Mathtrace – user guide

VII. Applications: sciences, economics and optimization

Contents

VII. Applications: sciences, economics and optimization

47. Chemistry
48. Electrical engineering
49. Biology and medicine
50. Economics, finance and insurance
51. Linear programming, transportation, assignment and matrix games
52. Graphs, networks and logistics
53. Psychometrics

47. Chemistry

Chemistry calculations are in the "Applications" section of the home screen, the Chemistry field: balancing a chemical equation, the pH of an acid, a base, a salt or a polyprotic acid, a buffer, a titration with its curve, the Arrhenius equation and the Beer–Lambert law. In the form you choose the problem and the quantity sought and fill in the values with their units; the "?" icon explains where the formula comes from. Small constants can be typed as 1.8e-5 or 1,8·10^-5, measured values (a table of temperatures, a calibration) as numbers separated by spaces.

A chemical equation can simply be typed into the manual input, e.g. Fe + O2 -> Fe2O3 (the arrow -> or =, an equilibrium <=>), also with ions MnO4^- and Fe^3+, brackets Ca(OH)2, hydrates CuSO4·5H2O and states (aq). The coefficients are the kernel of the matrix of the elements' (and the charge's) balance: the steps write the balance equations, the matrix, its reduced form by Gaussian elimination, the free coefficient as a parameter and the smallest whole numbers, then a check. It also recognizes an equation that can't be balanced (an element on one side only, a substance on the wrong side) or whose balance isn't unique.

The pH is computed exactly (for a weak acid by a quadratic equation), with the school simplification and the degree of dissociation alongside; for a polyprotic acid the charge balance is solved numerically and the graph shows the shares of the acid's forms against pH. For a buffer with a strong acid or base added it computes the new equilibrium and the change of pH. A titration gets its pH by the region of the curve (the start, the buffer region, the equivalence point, the excess titrant) and the graph shows the whole titration curve. The Arrhenius equation from two measurements or from a table by the regression of ln k against 1/T; the Beer–Lambert law from a known ε, a standard or a calibration line.

Examples

Balancing the oxidation of hydrochloric acid by permanganate.

Input: `KMnO4 + HCl -> KCl + MnCl2 + H2O + Cl2`

Result

$$2\,\mathrm{KMnO_4} + 16\,\mathrm{HCl} \to 2\,\mathrm{KCl} + 2\,\mathrm{Mn(}$$

Steps

- The unknown coefficients get letters
  $a\,\mathrm{KMnO_4} + b\,\mathrm{HCl} \to c\,\mathrm{KCl} + d\,\mathrm{MnCl_2} + f\,\mathrm{H_2O} + g\,\mathrm{C}$
- Atoms are neither created nor destroyed – each element has as many atoms on the left as on the right

$$\begin{aligned} \mathrm{K}: &\quad a = c \\ \mathrm{Mn}: &\quad a = d \\ \mathrm{O}: &\quad 4a = f \\ \mathrm{H}: &\quad b = 2f \\ \mathrm{Cl}: &\quad b = c + 2d + 2g \end{aligned}$$

- The homogeneous system as a matrix – a column for each substance, the products with a minus sign

$$\begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 4 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 \\ 0 & 1 & -1 & -2 & 0 & -2 \end{pmatrix}$$

- Gaussian elimination brings the matrix to its reduced row echelon form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -\frac{2}{5} \\ 0 & 1 & 0 & 0 & 0 & -\frac{16}{5} \\ 0 & 0 & 1 & 0 & 0 & -\frac{2}{5} \\ 0 & 0 & 0 & 1 & 0 & -\frac{2}{5} \\ 0 & 0 & 0 & 0 & 1 & -\frac{8}{5} \end{pmatrix}$$

- One coefficient stays free, $g = t$; the others are expressed by it

$$a = \frac{2}{5}t, \quad b = \frac{16}{5}t, \quad c = \frac{2}{5}t, \quad d = \frac{2}{5}t, \quad f = \frac{8}{5}t$$

- We choose $t = 5$ so that all coefficients are the smallest whole numbers:
  $a = 2, \quad b = 16, \quad c = 2, \quad d = 2, \quad f = 8, \quad g = 5$
- Check – atoms on the left and on the right

$$\begin{aligned} \mathrm{K}: &\quad 2 = 2 \\ \mathrm{Mn}: &\quad 2 = 2 \\ \mathrm{O}: &\quad 8 = 8 \\ \mathrm{H}: &\quad 16 = 16 \\ \mathrm{Cl}: &\quad 16 = 16 \end{aligned}$$

An ionic redox equation with the charge balance.

