. اما في حالة ان المعلمات تاخذ القيم اقل من الواحد ففي هذه الحالة يتم توليد القيم بناء على طريقة الرفض والقبول للباحث johnk (1964) وفي حال ان معلمات التوزيع اكبر من الواحد في هذه الحالة يتم توليد القيم بناء على طريقة الباحثين schmeiser and Babu في عام 1980 . وسنتطرق الى الحالة التي تكون فيها قيم المعلمات اقل من الواحد.

خوارزمية johnk

الخطوة الاولى :- ولد u1 و u2 يتوزعان وفق التوزيع المنتظم U(0,1) واجعل
$v_1=u_1^{1/\alpha}$ و $v_2=u_2^{1/\beta}$
الخطوة الثانية :- اجعل $w=v_1+v_2$
الخطوة الثالثة :- اذا كانت w>1 اذهب الى الخطوة الاولى
الخطوة الرابعة :- اجعل $x=\frac{v_1}{w}$
الخطوة الخامسة : اطبع x

مثال: اكتب برنامج لتوليد 50 قيمة تتبع توزيع $x\sim beta(0.2,0.8)$ ثم احسب الوسط الحسابي والتباين ؟ ثم ارسم .

الحل:

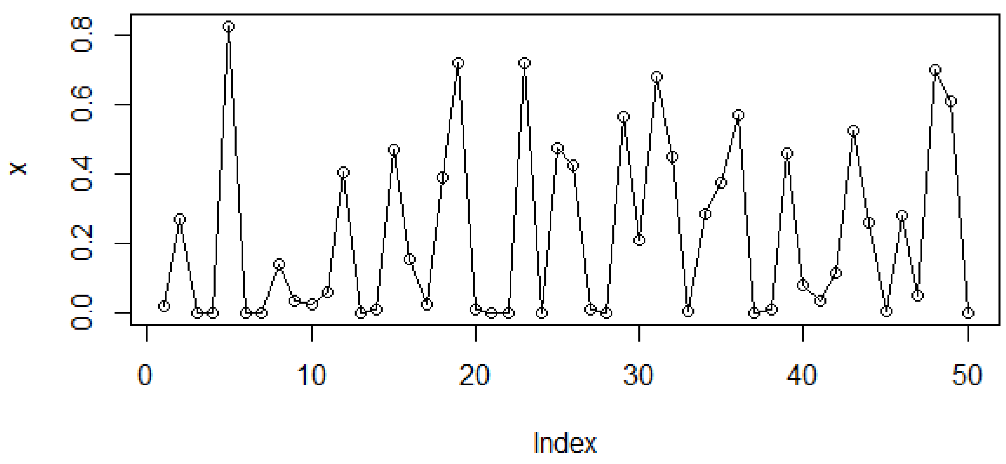
البرنامج:

```
a=0.2
b=0.8
n=50
  k=0
  while(k==0){
    u1=runif(n)
    u2=runif(n)
    v1=u1^(1/a)
    v2=u2^(1/b)
    w=v1+v2
    if(w<1){
      x=v1/w
      print(x)
    }
    k=k+1
  }
me=mean(x)
me
va=var(x)
va
plot(x)
lines(x)
```

يكون الناتج كما يأتي:-

```
> me=mean(x)
> a=0.2
> b=0.8
> n=50
> k=0
> while(k==0){
+ u1=runif(n)
+ u2=runif(n)
+ v1=u1^(1/a)
+ v2=u2^(1/b)
+ w=v1+v2
+ if(w<1){
+ x=v1/w
+ print(x)
+ }
+ k=k+1
+ }
 [1] 1.983726e-02 2.701778e-01 6.669765e-05 2.990709e-06 8.273065e-01 2.087102e-04
 [7] 3.983397e-05 1.403939e-01 3.696056e-02 2.552553e-02 5.876832e-02 4.040909e-01
[13] 2.499942e-10 8.751460e-03 4.730944e-01 1.537986e-01 2.369315e-02 3.908512e-01
[19] 7.189144e-01 9.275588e-03 5.280238e-04 4.943695e-04 7.202281e-01 7.149851e-06
[25] 4.769527e-01 4.252626e-01 1.153918e-02 4.928392e-05 5.654994e-01 2.082949e-01
[31] 6.805788e-01 4.514865e-01 4.637700e-03 2.833880e-01 3.758436e-01 5.694364e-01
[37] 9.034859e-05 1.038222e-02 4.610947e-01 8.151674e-02 3.681011e-02 1.161552e-01
[43] 5.246890e-01 2.589833e-01 2.639154e-03 2.806534e-01 4.969857e-02 7.010978e-01
[49] 6.104754e-01 5.758002e-04
```

رسم البيانات :



الفصل العاشر

تحليل الانحدار

1-10 مقدمة

ان تحليل الانحدار هو كل طريقة إحصائية يتم فيها التنبؤ بمتوسط متغير عشوائي أو عدة متغيرات عشوائية اعتمادا على قيم وقياسات متغيرات عشوائية أخرى، له عدة أنواع مثل: الانحدار الخطي، والانحدار اللوجستي، وانحدار بواسون، والتعليم المراقب والانحدار موزون الوحدة.

تحليل الانحدار هو أكثر من عملية ملائمة منحنى (أي اختيار المنحنى الأكثر ملائمة لمجموعة نقاط بيانية معطاة) فهو يتضمن ملائمة نموذج باستخدام مكونات حتمية واعتباطية. المكونات الحتمية تدعى المتنبئات أما المكونات الاعتباطية فتدعى الخطأ.

الشكل الأبسط لنموذج الانحدار يحوي متغير تابع (غير مستقل) إضافة إلى متغير مستقل (يدعى العامل، أو المتغير الخارجي، أو المتغير x)

من الأمثلة النموذجية على تحليل الانحدار: اعتماد ضغط الدم Y على عمر الشخص X، أو اعتماد الوزن لحيوانات التجربة Y على معدل التغذية اليومي .X هذا الارتباط والتابعية بين X و Y هي ما ندعوه بالانحدار أو الارتباط فنقول ارتباط Y ب.X

ويلاحظ من ذلك أن نموذج الانحدار يعتمد دائماً على علاقة السببية بمعنى ان يكون التغير في المتغير المستقل مسبب رئيسي للتغير في المتغير التابع.

ونظرية تحليل الانحدار تعتمد على النظرية الاقتصادية بين متغيرين أي أنها تفترض ثبات العوامل الأخرى.

2-10الانحدار الخطي البسيط

في تحليل الانحدار البسيط، نجد أن الباحث يهتم بدراسة أثر أحد المتغيرين ويسمى بالمتغير المستقل أو المتنبأ منه، على المتغير الثاني ويسمى بالمتغير التابع أو المتنبأ به، ومن ثم يمكن عرض نموذج الانحدار الخطي في شكل معادلة خطية من الدرجة الأولى، تعكس المتغير التابع كدالة في المتغير المستقل كما يلي:

$$\boxed{y = \beta_0 + \beta_1 x + e}$$

اذ أن:

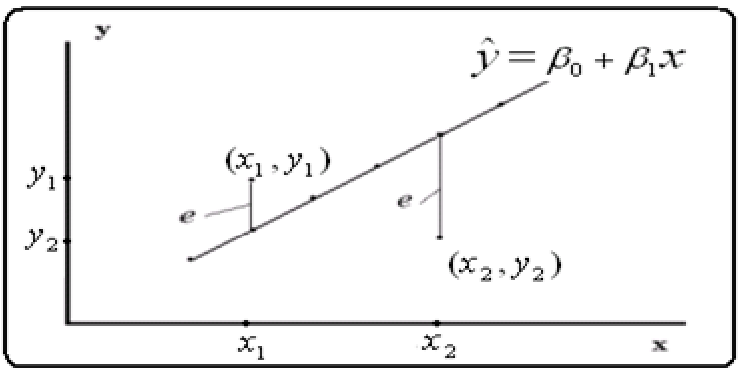
y : هو المتغير التابع (الذي يتأثر)

x: هو المتغير المستقل (الذي يؤثر)

β_0: هو الجزء المقطوع من المحور الرأسي ، وهو يعكس قيمة المتغير التابع في حالة انعدام قيمة المتغير المستقل ، أي في حالة

$\beta1$: ميل الخط المستقيم ، ويعكس مقدار التغير في إذا تغيرت بوحدة واحدة.

e: هو الخطأ العشوائي، والذي يعبر عن الفرق بين القيمة الفعلية ، والقيمة المقدرة ، ويمكن توضيح هذا الخطأ على الشكل التالي لنقط الانتشار.



10-2-1 تقدير نموذج الانحدار الخطي البسيط

يمكن تقدير معاملات الانحدار في النموذج السابق باستخدام طريقة المربعات الصغرى، وهذا التقدير هو الذي يجعل مجموع مربعات الأخطاء العشوائية أقل ما يمكن، ويحسب هذا التقدير بالمعادلة التالية:

$$\boxed{\begin{array}{l}\hat{\beta}_1 = \dfrac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2} \;, \\ \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}\end{array}}$$

حيث أن $\bar{x}$ هو الوسط الحسابي لقيم x ، $\bar{y}$ هو الوسط الحسابي لقيم y ،

وتكون القيمة المقدرة للمتغير التابع هو: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ ، ويطلق على هذا التقدير " تقدير معادلة انحدار y على x .

بعد الحصول على نتائج معادلة الانحدار يجب علينا أن نبين هل أن هذه المعاملات مقبولة من الناحية الإحصائية أي معنوية احصائياً مع التنويه بأن المعنوية تكون لكل معامل على حدة .

ولكي نحكم على معنوية معاملات الإنحدار نستعين باختبار T ومستوى الاحتمالية المقابل له وبالطبع فإن باستخدام لغة R سيتم استخراج اختبار T ومستوى الاحتمالية المقابل له .

كما سيتم الحصول على إحصائيات تستخدم لمعرفة المعنوية الإجمالية للنموذج ومنها (R) ، (R^2)

فان R هو معامل الأرتباط البسيط والذي يقيس قوة العلاقة بين متغيرين أو أكثر ، أما R^2 فهو يسمى بمعامل التحديد والذي يستخدم لمعرفة القوة التفسيرية للنموذج المقدر (المعادلة المقدرة) في حالة الإنحدار الخطي البسيط (متغير مستقل واحد مع متغير معتمد واحد) .

كذلك يمكن تقدير الانحدار الخطي البسيط باستخدام المصفوفات وكالاتي:-

$$\begin{bmatrix} Y_1 \\ Y_2 \\ . \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ . & . \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = X\beta$$

$$e = y - x\beta$$

$$Y = XB + e$$

Y: متجه عمودي أبعاده (n+1) يحتوي مشاهدات المتغير التابع .

X : مصفوفة أبعادها ($n \times k+1$) تحتوي مشاهدات المتغيرات المستقلة يحتوي عمودها الأول على قيم الواحد الصحيح ليمثل الحد الثابت .

B: متجه عمودي أبعاده ($K + 1 \times 1$) يحتوي على المعالم المطلوب تقديرها .

e: متجه عمودي أبعاده ($n \times 1$) يحتوي على الأخطاء العشوائية .

وبما أن المعادلة اعلاه هي العلاقة الحقيقية المجهولة والمراد تقديرها باستخدام الإحصاءات المتوفرة عن المتغير التابع , Y , والمتغير المستقل X_1, , فانه يستوجب تحقق الفروض الأساسية الخاصة e_i التالية :

$$e_i \sim N(0, \sigma^2 I n)$$

والذي يعني أن e_i يتوزع توزيعا طبيعيا (N) e وسطه صفري (0) وتباين (σ^2) .

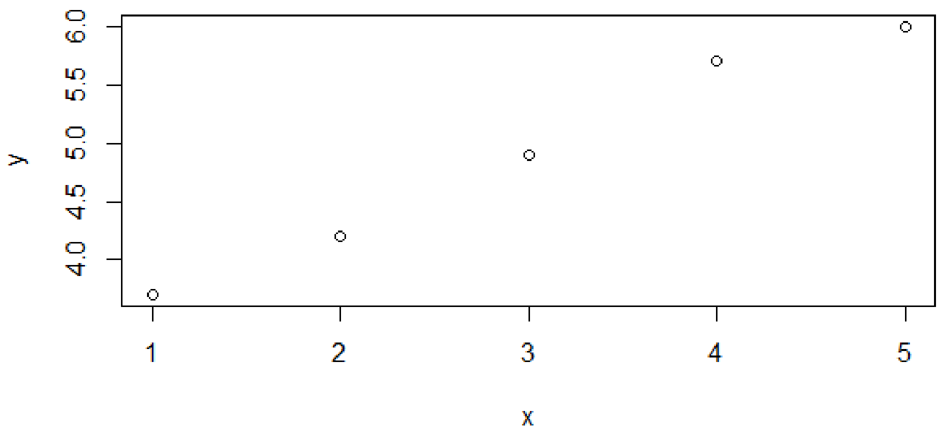
وعليه فان تقدير المعالم باستخدام المصفوفة يأخذ الصيغة التالية :

$$\hat{B} = (X'X)^{-1} X'Y$$

فاذا كانت لدينا قيم x ,y كالاتي:

x	1	2	3	4	5
y	3.70	4.20	4.90	5.70	6

سوف نرسم العلاقة الخطية بين المتغيرين باستخدام لغة R وكالاتي:



رسم يوضح العلاقة الخطية بين المتغيرين

اما تقدير معالم النموذج فيتم كما يأتي:-

Untitled1*

Source on Save

```
y=c(3.7,4.2,4.9,5.7,6)
x=cbind(rep(1,5),c(1,2,3,4,5))
beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
beta.hat
```

يكون الناتج اي تقدير معالم النموذج كالاتي:-

```
> y=c(3.7,4.2,4.9,5.7,6)
> x=cbind(rep(1,5),c(1,2,3,4,5))
> beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
> beta.hat
     [,1]
[1,] 3.07
[2,] 0.61
>
```

