**Bivariate Data**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Party A | Party B | Totals |
| Male | 215 | 104 | 319 |
| Female | 42 | 61 | 103 |
| Totals | 257 | 165 | 422 |

*Row percentages*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Party A | Party B | Totals |
| Male | 67% | 33% | 100% |
| Female | 41% | 59% | 100% |

*Column percentages*

|  |  |  |
| --- | --- | --- |
|  | Party A | Party B |
| Male | 84% | 63% |
| Female | 16% | 37% |
| Total | 100% | 100% |

Explanatory variable: the variable used to explain or predict a difference in the response variable

Response variable: also called the dependent variable

**Scatterplots**

Positive: from bottom left to top right

Negative: from top left to bottom right

Strong: how close the points follow a linear pattern

Eg: The graph above is a strong positive graph

**Correlation coefficient/determination**

R: measures the direction and strength of a linear relationship. Says that ‘X’ is a good predicter of ‘Y’

R^2: measures the percentage variation in both variable with respect to each other

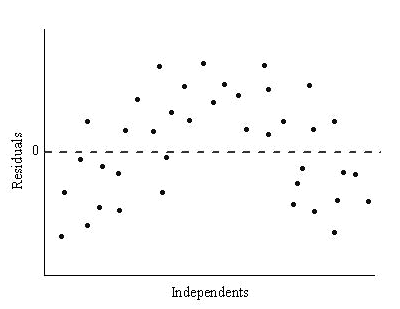
**Euler and planar graphs**

If Euler’s rule works, then the graph can be drawn as planar

Planar: no edges in a graph cross over each other

**Residuals**

Refers to the vertical distance between a data point and line of best fit



Linear model only appropriate if residuals are in random order

*How to calculate*

1. Statistics
2. Enter numbers
3. Calc
4. Regression
5. Linear regression
6. Copy residuals to list 3

**Sequences**

Arithmetic: each term is found by adding or subtracting/linear growth

|  |  |
| --- | --- |
| Arithmetic sequeces | |
| Explicit |  |
| Recursive |  |

Where;

A: the first term in the sequence

N: the ‘nth’ term in the sequence

D: the common difference

*Examples*

OR

Geometric: each term is found by multiplying or dividing/exponential growth

|  |  |
| --- | --- |
| Geometric sequences | |
| Explicit |  |
| Recursive |  |

Where;

r: the common ratio

A: the first term

N: the ‘nth’ term in the sequence

*Examples*

OR

**Types of graphs**

Simple graph: does not contain loops or multiple edges

Connected graph: a possible path between every vertex

Directed graph: the is directed by the use of arrows

Weighted graph: edges on a graph that have been assigned a numerical value

Tree: connected graph that doesn’t contain any cycles or multiple edges

Bipartite: has 2 sets of vertices/2 groups

**Network terminology**

Loop: an edge joining a vertex to itself

Multiple edges: 2 or more edges that have the same start and end vertices

Bridge: an edge that connects 2 parts of a graph that would otherwise result in an isolated vertex

Degree of a vertex: number of edges connected to a vertex

Open walk: starting and ending vertices are different

Closed walk: starting and ending vertices are the same, without the end and start vertices being connected

Open path: starts in one place and finishes in another. Can’t repeat vertices or edges

Closed path (cycle): visits every vertex and end at the starting vertex (the last vertex is joined to the first vertex). Can’t repeat vertices or edges

Open trail: a walk with no repeated edges and starts and finishes at different vertices. May repeat vertices but not edges

Closed trail (circuit): a walk with no repeated edges that starts and ends at the same vertex. May repeat vertices but not edges

**Eulerian graphs**

Eulerian graph: contains a circuit/closed trail that visits all edges in the graph once only and may repeat vertices

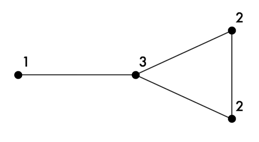
*Is Eulerian if all the degrees of the vertices are even*

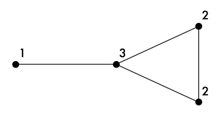
*A close up of a wire fence

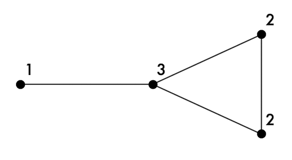
Description automatically generated*

Semi Eulerian graph: contains an open trail that visits all edges in the graph once only and may repeat vertices

*Is semi-Eulerian if only 2 vertices have an odd degree*

****

****

****

**A picture containing object, photo, clock, watch

Description automatically generated**

**A picture containing object, watch, clock

Description automatically generated**

**Interest**

*Simple interest*

Where;

P: principle

R: rate as a decimal

T: time in years

*Compound interest*

Where;