Input: `MnO4^- + Fe^2+ + H^+ -> Mn^2+ + Fe^3+ + H2O`

Result

$$\mathrm{MnO_4^-} + 5\,\mathrm{Fe^{2+}} + 8\,\mathrm{H^+} \rightarrow \mathrm{Mn^{2+}} + 5$$

Steps

- The unknown coefficients get letters

  $$a\,\mathrm{MnO_4^-} + b\,\mathrm{Fe^{2+}} + c\,\mathrm{H^+} \rightarrow d\,\mathrm{Mn^{2+}} + f\,\mathrm{Fe^{3+}} + g\,\mathrm{H_2O}$$

- Atoms are neither created nor destroyed and the charge is conserved (row q) – for each element and for the charge

  $$\begin{aligned} \mathrm{Mn}: &\quad a = d \\ \mathrm{O}: &\quad 4a = g \\ \mathrm{Fe}: &\quad b = f \\ \mathrm{H}: &\quad c = 2g \\ q: &\quad -a + 2b + c = 2d + 3f \end{aligned}$$

- The homogeneous system as a matrix – a column for each substance, the products with a minus sign

  $$\begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 \\ -1 & 2 & 1 & -2 & -3 & 0 \end{pmatrix}$$

- Gaussian elimination brings the matrix to its reduced row echelon form

  $$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ 0 & 1 & 0 & 0 & 0 & -\frac{5}{4} \\ 0 & 0 & 1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 & -\frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & -\frac{5}{4} \end{pmatrix}$$

- One coefficient stays free, $g = t$; the others are expressed by it

  $$a = \frac{1}{4}t, \quad b = \frac{5}{4}t, \quad c = 2t, \quad d = \frac{1}{4}t, \quad f = \frac{5}{4}t$$

- We choose $t = 4$ so that all coefficients are the smallest whole numbers:

  $$a = 1, \quad b = 5, \quad c = 8, \quad d = 1, \quad f = 5, \quad g = 4$$

- Check – atoms and charge on the left and on the right

  $$\begin{aligned} \mathrm{Mn}: &\quad 1 = 1 \\ \mathrm{O}: &\quad 4 = 4 \\ \mathrm{Fe}: &\quad 5 = 5 \\ \mathrm{H}: &\quad 8 = 8 \\ q: &\quad 17 = 17 \end{aligned}$$

pH of 0.1 M acetic acid (K_a = 1.8·10⁻⁵).

Input: `model(ph; find=pH[one]; c=0.1[mol_l]; K=1.8e-5[one]; kind=weakAcid; constant=K)`

Result

$$\mathrm{pH} \approx 2.88$$

Steps

- Find pH; given: $c = 0.1\,\mathrm{mol/l}, \quad K_a = 1.8 \cdot 10^{-5}$
- pH is the negative decimal logarithm of the concentration of hydrogen (hydronium) ions: pH = −log[H⁺]. A strong acid (HCl) dissociates completely, so [H⁺] = c; a strong base gives [OH⁻] = c and pH = 14 − pOH. A weak acid HA ⇌ H⁺ + A⁻ dissociates only partly, by its dissociation constant Ka = [H⁺][A⁻]/[HA]; this leads to the quadratic equation x² + Ka·x − Ka·c = 0. A salt of a weak acid and a strong base (sodium acetate) reacts with water as a base with Kb = Kw/Ka, a salt of a weak base (NH_4Cl) as an acid. For a polyprotic acid (H_2SO_4, H_3PO_4) the charge balance is solved numerically. At 25 °C, Kw = 10⁻¹⁴. The model's equation

$$\mathrm{HA} \rightleftharpoons \mathrm{H^+} + \mathrm{A^-}, \quad K_a = \frac{[\mathrm{H^+}][\mathrm{A^-}]}{[\mathrm{HA}]}$$