حيث ان قيمة $\beta_0=3.07$ وقيمة $\beta_1=0.61$

لان نقوم بتقدير متجه الاخطاء من العلاقة الاتية:-

$$e = y - x\beta$$

Untitled1*

Source on Save

```
y=c(3.7,4.2,4.9,5.7,6)
x=cbind(rep(1,5),c(1,2,3,4,5))
beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
beta.hat
error=y-x%*%beta.hat
error
z1=t(beta.hat)%*%t(x)%*%y
z2=t(y)%*%y
Rsqr=z1/z2
Rsqr;
```

يكون متجه الاخطا مساويا الى :-

```
Console ~/
> x=cbind(rep(1,5),c(1,2,3,4,5))
> beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
> beta.hat
     [,1]
[1,] 3.07
[2,] 0.61
> error=y-x%*%beta.hat
> error
              [,1]
[1,]  2.000000e-02
[2,] -9.000000e-02
[3,] -4.440892e-15
[4,]  1.900000e-01
[5,] -1.200000e-01
```

ثم نقوم بحساب كل الموشرات الخاصة بالانحدار الخطي البسيط من المعلومات التي حصلنا عليها اعلاه

لذلك سنقوم بحساب اختبار T حيث نقوم بايجاد تباين المعالم المقدرة ثم نستطيع ايجاد قيمة اختبار T وسوف نحسب احصاءة t بالنسبة للحد الثابت فقط واحصاءة t بالنسبة للميل الحدي بنفس الطريقة .

```
b0=3
b1=0.89
sigma=0.18
y=c(3.7,4.2,4.9,5.7,6)
x=cbind(rep(1,5),c(1,2,3,4,5))
beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
beta.hat
error=y-x%*%beta.hat
error
z1=t(beta.hat)%*%t(x)%*%y
z2=t(y)%*%y
Rsqr=z1/z2
Rsqr;
vab0=sigma^2*sum(x^2)/5*sum(x^2)
vab0
tb0=b0-3.07/sqrt(vab0)
tb0
```

```
[4,]  1.900000e-01
[5,] -1.200000e-01
> z1=t(beta.hat)%*%t(x)%*%y
> z2=t(y)%*%y
> Rsqr=z1/z2
> Rsqr;
          [,1]
[1,] 0.9995235
> vab0=sigma^2*sum(x^2)/5*sum(x^2)
> vab0
[1] 23.328
> tb0=b0-3.07/sqrt(vab0)
> tb0
[1] 2.364377
>
```

نلاحظ ان قيمة اختبار t بالنسبة لمعلمة الحد الثابت تساوي 2.364377 ثم نقوم بحساب القوة التفسيرية المتمثلة بمعامل التحديد وما كالاتي:

```
y=c(3.7,4.2,4.9,5.7,6)
x=cbind(rep(1,5),c(1,2,3,4,5))
beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
beta.hat
z1=t(beta.hat)%*%t(x)%*%y
z2=t(y)%*%y
Rsqr=z1/z2
Rsqr
```

يكون الناتج مساويا الى :-

```
> rm(list=ls())
> y=c(3.7,4.2,4.9,5.7,6)
> x=cbind(rep(1,5),c(1,2,3,4,5))
> beta.hat=solve(t(x)%*%x)%*%t(x)%*%y
> beta.hat
     [,1]
[1,] 3.07
[2,] 0.61
> z1=t(beta.hat)%*%t(x)%*%y
> z2=t(y)%*%y
> Rsqr=z1/z2
> Rsqr
          [,1]
[1,] 0.9995235
>
```

نلاحظ ان قيمة R^2=0.9995235 وهذا يعني ان المتغير المستقل x استطاع ان يفسر 0.99 من التغيرات الحاصلة في y والباقي يعزى الى عوامل اخرى.

3-10الانحدار الخطي المتعدد

ان نموذج الانحدار المتعدد هو عبارة عن انحدار للمتغير التابع (Y) على العديد من المتغيرات المستقلة X_1 , X_2 , ... X_K ويسمى هذا بنموذج الانحدار الخطي المتعدد , Multiple Linear Regression .

ويهدف هذا المقال إلى توضيح كيفية تقدير نموذج الانحدار الخطي المتعدد , , ثم تحديد أهم افتراضات النموذج , يضاف إلي ذلك بيان عدم وجود علاقة خطية تامة بين المتغيرات المستقلة وكيف أن المصفوفة (X'X) , تكون مصفوفة غير شاذة (Singular–Non) إذا كان محددها لا يساوي صفرا . ثم يتم بعد ذلك تقدير معلومات النموذج , تقدير التباين والتباين المشترك والانحراف المعياري لها للوصول إلى اختبار معاملات النموذج .

1-3-10 أنموذج الانحدار الخطي المتعدد :

يستند النموذج الخطي المتعدد على افتراض وجود علاقة خطية بين متغير تابع Y_i وعدد من المتغيرات المستقلة $X_1,X_2,...X_K$ وحد عشوائي U_i , ويعبر عن هذه العلاقة , بالنسبة لـn من المشاهدات وk من المتغيرات المستقلة , بالشكل آلاتي :

$$Y_i = B_0 + B_1X_{i1} + B_2X_{i1} + \dots + B_KX_{ik} + U_i \qquad \dots. (1)$$

هذه المعادلة تتضمن (k+1) من المعلومات المطلوب تقديرها علما بان الحد الأول منها (B_0) يمثل الحد الثابت , الآمر الذي يتطلب اللجوء إلى المصفوفات والمتجهات لتقدير تلك المعلمات. عليه يمكن صياغة هذه المعادلات في صورة مصفوفات وكآلاتي :

$$\begin{bmatrix} Y_1 \\ Y_2 \\ . \\ Y_n \end{bmatrix} \quad ....\ (2) \quad = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1K} \\ 1 & X_{21} & X_{22} & \cdots & X_{2K} \\ . & . & . & \cdots & . \\ 1 & X_{n1} & X_{n2} & \cdots & X_{nk} \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ . \\ B_K \end{bmatrix} + \begin{bmatrix} U_0 \\ U_1 \\ . \\ U_n \end{bmatrix}$$

وباختصار

$$Y = XB + U$$

Y: متجه عمودي أبعاده (n+1) يحتوي مشاهدات المتغير التابع

X : مصفوفة أبعادها (n × k+1) تحتوي مشاهدات المتغيرات المستقلة يحتوي عمودها الأول على قيم الواحد الصحيح ليمثل الحد الثابت .

B: متجه عمودي أبعاده (K + 1 ×1) يحتوي على المعالم المطلوب تقديرها .

U: متجه عمودي أبعاده (n ×1) يحتوي على الأخطاء العشوائية .

وبما أن المعادلة (1) هي العلاقة الحقيقية المجهولة والمراد تقديرها باستخدام الإحصاءات المتوفرة عن المتغير التابع , Y , والمتغيرات المستقلة , $X_1, X_2, .. X_K$, فانه يستوجب تحقق الفروض الأساسية الخاصة بU_i التالية :

$$U_i \sim N\ (0\ ,\ \sigma^2 I\, n\ )$$

والذي يعني أن U_i يتوزع توزيعا طبيعيا (N) متعدد المتغيرات لمتجه وسطه صفري (0) ومصفوفة تباين وتباين مشترك عددية هي (σ^2 In) .

2-3-10 طريقة المربعات الصغرى

في ضوء الفرضيات المذكورة أعلاه يمكن استخدام طريقة OLS في تقدير معلمات النموذج الخطي المتعدد , ولهذا الغرض يمكن كتابة المعادلة (1) بصيغتها التقديرية كآلاتي :

$$\hat{Y}_i = \hat{B}_0 + \hat{B}_1 X_{i1} + \hat{B}_2 X_{i2}$$

ولما كان هدفنا هو الحصول على قيم كل من $\hat{B}_0, \hat{B}_1, \hat{B}_2$ التي تجعل مجموع مربعات الانحرافات اقل ما يمكن , أي تصغير القيمة $\sum e_i^2$ (مبدا المربعات الصغرى) إلى اقل قيمة ممكنة , أي :

$$\text{Min} \rightarrow \sum_{i=1}^{n} e_i^2$$

$$\because e_i = Y_I - \hat{Y}_i$$

$$\sum e_i^2 = \sum (Y_I - \hat{Y}_i)^2$$

ومن خلال التعويض عن $\hat{Y}_i$ بما يساويها واخذ المشتقات الجزئية بالنسبة إلى $\hat{B}_2, \hat{B}_1, \hat{B}_0$ ومساواتها بالصفر نحصل على :

$$\sum e_i^2 = \sum (Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2})^2$$

$$\frac{\delta e_i^2}{\delta \hat{B}_0} = 2\sum (Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2})(-1) = 0$$

$$-2 \sum (Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2}) = 0$$

بالقسمة على (-2) وفك القوس , نحصل :

$$\sum Y_i - n\hat{B}_0 - \hat{B}_1 \sum X_{i1} - \hat{B}_2 \sum X_{i2} = 0$$

$$\sum Y_i = n\hat{B}_0 + \hat{B}_1 \sum X_{i1} + \hat{B}_2 \sum X_{i2} \qquad (3)$$

$$\frac{\delta\sum e_i^2}{\delta\hat{B}_1} = 2\sum(Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2})(-X_{i1}) = 0$$

$$-2\sum X_{i1}(Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2}) = 0$$

بالقسمة (-2) وفك القوس , نحصل :

$$\sum X_{i1}Y_i - \hat{B}_0\sum X_{i1} - \hat{B}_1 X_{i1}^2 - \hat{B}_2\sum X_{i1}x_{i2} = 0$$

$$\sum X_{i1}Y_i = \hat{B}_0\sum X_{i1} + \hat{B}_1\sum X_{i1}^2 + \hat{B}_2\sum X_{i1}X_{i2} \qquad (4)$$

$$\frac{\delta\sum e_i^2}{\delta\hat{B}_1} = 2\sum(Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2})(-X_{i2}) = 0$$

$$-2\sum X_{i2}(Y_i - \hat{B}_0 - \hat{B}_1 X_{i1} - \hat{B}_2 X_{i2}) = 0$$

بالقسمة على (-2) وفك القوس , نحصل :

$$\sum X_{i2}Y_i - \hat{B}_0\sum X_{i2} - \hat{B}_1\sum X_{i1}X_{i2} - \hat{B}_2\sum X_{i2}^2 = 0$$

$$\sum X_{i2}Y_i = \hat{B}_0\sum X_{i2} + \hat{B}_1\sum X_{i1}X_{i2} + \hat{B}_2\sum X_{i2}^2 \qquad \textbf{(5)}$$

وتمثل المعادلات (3) , (4) و (5) المعادلات الطبيعية الثلاث التي تستخدم في تقدير المعالم الثلاثة المجهولة $\hat{B}_2, \hat{B}_1, \hat{B}_0$. أن هذه المعادلات , يمكن حلها باستخدام طريقة الانحرافات.

4-10 اختبار الفرضيات لنموذج الخطي المتعدد :

يهدف هذا البحث إلى توسيع معارفنا الأساسية لنموذج الانحدار وذلك بأجراء اختبار معنوية الانحدار المتعدد والمقدر باستخدام توزيع اختبار إحصاءه F ومقارنته باختبار t ومن ثم تقييم كفاءة الأداء العام لنموذج الانحدار المتعدد R^2 ومقارنته بمعامل التحديد المقدر المعدل $\bar{R}^2$, وكذلك اختبار العلاقة بين F و R^2 من خلال جدول تحليل التباين , ANOVA , ثم علاقة R^2 بقيمة المتغير العشوائي , $\sum e_i^2$.

5-10 اختبار معنوية المعلمات (اختبار t) :

يستخدم اختبار t لتقييم معنوية تأثير المتغيرات المستقلة $x_1,x_2,\ldots x_k$ في التغير التابع y في نموذج الانحدار المتعدد يعتمد على نوعين من الفروض :

فرضية العدم H_0 $B_1 = B_2 = B_3 \ldots = B_K = 0$

الفرضية البديلة H_1 $B_1 = B_2 \# B_3 \# \ldots B_K = 0$

وبعد احتساب قيمة (t) تقارن مع قيمتها الجدولية لتحديد قبول او رفض فرضية العدم ومن ثم تقييم معنوية معلمات النموذج المقدر ، والصيغة الرياضية لهذا الاختبار يمكن بيانها كما يلي :

ا – بالنسبة

الى $\hat{B}_1$

$$t\hat{B}_1 = \frac{\hat{B}_1}{S_{\hat{B}_1}}$$

$$S_{\hat{B}_1} = \sqrt{S^2{}_{\hat{B}_1}}$$

$$S^2_{\hat{B}_1} = \mathrm{var}(\hat{B}_1) = S^2 e a_{11}$$

$$\mathrm{var}(\hat{B}) = S^2 e (x'x)^{-1}$$

$$S^2 e = \frac{e'e}{n-k-1} = \frac{Y'Y - \hat{B}'X'Y}{n-k-1} = \frac{\sum y^2 - (\hat{B}_1 \sum x_1 y + \hat{B}_2 \sum x_2 y)}{n-k-1}$$

ب – بالنسبة الى $\hat{B}_2$:

$$t_{\hat{B}_2} = \frac{\hat{B}_2}{S_{\hat{B}_2}}$$

$$S_{\hat{B}_2} = \sqrt{S^2{}_{\hat{B}_2}}$$

$$S^2{}_{\hat{B}_2} = \mathrm{var}(\hat{B}_2) = S^2 e a_{22}$$

$$S^2 e = \frac{e'e}{n-k-1}$$

6-10 معامل التحديد R^2 Multiple Coefficient of determination

ويعد مؤشر أساس في تقييم مدى معنوية العلاقة بين المتغير التابع (Y) والمتغيرات المستقلة (X_K) إذ (k ، ... ، k = 1) ، بعبارة أخرى هو مقياس يوضح نسبة مساهمة المتغيرات المستقلة في تفسير التغير الحاصل في المتغير التابع . ويمكن اشتقاقه باستخدام المصفوفات بالانحرافات كآلاتي :

$$\because\ y = x\hat{B} + e$$
$$e = y - \hat{B}$$
$$e'e = (y - x\hat{B})'(y - x\hat{B})$$
$$e'e = y'y - y'x\hat{B} - x'\hat{B}'y + \hat{B}'x'x\hat{B}$$

وبما أن التحديد الثاني الثالث قيمة واحدة كما وان كلا منها يمثل مبدلا للآخر فان :

$$\therefore\ e'e = y'y - 2\hat{B}x'y + \hat{B}'x'y + \hat{B}'x'x\hat{B}$$
$$\therefore\ \hat{B} = (x'x)^{-1}x'y$$
$$(x'x) = \hat{B} = x'y$$
$$e'e = y'y - 2\hat{B}'x'y + \hat{B}'x'y$$
$$e'e = y'y - \hat{B}'x'y$$

بذلك يمكن كتابة معادلة الانحرافات الكلية كآلاتي :

$$y'y = \hat{B}x'y - e'e$$

إذ أن :

$y'y$: تمثل الانحرافات الكلية .