P: principle

R: rate as a decimal

N: number of times compounded in a year

T: time in years

*Effective interest rate*

Where;

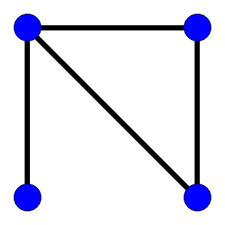
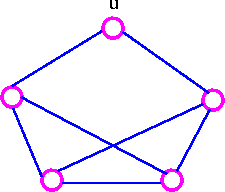
I: amount of interest per annum

N: number of compounding

**Hamiltonian graphs**

Hamiltonian graph: contains a closed path/cycle that visits all vertices in the graph once only (except start and end vertex) and does not have to pass through all edges

*Visits all vertices once except for start and end*

**

Semi-Hamiltonian: contains an open path that visits all vertices in the graph once only and does not have to pass through all edges

*Must repeat a vertex in order to visit them all*

**Financial calculator**

Jackson borrows $20,000 at 12% p.a. compounding monthly. He pays $350 every month to pay off the loan. How much would he still owe after 5 years of payments?

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | Loan | PMT | -350 |
| N |  | FV | **-$7749.55** |
| I% | 12 | P/Y | 12 |
| PV | 20000 | C/Y | 12 |

N: time in years × number of compounding periods annually

I: interest rate

PV: present value

PMT: amount paid each period

FV: future value

P/Y: payments per year

C/Y: compounding per year

Lily invests $10,000 at 7% p.a. compounding half-yearly. Lily wants her account to reach $50,000 in 10 years. How much does she need to deposit every six months to reach this goal?

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | Investment | PMT | **-$1064.44** |
| N | 10×2 | FV | 50000 |
| I% | 7 | P/Y | 2 |
| PV | 10000 | C/Y | 2 |

Grace invests $700,000 to buy an annuity that pays $50,000 at 5.4% p.a. compounding annually. How many years will Grace be able to withdraw money for?

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | Annuity | PMT | 50000 |
| N | **26.82** | FV | 0 |
| I% | 5.4 | P/Y | 1 |
| PV | 700000 | C/Y | 1 |

**Analysing loans**

*Calculating final payment*

R: regular payment amount

: first negative value that appears

*Total cost of a loan*

N: number of full payments

R: regular payment amount

F: final payment amount

**Depreciation**

*Flat rate*

Eg, 10% = 0.9

**Deseasonalising data**

|  |  |  |  |
| --- | --- | --- | --- |
| Yr/Pd | Period 1 | Period 2 | Period 3 |
| Year 1 | 411 | 648 | 699 |
| Year 2 | 532 | 632 | 741 |

1. *Average the rows*

|  |  |
| --- | --- |
| Year 1 | (411+648+699)/3  =586 |
| Year 2 | (532+632+741)/3  =635 |

1. *Divide averages into original values*

|  |  |  |  |
| --- | --- | --- | --- |
| Yr/Pd | Period 1 | Period 2 | Period 3 |
| Year 1 | 411/586  =0.7014 | 648/586  =1.1058 | 699/586  =1.1928 |
| Year 2 | 532/635  =0.8378 | 632/635  =0.9953 | 741/635  =1.1669 |

1. *Average the columns with the answers derived from the division in step 2/seasonal index*

|  |  |
| --- | --- |
| Average period 1 | (0.7014+0.8378)/2  =0.7696 |
| Average period 2 | (1.1058+0.9953)/2  =1.0505 |
| Average period 3 | (1.1928+1.1669)/2  =1.1799 |

1. *Deseasonalise data by dividing the original values into the answers derived from step 3*

|  |  |  |  |
| --- | --- | --- | --- |
| Yr/Pd | Period 1 | Period 2 | Period 3 |
| Year 1 | 411/0.7696  =534.04 | 648/1.0505  =616.85 | 699/1.1799  =592.42 |
| Year 2 | 532/0.7696  =691.27 | 632/1.0505  =601.62 | 741/1.1799  =628.02 |

**Seasonal index**

From example above

|  |  |
| --- | --- |
| Average period 1 | (0.7014+0.8378)/2  =**0.7696** |
| Average period 2 | (1.1058+0.9953)/2  =**1.0505** |
| Average period 3 | (1.1928+1.1669)/2  =**1.1799** |

If there is 3 periods then the seasonal index has to equal 300% or 3. Eg, 0.7696+1.0505+1.799 = 3 or 300%