- Expressing the sought quantity

$$K_a = \frac{x^2}{c - x} \Rightarrow x^2 + K_a x - K_a c = 0 \Rightarrow x = [\mathrm{H^+}] = -$$

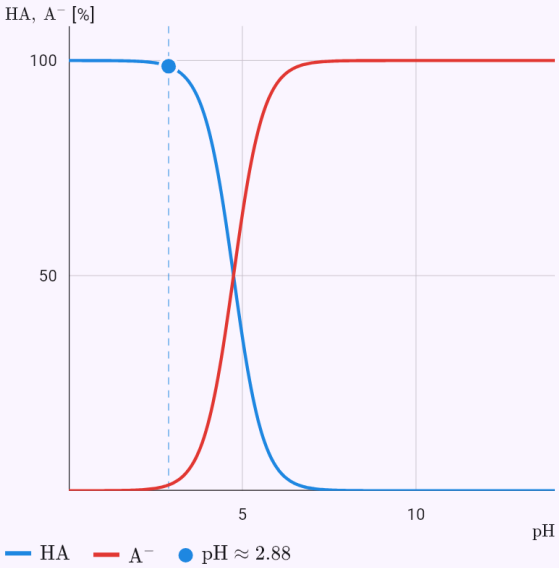
- Substituting

$$[\mathrm{H^+}] = \frac{-1.8 \cdot 10^{-5} + \sqrt{\left(1.8 \cdot 10^{-5}\right)^2 + 4 \cdot 1.8 \cdot 10^{-5} \cdot}}{2}$$

- Degree of dissociation: $\alpha = \frac{x}{c} = \frac{0.001333}{0.1} \approx 1.33\,\%$
- The degree of dissociation is below 5 %, so the simplification c − x ≈ c would do as well:
  $x \approx \sqrt{K_a c} = 0.001342, \quad \mathrm{pH} \approx 2.87$

- pH is the negative logarithm of the H⁺ concentration:
  $\mathrm{pH} = -\log[\mathrm{H^+}] = -\log 0.001333 \approx 2.88$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

pH of 0.1 M phosphoric acid and the shares of its forms.

Input: `model(ph; find=pH[one]; c=0.1[mol_l]; K1=7.5e-3[one]; K2=6.2e-8[one]; K3=4.8e-`