$\hat{B}'x'y$: تمثل الانحرافات الموضحة من قيل خط الانحدار .

$e'e$: تمثلا الانحرافات غير الموضحة .

وبما أن معامل التحديد R^2 عبارة عن نسبة الانحرافات الموضحة من قيل خط الانحدار إلى الانحرافات الكلية ، Total variation ، فانه يمثل نسبة مجموع مربعات التغير في المتغيرات المستقلة إلى مجموع المربعات الكلية :

$$\therefore R^2 = \frac{\hat{B}x'y}{y'y} = \frac{\hat{B}'x'y}{\sum y^2}$$

$$R^2 = 1 - \frac{e'e}{y'y - n\overline{Y}^2}$$

$$R^2 = \frac{\hat{B}_1 \sum x_1 y + \hat{B}_2 \sum x_2 y}{\sum y^2}$$

أن إضافة متغيرات مستقلة جديدة إلى المعادلة يؤدي إلى رفع قيمة R^2 ، وذلك لثبات قيمة المقام وتغير قيمة البسط بمقدار $(\hat{B}xy)$ غير أن الاستمرار بإضافة المتغيرات المستقلة سيؤدي إلى انخفاض درجات الحرية

(n-k-1) ، مما يتطلب استخراج معامل التحديد المعدل أو المصحح R^{-2} وعلى النحو الآتي :

$$\overline{R}^2 = \left[(1-R^2)\frac{n-1}{n-k-1} \right]$$

7-10 اختبار إحصائية F ، Statistics–F

يستهدف هذا الاختبار معرفة مدى معنوية العلاقة الخطية بين المتغيرات المستقلة X_1 , X_2 , ...X_K على المتغير التابع Y ، وكما هو الحال في الانحدار البسيط فأنه يعتمد على نوعين من الفروض :

فرضية العدم H0 : وتنص على انعدام العلاقة بين كل متغير من المتغيرات المستقلة XK…X1,X2, وبين المتغير التابع Y ، أي :

$$H_0 : \hat{B}_1 = \hat{B}_2 = \hat{B}_K = 0$$

الفرضية البديلة H1 : وتنص على وجود علاقة معنوية بين المتغيرات المستقلة والمتغير التابع ، أي :

$$H_1 : \hat{B}_1 \# \hat{B}_2 \# ... \hat{B}_k \# 0$$

والصيغة الرياضية لهذا الاختبار هي :

$$F = \frac{R^2 lk}{1 - R^2 \ln - k - 1}$$

وبعد احتساب قيمة F تقارن مع قيمتها الجدولية بدرجة حرية (k) و (n-k-1) للبسط والمقام ولمستوى معنوية معين . فإذا كانت القيمة المحتسبة اكبر من القيمة الجدولية ترفض H_0 وتقبل H_1 أي أن العلاقة المدروسة معنوية ، وهناك على الأقل متغير مستقل واحد من المتغيرات XK ذو تأثير في Y . أما إذا كانت القيمة المحتسبة اصغر من الجدولية فان ذلك يعني قبول H_0 أي أن العلاقة الخطية المدروسة غير معنوية أي انه ليس ثمة تأثير من أي متغير من المتغيرات المستقلة على المتغير التابع .

8-10 قياس حدود الثقة :

لاحتساب حدود الثقة لاية مشاهدة (نقطة) من مشاهدات خط الانحدار للمجتمع او بعبارة اخرى لحساب القيمة المتوسطة الحقيقية ال Y عند مستوى معنوية معين للمتغير المستقل في النموذج . نفترض ان النقطة المراد تقدير حدود ثقتها هي $E(Y_0)$. ولتقدير المجال الذي يمكن ان تقع فيه قيمة $E(Y_0)$ المقابلة لتشكيلة معينة من قيم المتغيرات المستقلة (K) يجب اشتقاق متباينة القيمة $E(Y_0)$.

$$X_0 = [1 X_{01} X_{02} ... X_{0K}]$$

$$\hat{Y}_0 = [1 X_{01} X_{02} ... X_{0K}] \begin{bmatrix} \hat{B}_0 \\ \hat{B}_1 \\ . \\ . \\ \hat{B}_K \end{bmatrix}$$

$$\hat{Y}_0 = \hat{B}_0 + \hat{B}_1 X_{01} + ... + \hat{B}_K X_{0K}$$

وباختصار :

$$\hat{Y}_0 = X_0\hat{B}$$

ولغرض اشتقاق المتباينة الخاصة بتقدير فترات حدود الثقة للقيمة $E(Y_0)$ يجب اشتقاق وتباين القيمة $(\hat{Y}_0)$ وكالاتي :

لايجاد الوسط فاننا ناخذ القيمة المتوقعة ل $(\hat{Y}_0)$:

$$E(\hat{Y}_0) = E(X_0\hat{B})$$
$$E(\hat{Y}_0) = X_0E(\hat{B})$$
$$\because E(\hat{B}) = B$$
$$\therefore E(\hat{Y}_0) = X_0B$$

ولايجاد التباين :

$$\text{Var}\ (\hat{Y}_0) = E\{(\hat{Y}_0 - E(\hat{Y}_0)(\hat{Y}_0 - E(\hat{Y}_0)'\}$$

$$= E\{(\hat{Y}_0 - X_0B)(\hat{Y}_0 - X_0B)'\}$$
$$\because \hat{Y}_0 = X_0\hat{B}$$
$$\therefore \text{var}(\hat{Y}_0) = E\{(X_0\hat{B} - X_0B)'\}$$
$$= X_0\{E(\hat{B} - B)(\hat{B} - B)'\}X_0'$$
$$= \sigma^2X_0(X'X)^{-1}X_0'$$

واذا رمزنا للقيمة التقديرية لتبلين قيمة حدود الثقة $\text{var}(\hat{Y}_0)$ بالرمز $S^2(\hat{Y}_0)$ ، فان :

$$S^2(\hat{Y}_0) = S^2X_0(X'X)^{-1}X_0'$$

وعليه فان حدود الثقة للقيمة $E(Y_0)$ تكون :

$$E(Y_0) = \hat{Y}_0 \mp t_{\alpha l2}.S(\hat{Y}_0)$$
$$E(Y_0) = X_0\hat{B} \mp T_{\alpha L2}.S(\hat{Y}_0)$$

وسيتم محاكاة النموذج من خلال توليد المتغيرات التفسيرية X1,X2,X3 من التوزيع المنتظم ويتم توليد الاخطاء حسب التوزيع الطبيعي القياسي بمتوسط صفر وتباين σ^2,اما المتغير المعتمد فيتم توليده من المتغيرات المستقلة والخطأ.وسنستخدم حجم عينة 20 مفردة

ويمكن معرفة الاثر او ا لعلاقة بين المتغيرات التفسيرية والمتغير المعتمد من خلال تقدير هذه العلاقة وبالشكل (شكل العلاقة) الاتي:

$$Y = f\,(X1,\ X2\ ,\ X3) \qquad \ldots.(1)$$

والمعادلة التقديرية حسب نموذج الانحدار الخطي المتعدد تكون وفق الاتي :

$$Y = B_0 + B_1X_1 + B_2X2 + B_3X3 + u \quad \ldots\ldots\ldots\ldots\ldots\ (2)$$

حيث تمثل :

B_0 : معامل التقاطع او الحد الثابت

B1,B2,B3: تمثل معاملات دالة معادلة الانحدار الخطي المتعدد

U ، تمثل الخطا القياسي او الخطا العشوائي للنموذج المقدر

9-10 تطبيق تحليل الانحدار في R

ولتقدير النموذج السابق نستخدم طريقة المربعات الاعتيادية . لاحظ الجزء الاول هو توليد مصفوفة x وهي كالاتي:-

```
rm(list=ls())
sigma=0.34
n=20
x0=rep(1,n)
x1=runif(n,0,1)
x2=runif(n,0,1)
x3=runif(n,0,1)
x=cbind(x0,x1,x2,x3)
x
```

ستكون المصفوفة بالشكل الاتي:-

```
> x
              x1         x2          x3
 [1,] 0.77638804 0.57394382 0.930142649
 [2,] 0.91631623 0.18097728 0.472454487
 [3,] 0.43390258 0.85802301 0.527974153
 [4,] 0.93856367 0.15150463 0.459507779
 [5,] 0.88044022 0.32536312 0.509274228
 [6,] 0.86873319 0.35288281 0.187818287
 [7,] 0.36186926 0.88858303 0.196462843
 [8,] 0.96568777 0.11345540 0.979551286
 [9,] 0.78856850 0.24947052 0.360184819
[10,] 0.76837161 0.01366969 0.714204251
[11,] 0.70240103 0.37219554 0.156166797
[12,] 0.68590747 0.59773810 0.180479919
[13,] 0.14434450 0.46037159 0.749539664
[14,] 0.10195179 0.63349232 0.574376905
[15,] 0.62660243 0.39115329 0.722079695
[16,] 0.81947648 0.05891295 0.557810958
[17,] 0.40189474 0.30313359 0.348744481
[18,] 0.71290460 0.48877781 0.078772920
[19,] 0.69957139 0.03631461 0.008878461
[20,] 0.02763185 0.95073471 0.564800482
```

بعد ان تم توليد مصفوفة x لان يتم توليد المصفوفة القياسية اي مصفوفة x'x و كذلك x'y حتى نتمكن من اجراء كافة الخطوات المطلوبة مثل التقدير وحساب التباين وحساب فترة الثقة وهكذا كماياتي:-

وكما في الاتي:-

Untitled1* diff simulation.R

Source on Save

```
##########################
mean_1 = mean(x[,1])
mean_1
sd_1 = sd(x[,1])
sd_1
mean_2 <- mean(x[,2])
mean_2
sd_2 <- sd(x[,2])
sd_2
mean_3<- mean(x[,3])
mean_3
sd_3 <- sd(x[,3])
sd_3
#################################
###generate response veriable ###
#################################
y <- numeric(n)
u<- numeric(n)
u=rnorm(n,0,sigma)
for(i in 1:n){
  y[i]=b0+b1*x[i,1]+b2*x[i,2]+b3*x[i,3]+u[i]
}
y
mean_y <- mean(y)
mean_y
sd_y <- sd(y)
sd_y
#######################################
std_yy <- (y - mean_y)/sd_y
std_y<-std_yy/sqrt(n-1)
std_y
st.x.10 <- (x[,1] - mean_1)/sd_1
st.x.20 <- (x[,2] - mean_2)/sd_2
st.x.30 <- (x[,3] - mean_3)/sd_3
st.x.13<-st.x.10/sqrt(n-1)
st.x.23<-st.x.20/sqrt(n-1)
```

```
###############################################################
x.matrix <- matrix(c(st.x.13,st.x.23,st.x.33), nrow = n)
print(x.matrix)
############## Standardized matrices ############
#####
t_x_matrix <- t(x.matrix)
new_x_matrix <- t_x_matrix %*% x.matrix
print(new_x_matrix)
###############################
t_x_y <- t_x_matrix %*% std_y
print(t_x_y)
```

وسيكون الناتج كما يأتي:-

```
Console ~/
> new_x_matrix <- t_x_matrix %*% x.matrix
> print(new_x_matrix)
           [,1]        [,2]        [,3]
[1,]  1.0000000 -0.26906564 -0.10013328
[2,] -0.2690656  1.00000000  0.02065215
[3,] -0.1001333  0.02065215  1.00000000
> ################################
> t_x_y <- t_x_matrix %*% std_y
> print(t_x_y)
          [,1]
[1,] 0.3244067
[2,] 0.2186477
[3,] 0.5874422
>
```

حيث ان المصفوفة الاولى تمثل المصفوفة القياسية والمتجة يمثل x'y ولاحساب القيم التقديرية لمعالم النموذج كما يأتي:-

```
> ##########################
> ## ordinary least square ###
> ##########################
> ols <- lm(std_y~st.x.13+st.x.23+st.x.33)
> print(ols)

Call:
lm(formula = std_y ~ st.x.13 + st.x.23 + st.x.33)

Coefficients:
(Intercept)      st.x.13      st.x.23      st.x.33
 -2.448e-16    6.767e-01    2.984e-01    8.764e-02
```

حيث ان B0= -2.448e-16 وان B1=0.6767 وان B2=0.984 وان B3=0.0846

ولاجراء اختبار معنوية المعاملات المقدرة تتم حسب توزيع t لكل معلمة من معالم النموذج وكذلك معرفة القوة التفسيرية للنموذج ومعرفة معنوية النموذج اي معنوية كل المتغيرات على المتغير المعتمد وكآلاتي:-

```
Console ~/
      Min        1Q    Median        3Q       Max
-0.269107 -0.115926 -0.006112  0.060880  0.304701

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.006e-16  3.442e-02   0.000  1.00000
st.x.13      3.025e-01  1.671e-01   1.810  0.08907 .
st.x.23      4.978e-01  1.662e-01   2.996  0.00855 **
st.x.33      4.508e-01  1.549e-01   2.910  0.01022 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1539 on 16 degrees of freedom
Multiple R-squared:  0.6209,    Adjusted R-squared:  0.5498
F-statistic: 8.734 on 3 and 16 DF,  p-value: 0.00116
```

نلاحظ ان المعلمة الثانية والثالثة ظهرت معنوية وكذلك النموذج ككل ظهر ايضا معنوي من خلال إحصاءه F=8.734 وقيمة P-value =0.00116 اي اقل من 0.05 وهذا دليل على معنوية النموذج .وكذلك ان تباين البواقي يكون مساويا الى 0.1539 وان قيمة معامل التحديد مساوية الى 0.54 اي ان المتغيرات المستقلة تفسر 54% من التغيرات الحاصلة في المتغير الاستجابة او المعتمد والباقي 46% يعتمد على عوامل اخرى موجودة ضمن عنصر الخطأ العشوائي .