**Perpetuities**

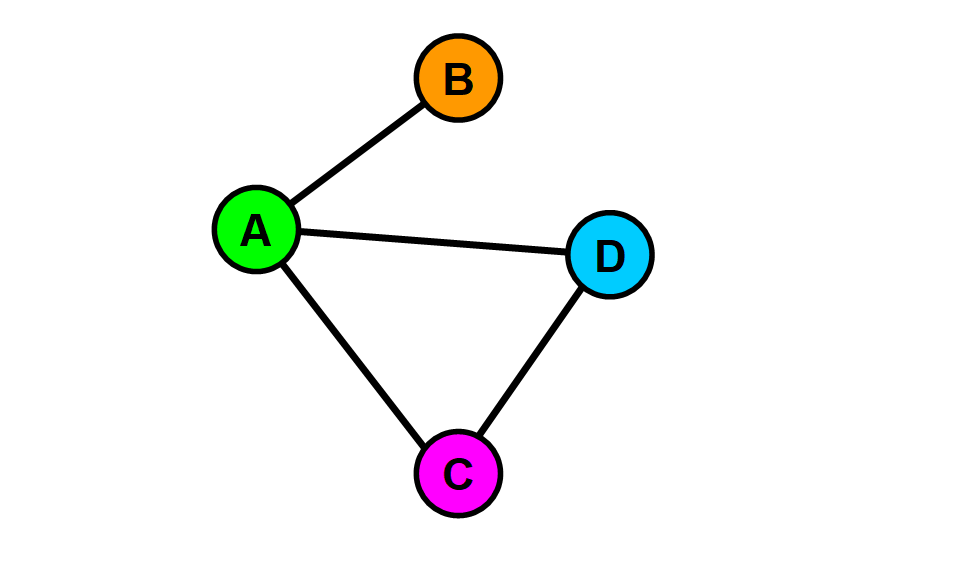
Substitute **x** into principle rate or withdrawal to find answer

**Annuities**

WI: percentage increase of withdrawal, eg 1.02 for 2%

Bolded: only happens if withdrawal increases each period

**Adjacency matrices**



A

B

C

D

C

B

A

D

**Moving averages**

*Odd moving averages*

|  |  |  |
| --- | --- | --- |
| Period | Value | 3-PMA |
| 1 | 15 |  |
| 2 | 16 | 17 |
| 3 | 20 | 17.67 |
| 4 | 17 | 18.33 |
| 5 | 18 |  |

*Even moving averages*

|  |  |  |  |
| --- | --- | --- | --- |
| Period | Value | 4-PMA | 4-PCMA |
| 1 | 23 |  |  |
| 1.5 |  |  |  |
| 2 | 22 |  |  |
| 2.5 |  | 20 |  |
| 3 | 18 |  | 19.5 |
| 3.5 |  | 19 |  |
| 4 | 17 |  |  |
| 4.5 |  |  |  |
| 5 | 19 |  |  |

**Minimum spanning trees**

A close up of a logo

Description automatically generated

Reach every vertex in the fewest amount of time

*Prim’s algorithm*

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 230 | 210 | 340 |
| B | 230 | 180 | 215 |
| C | 210 | 200 | 185 |

1. Choose a vertex, say A
2. Cross row A out since A can’t go to A
3. Choose the smallest number in column A, 210
4. Cross row C out since you’ve just picked it
5. Start at the top of column C and do it again

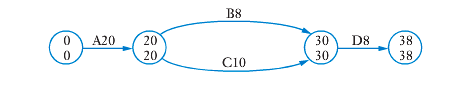
**Maximum flow**

A close up of a map

Description automatically generated

Minimum cut: cut can be made on the graph which equals the maximum flow

**Project networks**



Minimum completion time: how quick can all the jobs be done

Critical path: the path that equals the minimum completion time

Earliest start time: how early can a task start, eg D = 28

Latest start time: how late a task can start, eg D= 30

Float time: how long a task has to spare before it has to start, eg B has a float time of 2

**Hungarian algorithm**

1. Construct a cost matrix
2. **Take the largest number away from every number in the matrix = only for maximum**
3. Subtract the smallest number in each row from every number in that row
4. Subtract the smallest number in each column from every entry in that column
5. Draw as few vertical and horizontal lines as possible. If the number of lines matches with rows/column then your fine
6. If not, take the smallest number away that hasn’t been lined out from all the other numbers who haven’t been lined out

*No. people ≠ number of tasks*

1. Add a row or column of 0’s to make it even
2. Subtract the smallest number in each row or column which changes the matrix from every number in that row/column
3. Draw vertical and horizontal lines
4. Where 0’s cross over, add the number you took away in the row or column takeaways and takeaway smallest number that isn’t crossed out from all the other numbers
5. Draw lines and repeat until the amount of lines matches the rows or columns

**Other**

Extrapolation: estimating a value outside the original data given by assuming that existing trends present will continue

Interpolation: estimation of a value which is within the limits of the data set