`13[one]; kind=polyprotic; constant=K)`

Result

$\mathrm{pH} \approx 1.62$

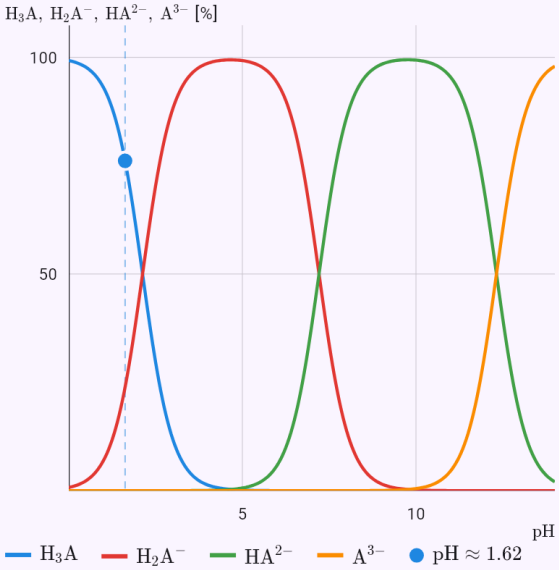
Steps

- Find pH; given
  $c = 0.1\,\mathrm{mol/l}, \quad K_{a1} = 0.0075, \quad K_{a2} = 6.2 \cdot 10^{-8}, \quad I$
- pH is the negative decimal logarithm of the concentration of hydrogen (hydronium) ions: pH = −log[H⁺]. A strong acid (HCl) dissociates completely, so [H⁺] = c; a strong base gives [OH⁻] = c and pH = 14 − pOH. A weak acid HA ⇌ H⁺ + A⁻ dissociates only partly, by its dissociation constant Ka = [H⁺][A⁻]/[HA]; this leads to the quadratic equation x² + Ka·x − Ka·c = 0. A salt of a weak acid and a strong base (sodium acetate) reacts with water as a base with Kb = Kw/Ka, a salt of a weak base (NH_4Cl) as an acid. For a polyprotic acid (H_2SO_4, H_3PO_4) the charge balance is solved numerically. At 25 °C, Kw = 10^{-14}. The model's equation
  $\mathrm{H_3A} \rightleftharpoons \mathrm{H^+} + \mathrm{H_2A^-} \rightleftharpoons 2\,\mathrm{H^+} + \mathrm{HA^{2-}} \rightleftharpoons 3\,\mathrm{H^+} + \mathrm{A^{3-}}$
- The solution is electrically neutral – the charge balance
  $[\mathrm{H^+}] = [\mathrm{OH^-}] + [\mathrm{H_2A^-}] + 2[\mathrm{HA^{2-}}] + 3[\mathrm{A^{3-}}]$
- The concentrations of the acid's forms from the dissociation constants
  $[\mathrm{H_{3-i}A^{i-}}] = c\,\alpha_i, \quad \alpha_i = \frac{K_{a1} \cdots K_{ai}\, h^{3-i}}{h^3 + K_{a1}h^2 + K_{a1}K_{a2}h + K_{a}}$
- Solving the equation numerically (bisection):
  $[\mathrm{H^+}] \approx 0.02389\,\mathrm{mol/l}$
- › The method: bisection, 14 iterations
- pH is the negative logarithm of the H⁺ concentration:
  $\mathrm{pH} = -\log[\mathrm{H^+}] \approx 1.62$
- The second constant is much smaller than the first, so the pH is set mainly by the first step – as for a weak monoprotic acid

$$x = \frac{-K_{a1} + \sqrt{K_{a1}^2 + 4K_{a1}c}}{2} \approx 0.02389, \quad \mathrm{pH} \approx 1.62$$

- The shares of the acid's forms at this pH
  $\mathrm{H_3A}: 76.1\,\%, \quad \mathrm{H_2A^-}: 23.9\,\%, \quad \mathrm{HA^{2-}}: 6.2 \cdot 10^{-5}\,\%$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

An acetate buffer after adding 1 mmol of a strong acid.

Input: `model(buffer; find=pH[one]; pK=4.74[one]; ca=0.1[mol_l]; cs=0.1[mol_l]; V=100[ml];`

```
n=1[mmol]; kind=acidic; constant=pK; add=acid)
```

Result

$$\mathrm{pH} \approx 4.65$$

Steps

- Find pH; given
  $\mathrm{p}K_a = 4.74, \quad c_{\mathrm{HA}} = 0.1\,\mathrm{mol/l}, \quad c_{\mathrm{A^-}} = 0.1\,\mathrm{mol/l}, \quad V$
- Converting the units:
  $V = 100\,\mathrm{ml} = 0.1\,\mathrm{l}, \quad n = 1\,\mathrm{mmol} = 0.001\,\mathrm{mol}$
- The constant from pK:
  $K_a = 10^{-\mathrm{p}K_a} = 10^{-4.74} \approx 1.82 \cdot 10^{-5}$
- A buffer is a mixture of a weak acid and its salt (CH_3COOH + CH_3COONa) or a weak base and its salt (NH_3 + NH_4Cl). The dissociation constant gives the Henderson–Hasselbalch equation pH = pKa + log([A^-]/[HA]). An added strong acid turns the basic component into the acidic one (A^- + H^+ → HA), a base the other way round – the ratio of the components changes only a little, so the pH hardly changes. A buffer works best for a ratio of 0.1 to 10, that is pH = pKa ± 1. The model's equation
  $$\mathrm{pH} = \mathrm{p}K_a + \log \frac{[\mathrm{A^-}]}{[\mathrm{HA}]}$$