الفصل الحادي عشر

الرسوم البيانية

1-11 المقدمة

تعتبر الرسوم البيانية تمثيل رسومي للبيانات، حيث تمثّل البيانات بواسطة رموز، كالأشرطة في المخطط البياني الشريطي أو الخطوط في المخطط البياني الخطي أو الشرائح في المخطط البياني الدائري. يمكن أن يمثّل المخطط البياني بيانات رقمية من مجدولة، أو بيانات اقترانيه أو بعض أنواع التركيبات البيانية النوعية. يستخدم التعبير "مخطط بياني" كتمثيل رسومي للبيانات التي تحتمل عدة معاني منها مخطط بياني من نوع تخطيط أو رسم غرافيك، والتي تنظم وتمثل مجموعة بيانات رقمية أو نوعية. وغالباً ما تعرف الخرائط المزخرفة بمعلومات إضافية لأغراض محددة بالمخططات البيانية، كالمخططات البيانية البحرية أو مخططات الطيران.

وتستخدم المخططات البيانية لتسهيل فهم كميات كبيرة من البيانات والعلاقات التي تربط بينها. يمكن قراءة المخطط البياني بسرعة أكبر من قراءة البيانات الخام. وتستخدم المخططات البيانية في مجالات عديدة ويمكن رسمها يدوياً أو بواسطة الكمبيوتر باستخدام برمجيات الرسم البياني. بعض أنواع المخططات البيانية أكثر فائدة في تمثيل مجموعة معطاة من البيانات من غيرها من الأنواع. على سبيل المثال، غالباً ما يتم تمثيل النسب المئوية في مجموعات مختلفة (مثلاً، راضٍ، غير راضٍ، غير متأكد) بمخطط بياني دائري، لكن قد تكون أكثر فهماً إن مُثّلت بمخطط بياني شريطي أفقي . من ناحية أخرى، فإن رسم البيانات التي تمثل أرقاماً متغير خلال فترة زمنية (كالأرباح السنوية منذ عام 1990 إلى عام 2000) يكون أفضل ما يكون باستخدام مخطط بياني خطي.

2-11 الايعازات الخاصة بالرسوم ثنائية الابعاد

يتضمن برنامج R العديد من الايعازات الخاصة بالرسوم ثنائية الابعاد كالإيعازات الخاصة بالرسوم البيانية الخطية والاشرطة البيانية وغيرها من الانواع.

1-2-11 الايعاز plot

يعتبر الايعاز من أكثر ايعازات الرسم استخداماً في برنامج R حيث يعتبر ايعاز عام يمكن كتابة اكثر من ايعاز بداخله أي أنه يمتلك العديد من الطرائق أبسط استخدام للإيعاز plot هو عندما نمرر له متجه فيقوم برسم مخطط انتشار له والشكل العام للايعاز هو :

plot(x)

plot(x,y)

اذ ان :

Plot: يمثل الايعاز

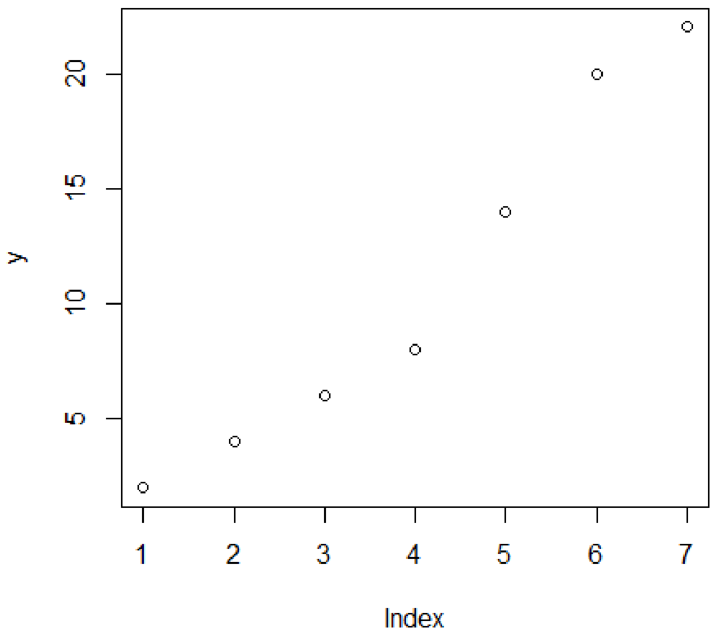
(x,y): المتجهات المطلوب رسمهما .

مثال: اذا كان لدينا المتجه y يمثل عدد الطلاب الذين يفضلون القراءة لساعات طويلة وهو كالاتي:$\boldsymbol{y = (2,4,6,8,14,20,22)}$ مثل المتجه y بشكل رسم بياني.

الحل:

```
Console ~/
> y<-c(2,4,6,8,14,20,22)
> y
[1]  2  4  6  8 14 20 22
> plot(y)
```

ويكون الرسم البياني كالاتي:



مثال: اذا كان لدينا المتجهين v,w على النحو الاتي:

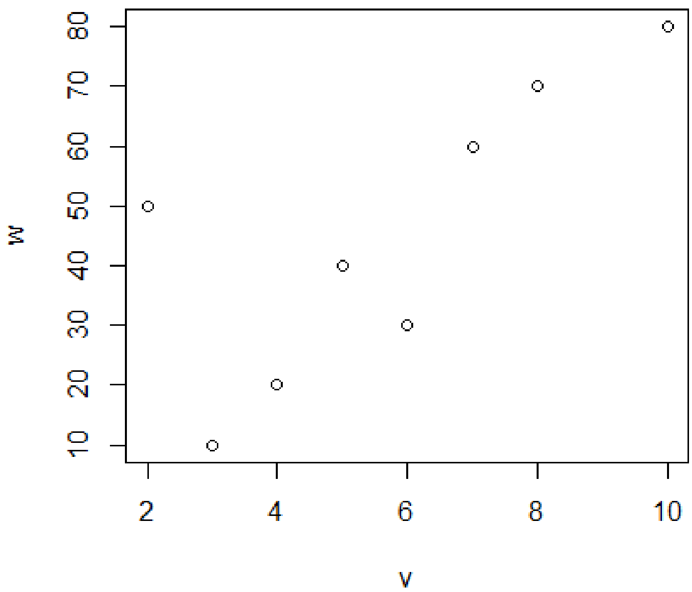
$$\boldsymbol{v} = (\mathbf{3, 4, 6, 5, 2, 7, 8, 10})\ \ , \boldsymbol{w} = (\mathbf{10, 20, 30, 40, 50, 60, 70, 80})$$

ارسم المتجهين بحيث يكون v في المحور السيني و w في المحور الصادي.

الحل:

```
Console ~/
> v<-c(3,4,6,5,2,7,8,10)
> w<-c(10,20,30,40,50,60,70,80)
> plot(v,w)
```

ويكون الرسم البياني بالشكل التالي:



ملاحظه : يجب ان يكون المتجهين المطلوب رسمهما بنفس الطول أي يمتلكان نفس العدد من القيم .

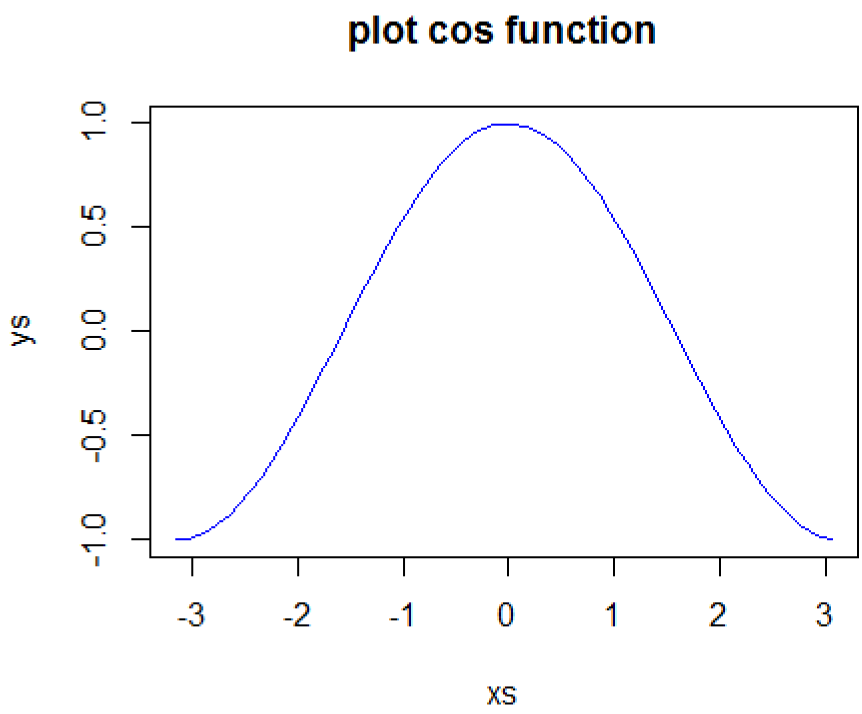
الجدول الآتي يبين أهم الايعازات التي يمكن استخدامها مع الايعاز plot .

الايعاز	الوظيفة	الخصائص
main="text"	لتعيين عنوان للرسم حيث text يمثل العنوان المطلوب	
xlab="text"	لتحديد تسمية للمحور السيني	
ylab="text"	لتحديد تسمية للمحور الصادي	
type="tp"	لتغيير نمط الرسم	من الأنماط: p للنقاط، l للمستقيم، b نقاط ومستقيمات، ...
col="cl"	لتغير لون الرسم	الألوان من: **blue,red,yellow,...**

مثال: ارسم دالة الجيب تمام $y = \cos(x)$ للفترة $-\pi \leq x \leq \pi$

الحل:

```
Console ~/
> x<-seq(-pi,pi,0.1)
> y<-cos(x)
> plot(x,y,col="blue",type="l",main="plot cos function",xla
b="xs",ylab="ys")
```



2-2-11 الايعاز lines

يستعمل هذا الايعاز لأدراج اكثر من رسم في نافذة واحدة حيث يعمل مع الايعاز plot حصراً و الشكل العام للإيعاز هو :

lines(x,y)

اذ ان :

lines :يمثل الايعاز

(x,y): المتجهات المطلوب اضافتها مع النافذة plot

مثال: اذا كان لدينا المتجه x وكان لدينا الدلة الكثافة الاحتمالية للتويع الاسي بالمعلمة $\boldsymbol{\lambda = 2, \lambda = 0.2}$ ارسم شكل التوزيع بالمتجهين بنافذة واحدة.

$$\boldsymbol{x = (0.2, 0.4, 0.6, 0.8, 1, 2, 2.6)}$$

الحل:

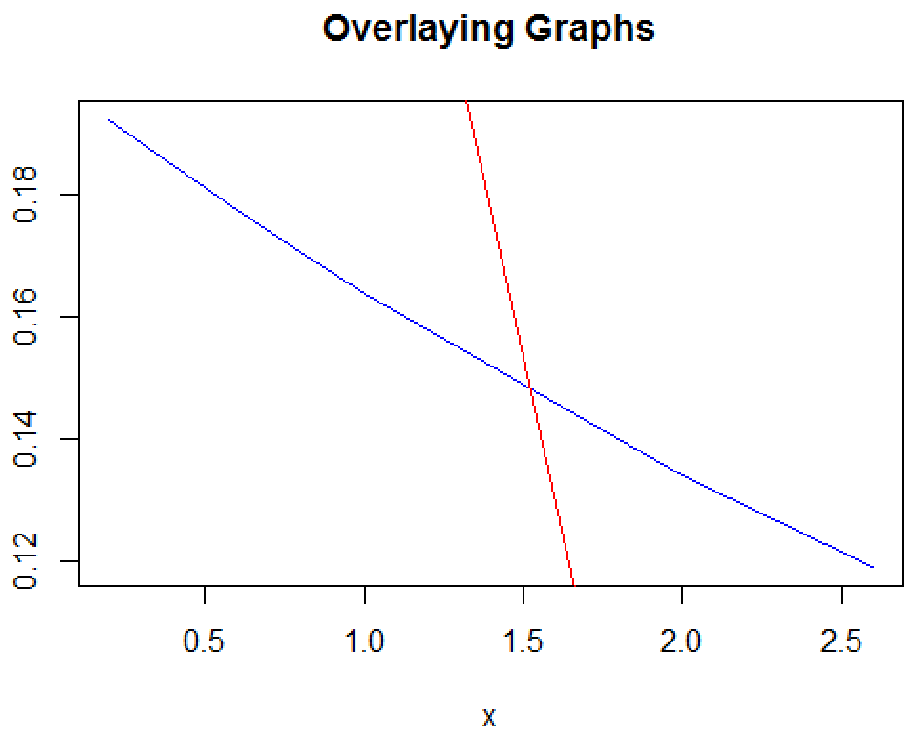
ان دالة الكثافة الاحتمالية لتوزيع الاسي تاخذ الصيغة الاتية :

$$\boldsymbol{f(x, \lambda) = \lambda e^{-\lambda x} \quad , \qquad x, \lambda > 0}$$

لذلك ستكون الخطوات كما ياتي:

```
Console ~/
> x<-c(0.2,0.4,0.6,0.8,1,2,2.6)
> lembda1<-0.2
> fx1<-lembda1*exp(-lembda1*x)
> lembda2<-2
> fx2<-lembda2*exp(-lembda2*x)
> plot(x,fx1,main="Overlaying Graphs",ylab="",type="l",col=
"blue")
> lines(x,fx2, col="red")
> |
```

ويكون الرسم البياني كالاتي:



3-2-11 الايعاز legend

يستعمل هذا الايعاز لاضافة دليل المخطط أي توضيح حالات الرسوم المدرجة بنافذة واحدة ويدرج الدليل اما في اعلى يسار الرسم او على يمين او اسف يمين او يسار والشكل العام للايعاز هو :

```
legend('topleft', c('first','second'), col=c('red','blue'), pch=c('*','o'))
```

اذ ان :

legend : يمثل الايعاز

topleft:يمثل موقع دليل المخطط اعلى يسار المخطط او يكون top right .

c('first','second'): يمثل اسم الرسم او الخط داخل الدليل

col=c('red','blue') : تمثل الوان الرسم او الخط داخل مربع الدليل

pch=c('*','o'): يمثل نوع الخط او الرسم داخل مربع الدليل .

