Terms and Sequences:

Term 1 = T_1 etc.

$2T_1$ = 2x T_1

60, 45, 30, 15, 0 = **$Tn=Tn_{-1}$ -15, T_1 = 60**

Sequences on Calculator:

1. Lists and Spreadsheets
2. Equation Box
3. Menu
4. Data (3)
5. Generate Sequence (1)

First Order Linear Recurrence:

Determine: Steady State, Increasing or Decreasing

On Calculator:

$Tn = Tn_{-1} + 4 = U(n) = U(n-1) +4$

Recursive Rule: Defines a sequence by its previous terms

E.g. **$Tn_{+1}=Tn + d, T_1 = x$** or **$Tn_{+1}=r \times Tn, T_1 = x$**

General Rule: Used to jump to a later term

E.g. **$Tn= 4+(n-1)x3$** etc. **$Tn= 2x2^{n-1}$** etc.

Arithmetic Progression:

$Tn = a + (n - 1) d$

a= First Term, **d**= Common Difference, **n**=Term Number

Geometric Progression:

$Tn = ar^{n-1}$

a= First Term, **r**= Common Ration, **n**=Term Number

Initial Amount = T_0

If there is T_0 remove '-1' from 'n-1' in equation

Time Series Data

Describe:

Linear/Non-Linear

Long Term Trend:

- Positive/Increasing
- Negative/Decreasing

Seasonal Pattern:

- Peaks (Highs)
- Troughs (Lows)

Random Variations:

- A short-term variable which does not fit the visual pattern

Moving Averages:

Used to smooth out data:

- To fit linear line of regression
- Identify overall long-term trend

Choose moving average according to length of cycle

How to find 3-pt MA:

3PMA = (a + b + c) / 3

How to find 4-pt CMA:

4PMCA = (0.5a + b + c + d + 0.5e) /4

Seasonal Indices:

Calculated using average percentage method

Example:

Year	Period of Time	Visitors (Thousands)
2011	1st 4 Months	45
	2nd 4 Months	26
	3rd 4 Months	34

Total Number Visitors = 45000 + 26000 + 34000 = 10500

Average Visitors per 4-month period = 105000 / 3 = 35000

1st 4 Months = 45000 / 35000 = 128.6%
2nd 4 Months = 26000 / 35000 = 74.3%
3rd 4 Months = 34000 / 35000 = 97.1%
(100% = Average)

Note: For more years work out seasonal effect for each season. Then find overall average of season across all years
E.g. (128.6+120.1+115.2)/3 = 121.3 = 1.213

To seasonalise data multiply by SI (decimal)

Deseasonalised Data:

- Removes Cycles and Smooths out the Data
- Reduces Presence of Seasonal Fluctuations

Deseasonalised Data = Real Data / Seasonal Index (Decimal)
Real Data = Deseasonalised Data *x* Seasonal Index

Finance:

Simple Interest:

I = Prt

I = Interest Earned
P = Principal Amount
r = Interest Rate (As Decimal)
t = Time

Note: ***r*** and ***t*** must use same period of time

A = I + P

Compound Interest:

$$A = P\left(1 + {}^{r}/_{n}\right)^{nt}$$

A = Total Amount
P = Principal Amount
r = Interest Rate (As Decimal – Per Year)
n = How many times interest is calculated
t = Time in years

I = A – P

Effective Annual Interest Rate:

(On Formula Sheet)

$i_{effective}$ = Effective annual interest rate (as decimal)
i = Annual interest rate (as decimal)
n = number of times interest is compounded per year

Finance Solve:

Calc → Menu → Finance (8) → Finance Solver (1)

N: Number of times interest is calculated

I: Interest Rate (Annual) (Percentage)

PV: Present Value

Pmt: Payment Amount

FV: Future Value

Ppy: Payments Per Year

Cpy: How many times interest is compounded per year

Monthly Compounding Interest Rate (Recursive):

Divide interest rate (as decimal) by 12
E.g. 0.075/12 = 0.00625

Loans:

Borrowing a sum of money that needs to be payed back in full over a period of time

PV	PMT	FV
Positive	Negative	0

E.g. Borrow 1500 @ 18%p.a. comp. monthly. Pay off 320 per month
$Tn=T_{n-1}$ x 1.015 – 320, T_1 = 1500

Investments:

Depositing Money into the bank that grows over time due to interest, whilst making regular contributions to grow the account

PV	PMT	FV
Negative	Negative	Positive

E.g. 100 000 invested @ 7% p.a. comp. annually
$Tn=T_{n-1}$ x 1.07, T_1 = 100 000

Annuities:

A sum of money that is withdrawn from the bank at regular intervals until it fully depletes

PV	PMT	FV
Negative	Positive	0

Perpetuities:

Annuity where the money lasts forever. Interest is the same as regular withdrawal

Q = PE

- Q: Annual withdrawal amount
- P: Principal
- E: Effective annual interest rate

Depreciation:

$100 000 after 5 years

Flat rate of $10 000 per year: 10 000 x 5

10% p.a.: $100\,000 \times 0.9^5$
(In Calc: Un=U(n-1) x 0.9, T1 = 100 000)

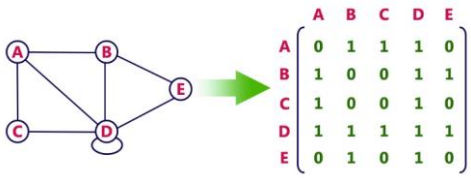
Recursive Rule:

When compounded monthly divide interest rate by 12

Superannuation: (Annuity)
E.g. $750,000 giving 6% p.a. compounded annually withdrawing $60,000 yearly
$Tn = 1.06Tn_{-1} – 60,000, T_o = 750,000$

Graphs and Networks:

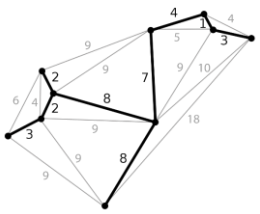
Euler's Rule: V – E + F = 2 (V = No. of Vertices, E = No. of Edges, F = No of Faces)
Works in any connected planar network.
Planar Graphs: Can be re drawn so no edges crossing
Order (Degree) of Vertex: Number of edges meeting at a vertex
Simple Graphs: Undirected, Unweighted, No Multiple Edges, No loops
Connected Graph: Graph with a possible path between every vertex
Subgraphs: Section of a network
Directed Graph: Where all edges are directed
Weighted Graphs: Information on edges
Tree: Connected simple graph with no cycles or multiple edges
Bipartite Graph: Vertices split into 2 groups. Edges can't connect vertices in same group.
Complete Graph: Simple Graph where every vertex is connected
Adjacency Matrix: Shows how many ways adjacent vertices are connected.



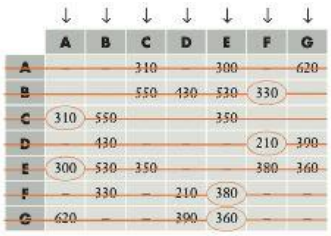
	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	1	1
E	0	1	0	1	0

Adjacency Matrix^2: How many ways you can do between vertices using 2 edges.
Open: Ends on different vertex
Closed: Ends on same vertex
Walk: A sequence of vertices connected by edges. Can use repeated vertices and edges
Path: Walk with no repeated edges or vertices (Closed: Cycle)
Trail: No repeated edges (Closed: Circuit)
Traversability: Can be traced without going over the same edge twice. Must have 0 or 2 odd vertices
Eulerian Circuit: Traversable. Begins and ends on same vertex
Semi Eulerian: Begins and ends at different vertex
Hamiltonian Cycle: Visit every vertex once. Returns to first vertex. (Don't need to use all edge)
Semi-Hamiltonian: Every vertex once. Ends on different vertex

Minimum Spanning Tree: A subgraph that connects all vertices into a tree



Minimum Spanning Tree - Prims Algorithm:
Find 'Nearest Neighbour' of vertices
Minimum Spanning Tree - Kruskal's Algorithm:
Keep finding shortest edge until edges are connected. Don't introduce any cycles.
Minimum Spanning Tree - With Matrix:
Steps:
1. Cross out top row
2. Look down column to find shortest connection – cross out row
3. Find shortest connection in columns of rows that have been crossed out
4. Continue process until all rows are crossed out



	A	B	C	D	E	F	G
A			310		300		620
B			550	430	530	330	
C	310	550			350		
D		430				210	390
E	300	530	350			380	360
F		330		210	380		
G	620			390	360		