- Amounts of the buffer's components
  $n_{\mathrm{HA}} = c_{\mathrm{HA}} V = 0.01\,\mathrm{mol}, \quad n_{\mathrm{A^-}} = c_{\mathrm{A^-}} V = 0.01\,\mathrm{mol}$
- The substance added reacts with one component of the buffer
  $\mathrm{A^-} + \mathrm{H^+} \to \mathrm{HA}: \quad n'_{\mathrm{HA}} = 0.011\,\mathrm{mol}, \quad n'_{\mathrm{A^-}} = 0.009\,\mathrm{m}$
- Expressing the sought quantity: $\mathrm{pH} = \mathrm{p}K_a + \log \frac{n'_{\mathrm{A^-}}}{n'_{\mathrm{HA}}}$
- Substituting: $\mathrm{pH} = 4.74 + \log \frac{0.009}{0.011} \approx 4.653$
- pH before the addition and its change:
  $\mathrm{pH}_0 = 4.74, \quad \Delta\mathrm{pH} = -0.087$

Titration of acetic acid with sodium hydroxide – the equivalence point.

Input: `model(titration; find=pH[one]; ca=0.1[mol_l]; Va=25[ml]; ct=0.1[mol_l]; K=1.8e-`

`5[one]; Vt=25[ml]; kind=weakAcid; constant=K)`

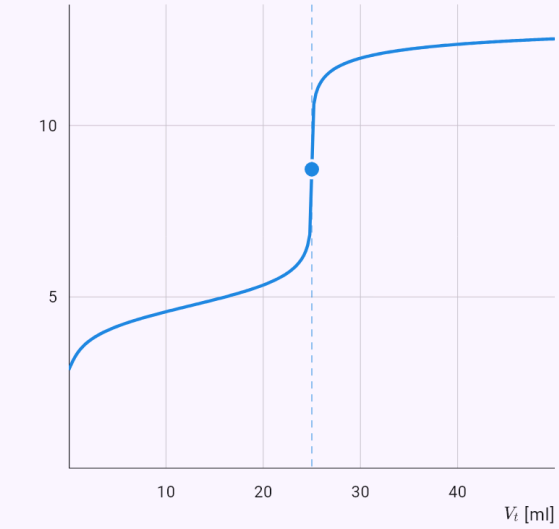
Result

$$\mathrm{pH} \approx 8.72$$

Steps

- Find pH; given
  $c_a = 0.1\,\mathrm{mol/l}, \quad V_a = 25\,\mathrm{ml}, \quad c_t = 0.1\,\mathrm{mol/l}, \quad K_a =$
- In a titration a titrant of known concentration is added from a burette until all of the analyte has reacted. The equivalence point comes when the titrant's amount $c_t \cdot V_t$ equals the analyte's amount: $V_e = c_a \cdot V_a / c_t$. Before it the pH is set by the unreacted analyte (a weak acid forms a buffer), at it by the salt formed reacting with water, after it by the excess titrant. The titration curve in the graph is computed exactly from the charge balance. The model's equation: $\mathrm{HA} + \mathrm{OH^-} \to \mathrm{A^-} + \mathrm{H_2O}$
- Equivalence point – as much titrant as analyte
  $n_a = c_a V_a = 2.5\,\mathrm{mmol}, \quad V_e = \frac{c_a V_a}{c_t} = \frac{2.5}{0.1} \approx 25\,\mathrm{ml}$
- Titrant added and the total volume:
  $n_t = c_t V_t = 2.5\,\mathrm{mmol}, \quad V = V_a + V_t = 50\,\mathrm{ml}$
- The equivalence point – all of the analyte has reacted.
- At the equivalence point only the salt formed is left – its reaction with water
  $c_{\mathrm{A^-}} = \frac{n_a}{V_a + V_e} = 0.05\,\mathrm{mol/l}, \quad K_b = \frac{K_w}{K_a} \approx 5.556 \cdot 10$
- pOH and pH (pH + pOH = 14)
  $\mathrm{pOH} = -\log\left[\mathrm{OH^-}\right] = -\log 5.27 \cdot 10^{-6} \approx 5.28, \quad \mathrm{pH}$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

Activation energy from a table of rate constants.