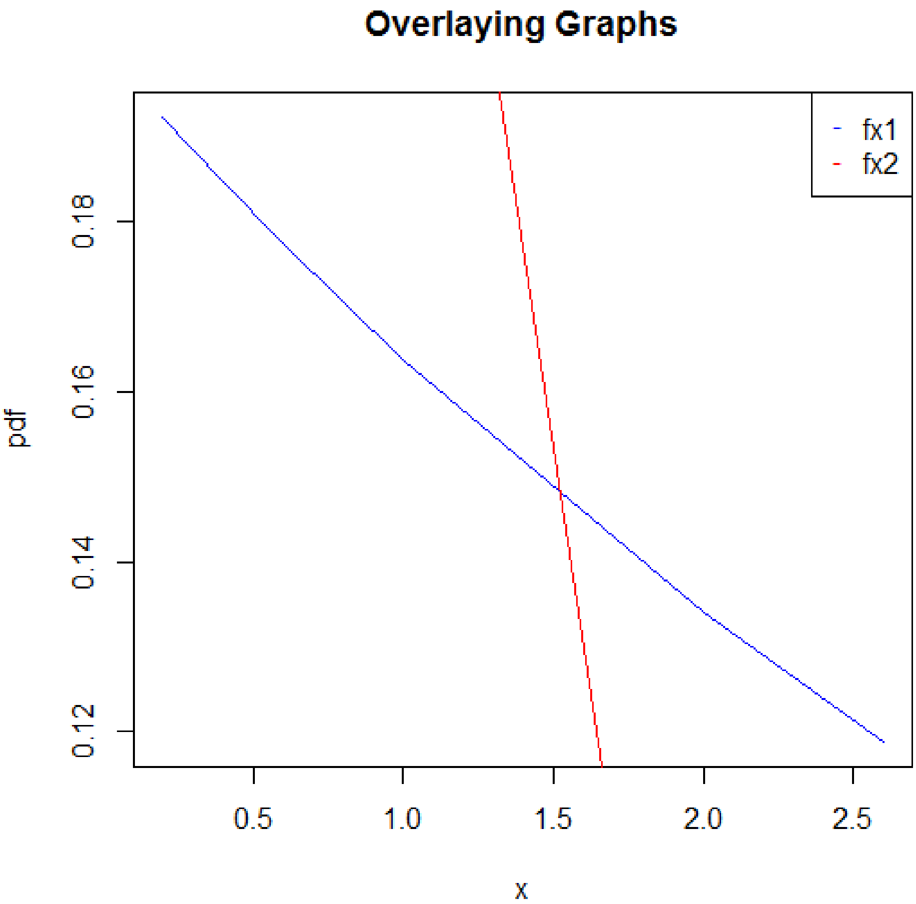
مثال: للمثال السابق اضف دليل المخطط .

الحل:

R Console

```
> rm(list = ls())
> x<-c(0.2,0.4,0.6,0.8,1,2,2.6)
> lembda1<-0.2
> fx1<-lembda1*exp(-lembda1*x)
> lembda2<-2
> fx2<-lembda2*exp(-lembda2*x)
> plot(x,fx1,main="Overlaying Graphs",ylab="pdf",type="l",col="blue")
> lines(x,fx2, col="red")
> legend("topright",c("fx1","fx2"),col=c("blue","red"),pch=c('-','-'))
> |
```

وسيكون الرسم بالشكل التالي:



4-2-11 الايعاز barplot

يمكن انشاء رسم بياني على شكل اشرطة مستطيلة باستعمال الايعاز barplot والشكل العام للايعاز هو :

barplot(f,xlab=”xlab”,ylab=”ylab”,main=”Title”,names.arg= names)

اذ ان:

barplot: الايعاز

f: المتجه المطلوب رسمه (التكرارات)

xlab=”xlab”,ylab=”ylab”: يمثلان عنوان المحور السيني والصادي

main=”Title”: لوضع عنوان رئيسي للرسم

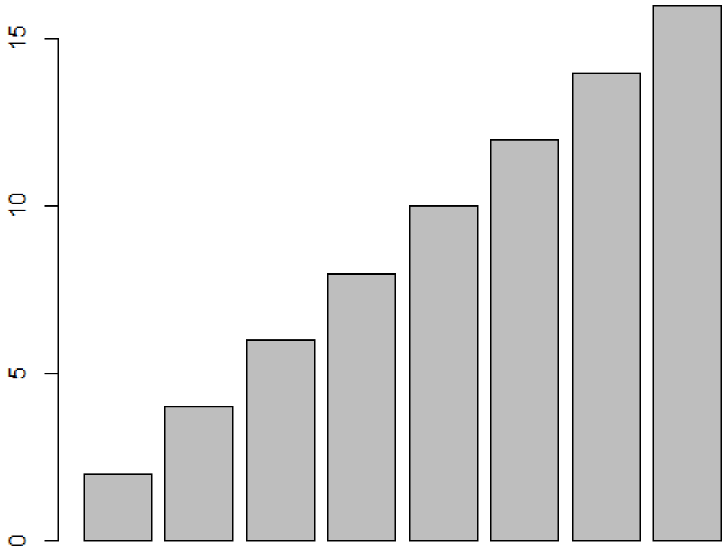
names.arg=names : لوضع تسمية توضيحية لما تمثله التكرارات f على المحور x على شكل متجه names .

مثال: ارسم قيم المتجه $\boldsymbol{D} = (\mathbf{2, 4, 6, 8, 10, 12, 14, 16})$ على شكل اشرطة مستطيلة.

الحل:

```
Console ~/
> D<-c(2,4,6,8,10,12,14,16)
> D
[1]  2  4  6  8 10 12 14 16
> barplot(D)
```

ويكون الرسم بالشكل التالي:



مثال: اذا كانت لدينا دالة الكثافة الاحتمالية للتوزيع الاسي بالمعلمة $\lambda = 2$ كالاتي:

$$f(x, \lambda) = \lambda e^{-\lambda x} \quad , \qquad x, \lambda > 0$$

وكان لدينا المتجه x ياخذ القيم الاتية :

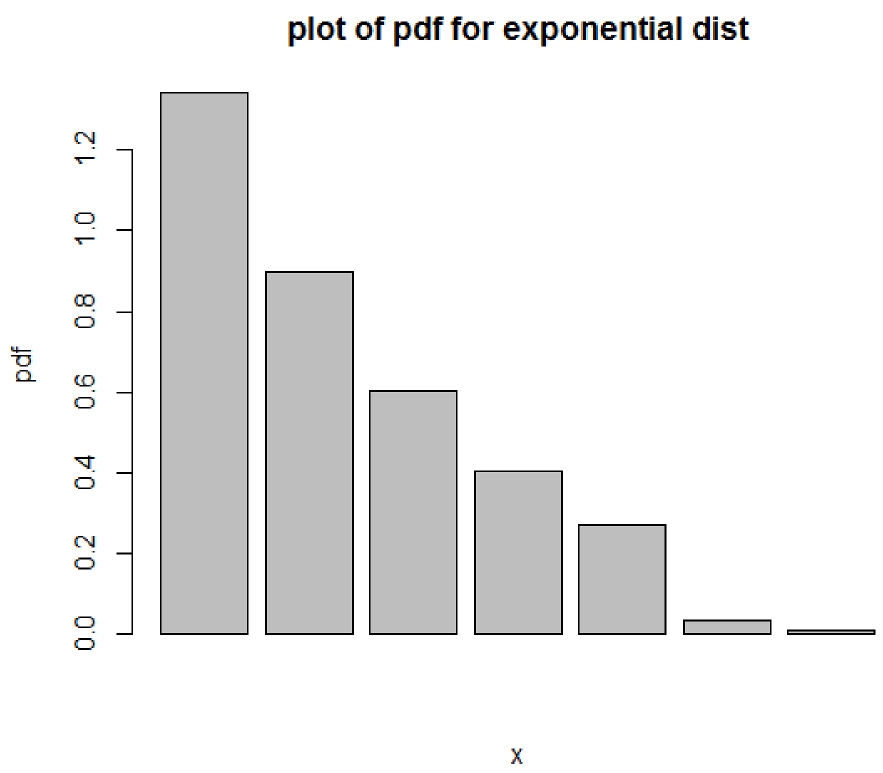
$$x = (0.2, 0.4, 0.6, 0.8, 1, 2, 2.6)$$

ارسم الدالة بشكلة اشرطة مستطيلة .

الحل:

```
Console ~/
> rm(list = ls())
> x<-c(0.2,0.4,0.6,0.8,1,2,2.6)
> lembda2<-2
> fx<-lembda2*exp(-lembda2*x)
> barplot(fx,xlab = 'x',ylab = 'pdf',main = 'plot of pdf fo
r exponential dist')
```

اما الرسم فيكون كالاتي:



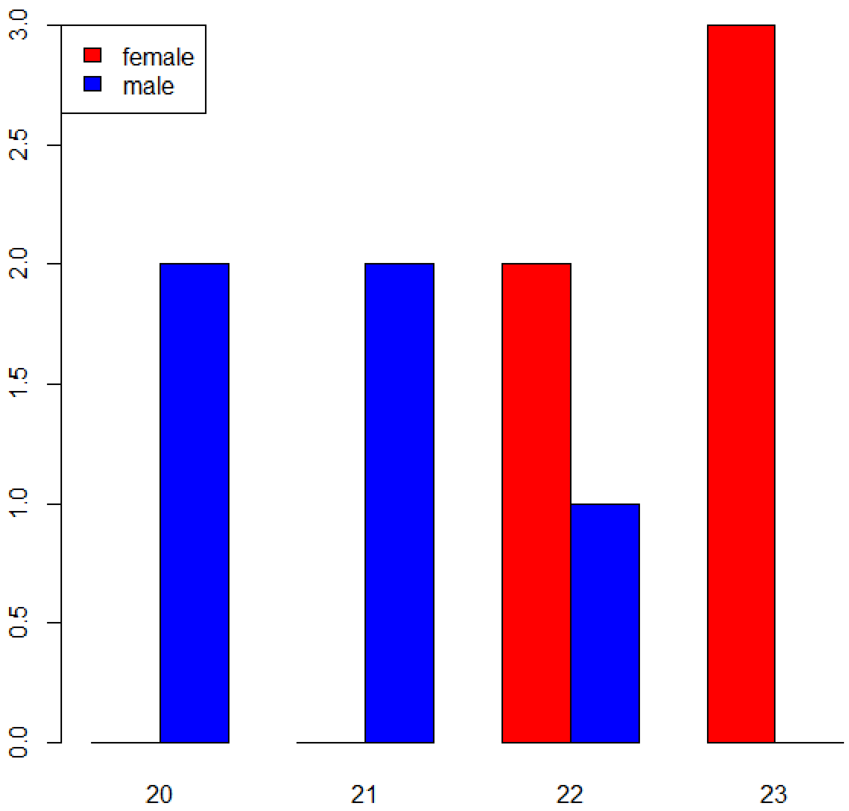
ملاحظة: يمكننا رسم مخطط أعمدة مزدوج وذلك بشرط أن تكون بياناتنا على شكل جدول *table* .

مثال: اذا كان لدينا اعمار افتراضية لكلا الجنسين ذكور واناث ارسم الاعمار حسب الجنس .