Maximum Flow:
Source: Start Point
Sink: End Point
Minimum Cut: Smallest cut that minimises sum of edge weights
Steps:
1. Find path that that's from source and ends at sink
2. Determine maximum flow
3. Subtract flow from all edges along path
4. Continue until no longer possible
5. Determine maximum flow by adding flow amount of all paths chosen

Project Networks:
Displays a network showing tasks and times to complete a project
Critical Path: The path with the longest time. Allows all jobs to be done. Use to MCT
Minimum Completion Time: Least time needed to complete all activities
Float/Slack Time: Extra time you can spend on an activity without affecting minimum completion time
EST: Forward Scanning – Choose largest number
LST: Backward Scanning – Choose smallest number
Float Time= LST – EST – TASK

Hungarian Algorithm:
If rows do not equal columns add dummy row/column
Steps:
(For Maximisation Only: Subtract every entry from largest entry**)**
1. Subtract smallest number from each row from other numbers in row
2. Subtract smallest number from each column from other numbers in column
3. Draw straight lines through rows/columns so all zeros are covered – use minimum number of lines
4. If lines = rows go to Step 6. If not…
5. Find smallest number in matrix not covered by any line. Subtract from all uncovered numbers and add to any numbers covered by line twice. Return to Step 3.
6. Select and circle a 'zero' in each column so no other 'zeros' are in the row
7. Match circled entries with original cost matrix to find solution

Bivariate Data:

Notes – Applications Unit 3 + 4 – Cameron Crocker

Row and Column Percentages:

Example:

	Romance	Sci-Fi	Horror	
20-24	6	20	10	**=36**
25-29	11	13	7	**=31**
30-34	8	4	20	**=32**
	=25	**=37**	**=37**	

Row Percentage:

	Romance	Sci-Fi	Horror	
20-24	$^{6/36x100}$ 16.6%	$^{20/36x100}$55.5%	$^{10/36x100}$27.7%	**100%**
25-29	35.5%	41.9%	22.6%	**100%**
30-34	25%	12.5%	62.5%	**100%**

Column Percentage:

	Romance	Sci-Fi	Horror
20-24	$^{6/25}$ 24%	54%	27%
25-29	$^{11/25}$ 44%	35%	19%
30-34	$^{8/25}$ 32%	11%	57%
	100%	**100%**	**100%**

What you are working out (100%) is always on base of graph.

Explanatory Variable: (x)

100% totals are always explanatory variables.

e.g. Gender *explains* what you like to watch.

Correlation Coefficient:

r= Strength of relationship between two variables.

Strong correlation is reliable and supports linear model

Always $-1 \leq r \leq 1$

Strong Positive	$0.75 < r < 0.99$
Moderate Positive	$0.5 < r < 0.74$
Weak Positive	$0.25 < r < 0.49$
None	$-0.24 < r < 0.24$
Weak Negative	$-0.25 < r < -0.49$
Moderate Negative	$-0.5 < r < -0.74$
Strong Negative	$-0.75 < r < -0.99$

STEPS:

1. Lists and Spreadsheets
2. Menu
3. Statistics (4)
4. Stat Calculations (1)
5. Linear Regression (mx+b) (3)
6. r= correlation coefficient

Correlation tells you about the strength of association, **NOT CAUSE**

Start New Problem:

Document (doc) → Insert (4) → Problem (1)

Scatter Graphs:

Line of Best Fit: y=mx+c

Used for predictions:

-**Interpolation:** Predicting within data set

-**Extrapolation:** Predicting outside data set

STEPS:

1. Lists and Spreadsheets
2. Add variables and values
3. Open Data and Statistics
4. Add variables on axis

To Add LOBF (Linear Regression):

1. Menu
2. Analyse (**4**)
3. Regression (**6**)
4. Show linear (mx+b) (**1**)

View Residual Plot:

1. Menu
2. Analyse (**4**)
3. Residuals (**7**)
4. Show Residual Plot (**2**)

Predictions:

Sub into linear regression equation

Residuals:

Vertical distances between data points and the line of regression
Random: Fits linear model
Non-Random: Does **not** fit linear model

Residual value:

$e = y - \hat{y}$

Residual = Observed (real) – Predicted (From line of regression)

Coefficient of Determination:

r^2 = Percentage of variation in response variable that can be explained by explanatory variable

Tells us x% of variation can be explained by etc… The rest is a result of other factors