الحل:

```
Console ~/
> data<-data.frame("gender"=c(rep("m",5),rep("f",5)),"age"=
c(rep(20:21,2),rep(22:23,3)))
> dt<-table(data)
> dt
      age
gender 20 21 22 23
     f  0  0  2  3
     m  2  2  1  0
> dev.new()
NULL
> barplot(dt,beside=TRUE,col=c("red","blue"))
> legend("topleft",c("female","male"),fill=c("red","blue"))
```

اما الرسم يكون بالشكل الاتي:



5-2-11 الايعاز hist

يمكن انشاء رسم بياني على شكل مدرج تكراري باستعمال الايعاز hist والشكل العام للايعاز هو :

hist(x, breaks=k-1)

اذ ان :

hist: الايعاز

X: المتجه المطلوب رسمه (التكرارات)

Breaks: يمثل عدد النقاط التي نريد استخدامها لتجزئة البيانات، وهو أيضاً يمثل عدد الفئات التي نريد أن نقسم البيانات لها ناقصاً واحد

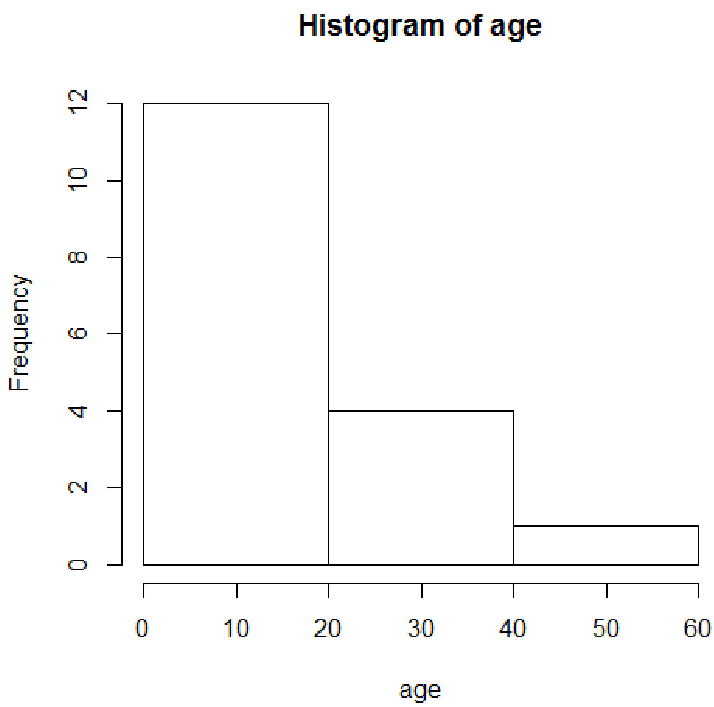
مثال: اذا كان لدينا المتجه age يمثل اعمار عينة من الطلاب في مرحلة دراسية ما

$$age = (10, 7, 18, 10, 12, 19, 18, 20, 3, 14, 19, 55, 25, 31, 26, 11, 34)$$

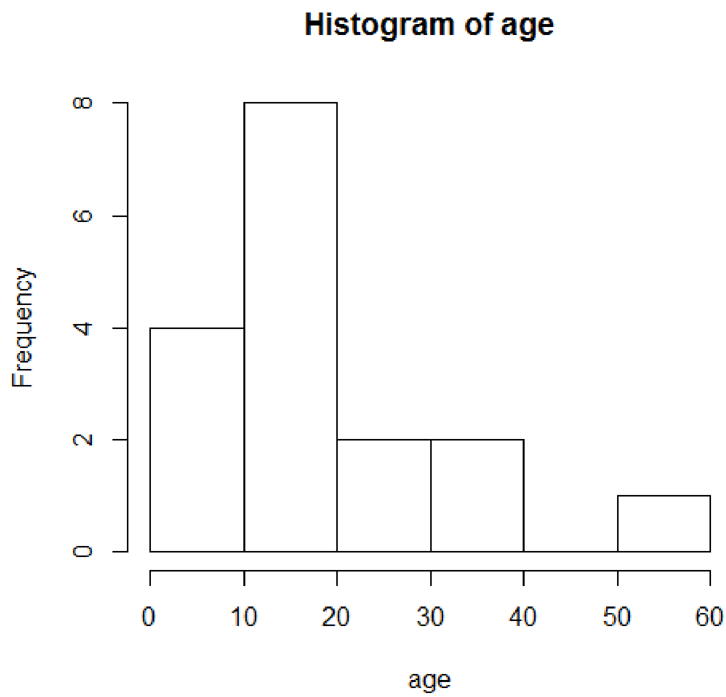
مثل الاعمار بشكل مدرج تكراري.

الحل:

```
Console ~/
> age <- c(10,7,18,10,12,19,18,20,3,14,19,55,25,31,26,11,34
)
> age
 [1] 10  7 18 10 12 19 18 20  3 14 19 55 25 31 26 11 34
> hist(age,breaks=3)
> hist(age,breaks=5)
```



اما في حالة $breaks=5$ يكون الرسم كالاتي:



6-2-11 الايعاز pie

يستعمل هذا الايعاز لأنشاء رسم بياني على شكل دائرة و الشكل العام للايعاز هو :

pie(f , main="Title", labels=names)

pie: الايعاز

main="Title": لوضع عنوان رئيسي للرسم

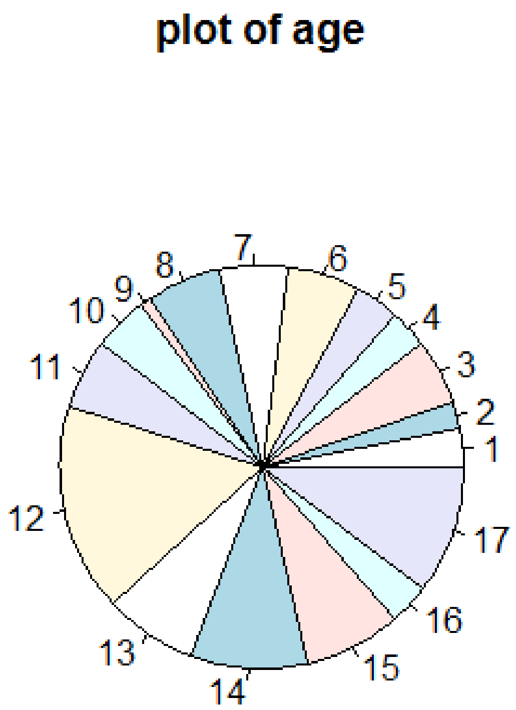
labels=names : أسماء ووصف التكرارات f

مثال: لبيانات الاعمار في المثال السابق ارسم الاعمار بشكل دائري.

الحل:

```
Console ~/
> age=c(10,7,18,10,12,19,18,20,3,14,19,55,25,31,26,11,34)
> pie(age,main = 'plot of age')
```

بذلك يكون الرسم كالاتي:



7-2-11 الايعاز boxplot

يستعمل هذا الايعاز لأنشاء رسم مخطط الصندوق لتمثيل متجه البيانات الكمية x

والشكل العام للايعاز هو :

boxplot(x, ylab="ylab" , main="Title")

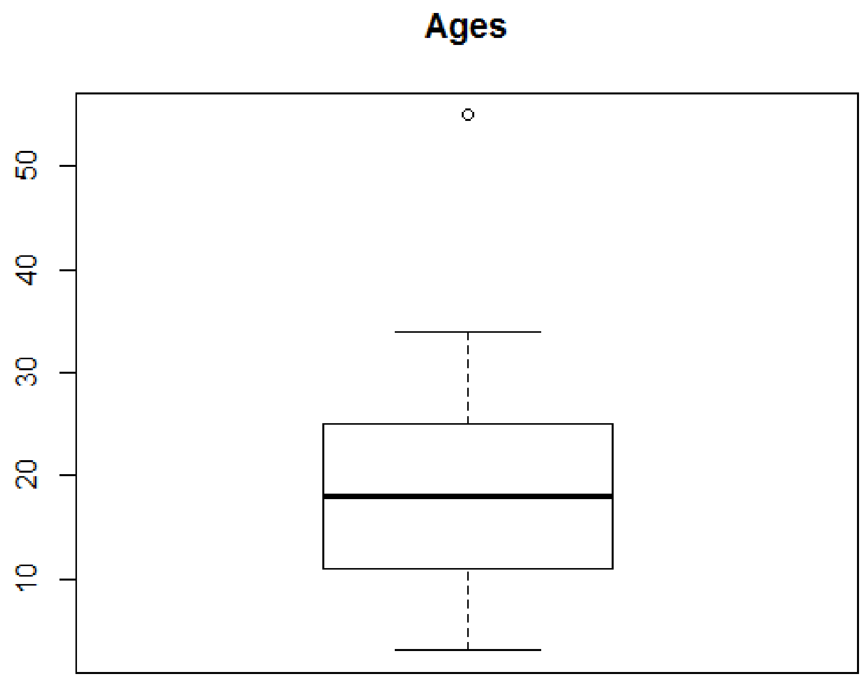
تعريفات الايعاز اصبحت معروفة للمستخدم

مثال: لبيانات العمر في المثال السابق ارسم مخطط الصندوق .

الحل:

```
Console ~/
> age=c(10,7,18,10,12,19,18,20,3,14,19,55,25,31,26,11,34)
> boxplot(age,main="Ages")
```

وبذلك يكون المخطط الصندوقي كالاتي:



3-11 التحكم بتنسيقات الرسوم البيانية

يمكن التحكم بمظهر الرسم العام في برنامج *R* من حيث نوع الخط ونوع لون الخط ونوع النقطة مع الايعاز plot والايعاز lines ولذلك لتميز وتوضيح الرسم البياني للمستخدم .

1-3-11 الالوان المتاحة

يتيح لنا برنامج R العديد من الالوان التي يمكن تطبيقها الى الرسوم البيانية من خلال الايعاز plot يتم تحديد اللون من خلال الايعاز col وهذه الالوان كما موضحة بالجدول الاتي:

استعماله في الايعاز col=c('Red')	اللون
Blue	الازرق
Green	الاخضر
Red	الاحمر
Cyan	السمائي
Magenta	الارجواني
Yellow	الاصفر
Black	الاسود

11-3-2 نوع الخط

يتيح لنا برنامج R العديد من انواع الخطوط الخاصة بالرسم البياني التي يمكن تطبيقها من خلال الايعاز plot والايعاز lines حيث يتم تحديد نوع الخط من خلال الايعاز **type** وهذه انواع الخطوط كما موضحة بالجدول الاتي:

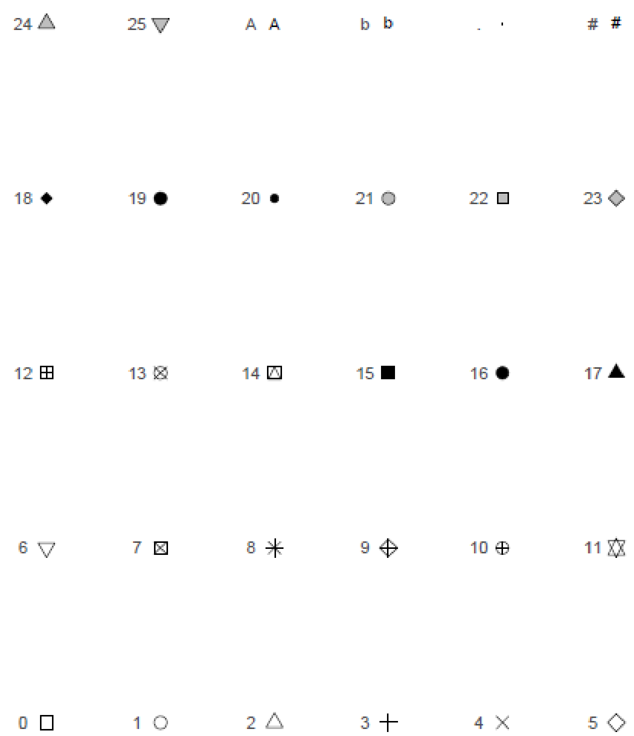
استعماله في الايعاز type=c('l')	نوع الخط
p	النقاط
l	الخطوط
b	خط ونقطة
c	خطوط متقطعة او (شارحتين)
o	خط مع تحديد نقاط التقاطع مع المحورين
h	خطوط بشكل مدرج تكراري
s	خطوات

3-3-11 نوع النقطة

يتيح لنا برنامج R العديد من انواع النقاط الخاصة بالرسم البياني التي يمكن تطبيقها من خلال الايعاز plot والايعاز lines حيث يتم تحديد نوع النقطة من خلال الايعاز **pch** وهذه انواع النقاط هي كما موضحة بالجدول الاتي:

نوع النقطة	**استعماله في الايعاز pch=c('a')**
الحرف مثلا الحرف *a*	*a*
مربع	0
دائرة	1
مثلث	2
علامة الجمع (+)	3
علامة الضرب (×)	4
معين	5

والشكل الاتي يوضع نوع النقاط ورمز كل نوع.



مثال: اذا كان لدينا المتجه w ياخذ القيم التالية :

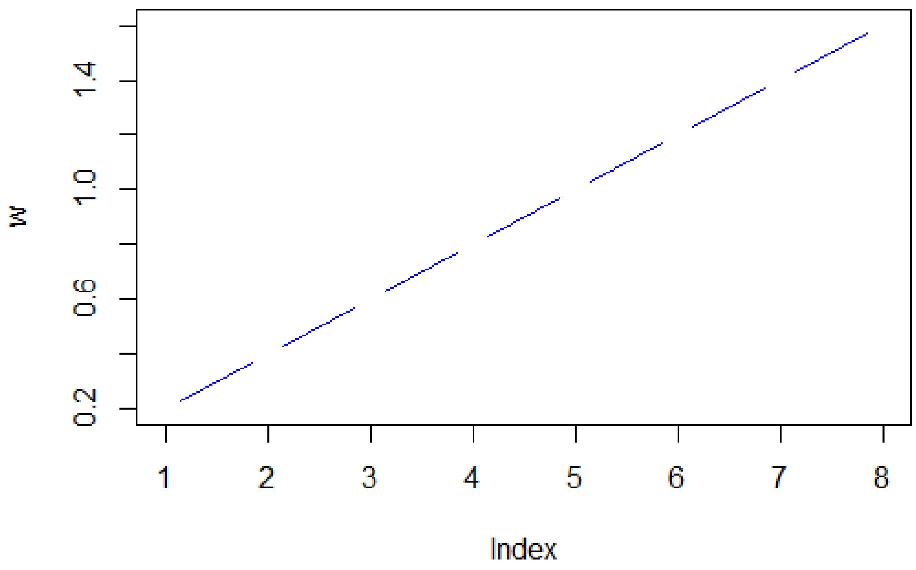
$$w = (0.2,0.4,0.6,0.8,1,1.2,1.4,1.6)$$

ارسم قيم المتجه بحيث يكون نوع اللون ازرق ونوع الخط خطوط متقطعة ؟

الحل:

```
Console ~/
> w<-c(0.2,0.4,0.6,0.8,1,1.2,1.4,1.6)
> w
[1] 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6
> plot(x,col=c('blue'),type = c('c'))
> plot(w,col=c('blue'),type = c('c'))
>
```

وتكون نتيجة الرسم كما يأتي:



4-11 الايعاز persp

تشبه الرسوم البيانية السطحية تلك الرسوم البيانية عدا انها تعبر عن المساحات الواقعة , وذلك باستعمال الايعاز **persp** والشكل العام للايعاز هو :

persp(x, y, z)

persp: الايعاز

x, y,z : المتجهات المطلوب رسمها

مثال: اذا كان لدينا المتجه x بالفترة [0,10] طوله 100 وحدة والمتجه y بالفترة [2,8] طوله 90 وحدة وان الدالة z تاخذ الصيغة الاتية :

$$z(x,y) = \sqrt{y}\,\sin(x+y)$$

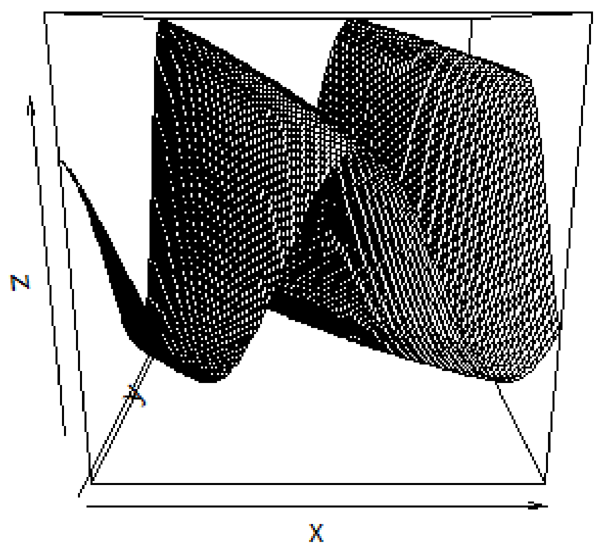
ارسم الدالة z بشكل رسم بياني سطحي :

الحل:

```
x = seq(0,10,len=100)
y = seq(2,8,len=90)
f = function(x,y)
  return(sin(x+y)*sqrt(y))
z = outer(x,y,f)
persp(x,y,z)
```

للتوضيح تم تكوين دالة f التي تمثل الدالة z وباستعمال الايعاز outer الذي يمثل الضرب الخارجي للمتجهات أي ايجاد قيمة الدالة z ثم يتم تعويضها بالايعاز وعند تشغيل البرنامج يكون الناتج كالاتي:

```
> x = seq(0,10,len=100)
> y = seq(2,8,len=90)
> f = function(x,y)
+ return(sin(x+y)*sqrt(y))
> z = outer(x,y,f)
> persp(x,y,z)
```



مثال: اذا كان لدينا المتجه x بالفترة [0,10] والمتجه y بالفترة [2,8] وان الدالة z تاخذ الصيغة الاتية :

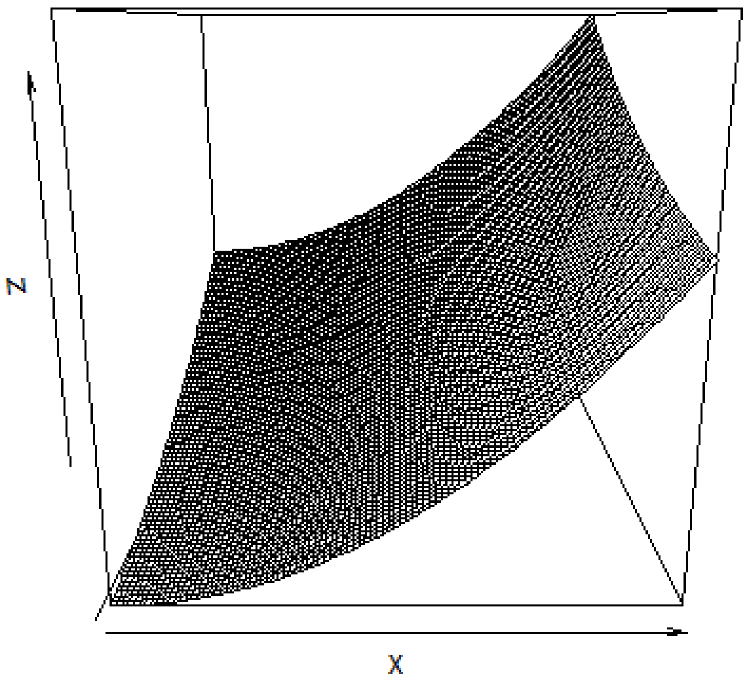
$$Z(x,y) = x^2 + y^2$$

ارسم الدالة z بشكل رسم بياني سطحي.

الحل:

```
x = seq(0,10,len=100)
y = seq(2,8,len=90)
f = function(x,y)
  return(x^2+y^2)
z = outer(x,y,f)
persp(x,y,z)
```

وعند تنفيذ البرنامج يكون الناتج كالاتي:



11-5 الايعاز **par**

يستعمل هذا الايعاز لانشاء عدة مخططات متجاورة في نفس النافذة (Subplots) من خلال تقسيم النافذة إلى m صف و n عمود ثم رسم مخطط محدد في كل ساحة عمل جزئية والشكل العام للايعاز هو :

par(mfrow=c(m,n))

اذ ان :

Par: الايعاز

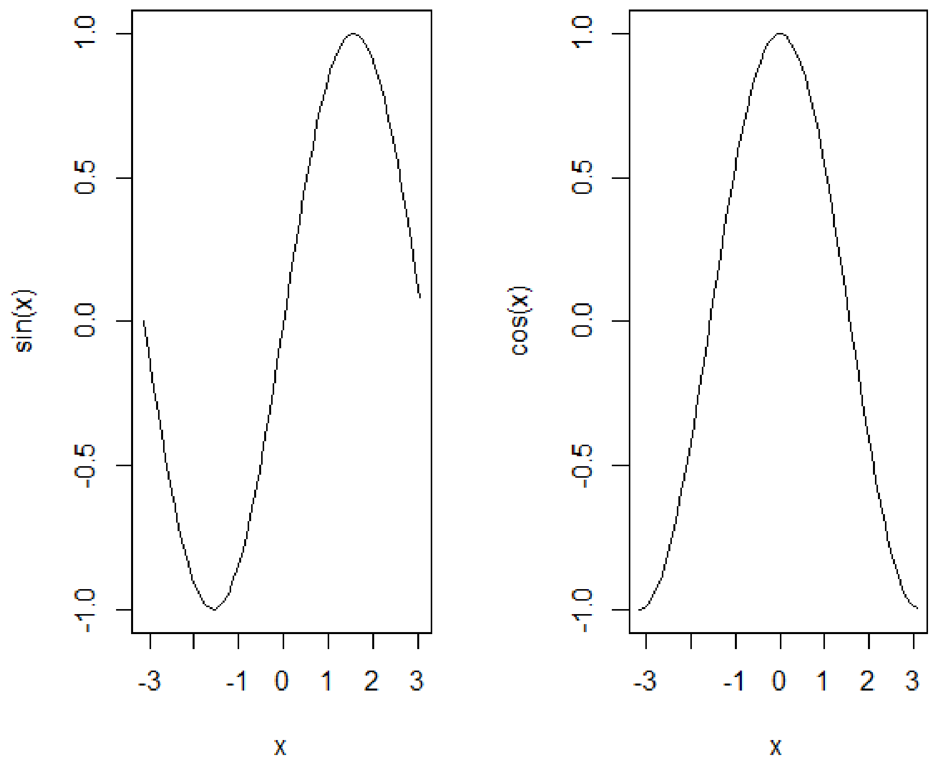
mfrow=c(m,n) : يمثل تقسيم النافذة إلى m صف و n عمود

مثال: ارسم الدالتين $\sin(x)$, $\cos(x)$ على الفترة $[-\pi,\pi]$ بنافذة واحدة .

الحل:

```
x<-seq(-pi,pi,0.1)
par(mfrow=c(1,2))
plot(x,sin(x),type="l")
plot(x,cos(x),type="l")
```

وعند تنفيذ البرنامج يكون الناتج كالاتي:


sin(x)
1.0
0.5
0.0
-0.5
-1.0
-3
-1
0
1
2
3
x
cos(x)
1.0
0.5
0.0
-0.5
-1.0
-3
-1
0
1
2
3
x

الفصل الثاني عشر

التوزيعات الاحتمالية

1-12 المقدمة

في كثير من الأحيان نرغب في التعامل مع قيم عددية مرتبطة بنقاط العينة للتجربة العشوائية بدلا من التعامل مع نقاط العينة نفسها إذ أن نقاط العينة أو النتائج الممكنة للتجربة العشوائية تكون في بعض الأحيان عبارة عن صفات أو مسميات يصعب التعامل معها رياضيا. وفي هذه الحالة فإننا نقوم بتحويل هذه القيم الوصفية إلى قيم عددية حقيقية تسمى قيم المتغير العشوائي . إن الآلة المستخدمة لتحويل عناصر فضاء العينة للتجربة العشوائية إلى قيم عددية حقيقية هي ما يسمى بالمتغير العشوائية . إذن فالمتغيرات العشوائية تستخدم للتعبير عن نتائج التجربة العشوائية وعن الحوادث بقيم عددي ة بدلا من مسميات أو صفات . فعلى سبيل المثال قد نكون مهتمين فقط بعدد الصورة الظاهرة على الوجه العلوي عند رمي قطعة عملة عشر مرات متتالية بغض النظر عن التفصيلات الأخر ى . إن عدد الصور في هذه الحالة عبارة عن متغير عشوائي تتغير قيمته بتغير نتيجة التجربة العشوائية . وهناك عدة أنواع للمتغيرات العشوائية نذكر منها نوعين هما متغيرات عشوائية منفصلة أو متقطعة و متغيرات عشوائية متصلة أو مستمرة لذلك يمكن تعريف المتغير العشوائي هو دالة حقيقية معرفة على فضاء العينة s وان لكل نوع من المتغيرات العشوائية توزيعات مختلفة كلا حسب صفاته وهذه التوزيعات هي التوزيعات الاحتمالية المتقطعة والتوزيعات الاحتمالية المستمرة .

2-12 التوزيعات الاحتمالية

2-12 -1 توزيع ذي الحدين Binomial Distribution.

إذا كانت هناك تجربة عشوائية لها نتيجتان فقط هما ظهور حدث معين نجاح او عدم ظهور فشل مثل نجاح الطالب او فشلة، المصباح الكهربائي جيد او تالف، وصول طائرة في موعدها او عدم وصولها، ظهور الصورة عند القاء قطعة نقود او عدم ظهورها.

- و إذا أجريت هذه التجربة n من المرات

واذا افترضنا أيضا أن احتمال نجاح هذه التجربة هو p . و احتمال فشلها هو q =1-p حيث p + q = 1

- نفرض أن x هو التغير العشوائي المعرف على هذه التجربة و يرمز الى عدد مرات النجاح لهذه التجربة.

تعطى دالة كثافة الاحتمال للمتغير العشوائي X و التي نرمز لها بالرمز f (x) بالمعادلة

$$p\left(X=x\right)= f(x)=\begin{cases} \binom{n}{x} P^{x}(1-p)^{n-x} & x=0,1,2,....,n \\ 0 & else\ where. \end{cases}$$

حيث n عدد صحيح موجب، $0 \langle p \langle 1$

تحت هذه الشروط واضح أن $f(x) \geq 0$

يمكن التعبير ان دالة الكثافة الاحتمالية لهذا التوزيع في لغة البرمجة R بالايعاز dbinm والشكل العام للايعاز هو :

dbinom(x,n,p)

اذ ان :

الايعاز : dbinom

x: المتغير العشوائي

n,p: معلمات التوزيع التي تمثل عدد المحاولات واحتمال النجاح على التوالي.

واما دالة التوزيع التجميعية فيمكن التعبير عنها باستعمال الايعاز pbinom والشكل العام للايعاز هو :

pbinom(x,n,p)

مثال: القيت ستة زهرات نرد . نفرض اننا نراقب ظهور الوجه الذى يحمل الر قم (3) . اوجد دالة كثافة الاحتمال الخاصة بالمتغير (X) وهو يمثل ظهور الرقم (3) .

الحل :

$n = 6$ احتمال النجاح ثابت $p = \frac{1}{6}$

احتمال الفشل ثابت $= q = (1-p) = \frac{5}{6}$

تكون قيم دالة كثافة الاحتمال باستعمال الايعاز كما يأتي :

```
Console ~/
> x<-c(0,1,2,3,4,5,6)
> x
[1] 0 1 2 3 4 5 6
> n<-6
> p<-1/6
> fx<-dbinom(x,n,p)
> fx
[1] 3.348980e-01 4.018776e-01 2.009388e-01 5.358368e-02
[5] 8.037551e-03 6.430041e-04 2.143347e-05
```

ملاحظة هامة :

اذا كان عدد المحاولات يســـاوى n = 1 واحد فأن توزيع ذي الحدين يأخذ الشكل الآتي:-

$$f(x) = \begin{cases} \binom{1}{x} p^{x} \ (1-p)^{1-x} \ , & x = 0,1 \\ 0 & , \end{cases}$$

$$f(x) = \begin{cases} p^{x} \ (1-p)^{1-x} \ , & x = 0,1 \\ 0 & , \end{cases}$$

و يطلق علية اسم توزيع برنولي **Bernoulli Distribution** نسبة الى مكتشـــــفة الرياضي الســـويسري .

مثال : عند الغاء عملة مرة واحدة فأن دالة كثافة الاحتمال لظهور الوجه هي:

$$f(x) = \begin{cases} \left(\frac{1}{2}\right)^{x} \left(\frac{1}{2}\right)^{1-x} \ , & x = 0,1 \\ 0 & , \end{cases}$$

و معنى هذا أن التوزيع المذكور يعتبر حالة خاصة من توزيع ذي الحدين عندما تكون $n = 1$ وللتعبير عنها باستعمال الايعاز كما يأتي:

```
Console ~/
> x<-c(0,1)
> n<-1
> p<-0.5
> fx<-dbinom(x,n,p)
> fx
[1] 0.5 0.5
```

وذلك يكون احتمال النجاح 0.5 واحتمال الفشل 0.5 .

2-2-12 توزيع بواسون Poisson Distribution

في الحياة العملية أحيانا ما نقابل بعض الظواهر التـي ينطبق عليها شروط توزيع ذي الحدين و لكن هذه الحوادث تكون نـادرة الوقوع و هذا يعنى أن احتمال النجاح يكون صغير جدا يقترب من الصفر و علية فأنه يمكن القول أن $np=\lambda$ حيث λ هي مقدار ثابت و بذلك يكون احتمال الفشل كبير أي أنه يقترب من الواحد. و لكى نراقب بعض حالات النجاح فأننا سنجد أن n سوف تكون كبيرة جدا فمثلا لو اردنا حساب احتمال خروج القطار من على سكته " القضبان" فأننا سنقوم بمراقبة القطارات او عدد كبير جدا منها و نحسب عدد مرات خروج القطار من على سكته أي حالات النجاح (التي حققت فيها الحادثة) حتى نستطيع أن نحسب الاحتمال.

و بذلك تكون شروط هذا التوزيع كالآتي:ـ

1ـ أن تكون احتمال النجاح ثابت و كذلك احتمال الفشل في كل محاولة و يرمز لهما بالرمز q, p على التوالي.

2ـ أن يكون احتمال النجاح صغيرا و يقترب من الصفر و احتمال الفشل يقترب من الواحد الصحيح.

3ـ أن تكون عدد المحاولات كبيرا جدا حيث أن $np=\lambda$ = مقدار ثابت.

و يعتبر توزيع بواسون من التوزيعات الاحتمالية المتقطعة و سمى هذا التوزيع بهذا الاسم نسبة الى أحد مكتشـــفه و هو بواسون و يعتبر من اهم التوزيعات في المسائل المتعلقة بالمكالمات التليفونية و حركة المرور، بعض الظواهر النادرة مثل الزلزال، و الحرائق، الحوادث على أحدى الطرق، عدد الاخطاء المطبعية فى صفحة ما من كتاب و غير ذلك. و دالة كثافة الاحتمال لتوزيع بواسون هي :

$$f(x) = \begin{cases} \dfrac{\lambda^x \; e^{-\lambda}}{x\,!} & , \qquad x = 0,1,2,3,.......... \\ 0 & , \end{cases}$$

حيث $e = 2.718$ تمثل مقدار ثابت. $\lambda = n\;p$

و نأخذ X قيما صحيحة موجبة ذات مدى من الصفر الى ما لانهاية.

ويمكن التعبير عن دالة الكثافة الاحتمالية للتوزيع باستعمال الايعاز dpois والشكل العام للايعاز هو :

dpois(x, lambda)

اذ ان :

dpois: الايعاز

x: المتغير العشوائي

Lambda: معلمة التوزيع

واما دالة التوزيع التجميعية فيمكن التعبير عنها باستعمال الايعاز ppois والشكل العام للايعاز هو :

ppois(x , lambda)

مثال: اذا كانت نسبة الوحدات المعيبة في انتاج نوع من اللمبات الكهربائية هي 0.02 وان عدد الوحدات المعيبة يتبع توزيع بواســـون. نفرض أننا سحبنا عينة عشوائية من عشرة لمبات. المطلوب

1ـ ايجاد التوزيع الاحتمالي لهذا المتغير.

2ـ احتمال الحصول على مصدة واحدة معيبة.

3ـ احتمال الحصول على مصدة معيبة على الاكثر.

الحل:

$$\lambda = n\,p = (10)\;(0.02) = 0.2$$

1. أن X يتبع توزيع بواســون   ∴ دالة كثافة الاحتمال هي

$$f(x) = \frac{e^{-0.2}\,(0.2)^x}{x\,!} \qquad x = 0,1,2,........$$

2,3 باستعمال الايعاز وكما يأتي:

```
Console ~/
> x<-1
> lambda<-0.2
> fx<-dpois(x,lambda)
> fx
[1] 0.1637462
> #2.
> ppois(x,lambda)
[1] 0.9824769
```

3-2-12 التوزيع الاسي exponential distribution

عادة ما يستخدم التوزيع الأسي في مسائل متعلقة بقياس الزمن. من ذلك مدة خدمة شباك البريد، مدة المكالمة هاتفية، مدة تفريغ باخرة الشحن، مدة تصليح آلة، مدة انتظار زبون قبل الحصول على الخدمة. في العلوم الدقيقة يستخدم التوزيع الأسي لتمثيل مدة حياة الذرات المشعة قبل أن تتفكك، حيث يعبر الوسيط عن اللحظة التي يبقى فيها نصف المجتمع الأصلي. كقاعدة عامة يستخدم التوزيع الأسي لتمثيل مدة حياة ظاهرة ما إذا كان لها متوسط ثابت $\frac{1}{\lambda}$ وكانت هذه الظاهرة لا تخضع للتقادم أي أن مدة حياة الظاهرة بعد لحظة ما T لا تتبع اللحظة T؛ أي لا تتأثر بالمدة التي دامتها الظاهرة من قبل. مثلا قد نستبعد استخدام التوزيع الأسي لتمثيل مدة حياة آلة عاملة قبل تعطلها

لأن احتمال تعطلها في لحظة ليس مستقلا عن المدة التي عملتها الآلة من قبل، كذلك الأمر بالنسبة لمدة حياة الإنسان.

دالة الكثافة الاحتمالية للتوزيع الاسي

يقال أن لمتغير لعشوائي ما أنه يتبع التوزيع الأسي إذا كانت دالة كثافته تعطى بالشكل التالي:

$$f(x,\lambda)=\begin{cases}\frac{1}{\lambda}e^{\frac{-x}{\lambda}} & , \quad x\geq 0\\ 0 & , \quad x<0\end{cases}$$

ويم التعبير عن هذه الدالة في برنامج R بالايعاز dexp والشكل العام للايعاز هو :

dexp(x,1/mu)

اذ ان :

dexp: يمثل الايعاز

x: يمثل المتغير العشوائي

1/mu: معلمة التوزيع

اما دالة التوزيع

دالة التوزيع التراكمي لمتغير عشوائي يتبع التوزيع الأسي تعطى بالشكل التالي:

$$F(x)=1-e^{\frac{-x}{\lambda}}$$

ويتم التعبير عنها في **برنامج R** بالايعاز pexp والشكل العام للايعاز هو:

pexp(x,1/mu)

مثال: اذا كان x متغير عشوائي يتبع توزيع الاسي بالمعلمة $\lambda=1/4$ جد ما يلي :

1. $p(x=0.5)$

2. $p(x=2)$

الحل:

```
Console ~/
> #1.
> x<-0.5
> mu<-1/4
> pexp(x,mu)
[1] 0.1175031
> #2.
> x<-2
> pexp(x,mu)
[1] 0.3934693
> |
```

4-2-12 التوزيع الطبيعي Normal distribution

يعتبر التوزيع الطبيعي من أهم التوزيعات المستمرة حيث يلعب دوراً أساسياً في عملية المعاينة Sampling ، كما يستخدم لوصف الأنماط التكرارية للعديد من الظواهر الإحصائية مثل التغيرات الطبيعية التي تحدث للإنسان والحيوان والعوامل البيئية بشكل عام . والتوزيع الطبيعي يدرس سلوك المتغيرات العشوائية المتصلة مثل درجة الحرارة والطول والوزن والدخل والأخطاء العشوائية الناتجة عند تحليل الانحدار .

وبدراسة شكل منحنى التوزيع الطبيعي نجد أنه منحنى متماثل حول الوسط الحسابي للتوزيع ، ويأخذ الشكل الجرسي وله قمة واحدة ويمتد طرفاه إلى ما لانهاية مقتربين من المحور الأفقي شيئا فشيئا دون أن يتماسا مع هذا المحور . وإذا أسقطنا عموداً من قمة المنحنى على المحور الأفقي فإن هذا العمود يعتبر محوراً للتماثل لأنه يقسم المساحة تحت المنحنى إلى قسمين متساويين تماماً وينطبق كل منهما على الآخر تمام الانطباق ومساحة كل قسم تساوى 50% من المساحة الكلية تحت المنحنى وينتج عن هذا التماثل أن قيم الوسط الحسابي والوسيط والمنوال للتوزيع الطبيعي تكون متساوية .

دالة الكثافة الاحتمالية Probability Density Function

إذا كـان x متغيرا عشوائيا متصلا (مستمر) يتبع التوزيع الطبيعي بمتوسط μ وانحراف معياري σ فإن دالة كثافة الاحتمال للمتغير العشوائي X تعطى بالعلاقة :

$$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \; e^{-1/2(\frac{x-\mu}{\sigma})^2} \qquad -\infty < x < \infty$$

وحيث أن استخدام هذه الدالة لحساب الاحتمالات المختلفة تكتنفه صعوبات رياضية كثيرة تعتمد بالدرجة الأولى على معرفة تامة بعلم التكامل ، علاوة على أنه كما ذكرنا يوجد عدد لا نهائى من التوزيعات الطبيعية والتي تحدد كل منها قيم المعلمتين μ ، σ فإنه لحسن الحظ يمكن تحويل أي توزيع طبيعي إلى توزيع الطبيعي المعياري له جداول خاصة تعرف باسم جداول المساحات تحت منحنى التوزيع الطبيعي المعياري متى علم متوسطه وانحرافه المعياري .

وللتعبير عن دالة الكثافة الاحتمالية للتوزيع الطبيعي في **برنامج R** نستعمل الايعاز dnorm والشكل العام للايعاز هو:

dnorm(x , mu , sigma)

اذ ان:

x: المتغير العشوائي

mu: معلمة المتوسط

Sigma: معلمة الانحراف المعياري.

1-2-4-12 التوزيع الطبيعي القياسي Standard Normal Distribution

يعتبر التوزيع الطبيعي القياسي الة خاصة من التوزيع الطبيعي حيث إذا كـان X متغير عشـوائي يتبـع التوزيـع الطبيعـي بمتوسـط μ وانحـراف معيـاري σ فـإن Z تتبـع التوزيع الطبيعي المعياري (القياسي) بمتوسط صفر، وانحراف معياري يساوى الواحد الصحيح حيث:

$$Z = \frac{X - \mu}{\sigma}$$

وتسمى Z أيضا بالقيمة المعيارية أو الدرجة المعيارية وهى تساعد على إيجاد المسـاحة أو الاحتمال المطلوب أسفل أي منحنى توزيع طبيعي وذلك باستخدام جدول المسـاحات تحت منحنى التوزيع الطبيعـي المعيـاري . هذا وتأخذ دالـة كثافـة الاحتمـال للمتغيـر Z الشكل الآتـي :

$$f(Z) = \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2}Z^2} \qquad -\infty < Z < \infty$$

ومن الجدير بالذكر هنا أن الدالة لا تعتمد علـى معـالم مجهولـة القيمـة وبالتـالي فقد استخدمت في حساب جدول المساحات تحت منحنى التوزيع الطبيعي المعيـاري المشـار إليه .

وللتعبير عنه في برنـامج R نستعمل نفس الايعاز dnorm(x,0,1) لكن المعلمات تختلف .

اما بالنسبة الى دالة التوزيع للتوزيع الطبيعي فيتم التعبير عنها في برنـامج R بالايعـاز pnorm والشكل العام للايعاز هو :

pnorm(x,mu , sigma)

والجدول التالي يبين اهم دوال التوزيعات الاحتمالية وما يعبر عنها في برنامج R :

التوزيع	دالة الكثافة الاحتمالية pdf في برنامج R	دالة التوزيع التجميعية cdf في برنامج R
Beta	dbeta	pbeta
Binomial	dbinom	pbinom
Cauchy	dcauchy	pcauchy
Chi-Square	dchisq	pchisq
Exponential	dexp	pexp
F	df	pf
Gamma	dgamma	pgamma
Geometric	dgeom	pgeom
Hypergeometric	dhyper	phyper
Logistic	dlogis	plogis
Log Normal	dlnorm	plnorm
Negative Binomial	dnbinom	pnbinom
Normal	dnorm	pnorm
Poisson	dpois	ppois
Student t	dt	pt
Studentized Range	dtukey	ptukey

punif	dunif	Uniform
pweibull	dweibull	Weibull
pwilcox	dwilcox	Wilcoxon Rank Sum Statistic
psignrank	dsignrank	Wilcoxon Signed Rank Statistic

تم بحمد الله تعالى

المصادر

1. A Beginner's Guide to R, 2009
2. A First Course in Statistical Programming with R, 2007
3. A Handbook of Statistical Analyses Using R, 2006
4. Advanced R, Hadley Wickham, 2014
5. Data Analysis and Graphics: Using R - an Example-Based Approach. Cambridge Series in Statistical and Probabilistic Mathmatics, 2003
6. Extending the Linear Model with R, Julian J. Faraway, 2004
7. Hands-On Programming with R: Write Your Own Functions and Simulations, Garrett Grolemund, 2014
8. Introducing Monte Carlo Methods with R, 2009
9. Introductory statistics with R, Peter Dalgaard, 2002
10. Learning RStudio for R Statistical Computing, 2012
11. Learning RStudio for R Statistical Computing, 2012
12. Mastering Predictive Analytics with R, Rui Miguel Forte, 2015
13. Practical Data Science with R, Nina Zumel, 2014
14. R for Everyone: Advanced Analytics and Graphics, Jared P. Lander, 2013
15. Software for Data Analysis: Programming with R, John Chambers, 2008
16. Statistical Analysis with R, John M. Quick, 2010
17. Statistical Computing with R, Maria L. Rizzo, 2007

18. Statistics: An Introduction Using R, Michael J Crawley, 2005
19. The Art of R Programming: A Tour of Statistical Software Design, Norman Matloff, 2011
20. Using R for Introductory Statistics, John Verzani, 2004