Input: `model(arrhenius; find=Ea[kJ_mol]; TT=25 35 45 55[degC]; kk=0.001 0.0021 0.0042`

`0.0079[per_s]; input=table)`

Result

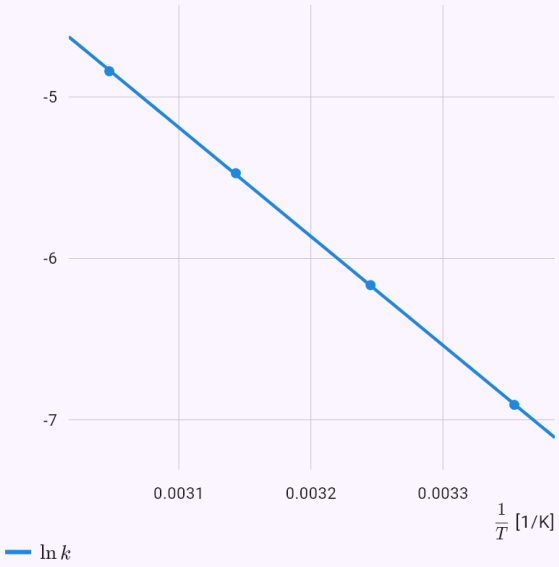
$$E_a \approx 56.1\,\mathrm{kJ/mol}$$

Steps

- Find E_a; given
  $T_i = (25;\ 35;\ 45;\ 55)\,^\circ\mathrm{C}, \quad k_i = (0.001;\ 0.0021;\ 0.004$
- Converting the units:
  $T_i = (298.1;\ 308.1;\ 318.1;\ 328.1)\,\mathrm{K}$
- A reaction's rate constant grows with temperature by the Arrhenius equation k = A·e^(−Ea/RT), where Ea is the activation energy – the energy barrier colliding molecules must overcome – and A the pre-exponential factor. Two measurements give $\ln(k_2/k_1) = (Ea/R)(1/T_1 - 1/T_2)$. From more measurements Ea comes from a regression line: ln k = ln A − (Ea/R)·(1/T) is a straight line in 1/T (the Arrhenius plot) with the slope −Ea/R. The model's equation:
  $k = A\,e^{-\frac{E_a}{RT}}, \quad R = 8.314\,\mathrm{J\,mol^{-1}\,K^{-1}}$
- Taking logarithms gives a straight line
  $\ln k = \ln A - \dfrac{E_a}{R}\cdot\dfrac{1}{T} \quad (y = a + bx,\ x = \frac{1}{T},\ y = \ln k)$
- Values for the regression line
  $x_i = (0.003354;\ 0.003245;\ 0.003143;\ 0.003047)\,\mathrm{K^{-1}},$
- Least squares regression line
  $b = \dfrac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \approx -6747, \quad a = \bar{y} - b\bar{x} \approx 1$
- Expressing the sought quantity: $E_a = -bR$
- Substituting:
  $E_a = -(-6747)\cdot 8.314\cdot 10^{-3} \approx 56.1\,\mathrm{kJ/mol}$

- Pre-exponential factor: $A = e^a \approx 6\,768\,000\,\mathrm{s^{-1}}$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

A sample's concentration from a calibration line.

Input: `model(beer; find=c[mmol_l]; cc=0 0.1 0.2 0.3 0.4[mmol_l]; AA=0.002 0.151 0.299 0.452`

`0.598[one]; A=0.25[one]; input=calibration; measure=absorbance)`

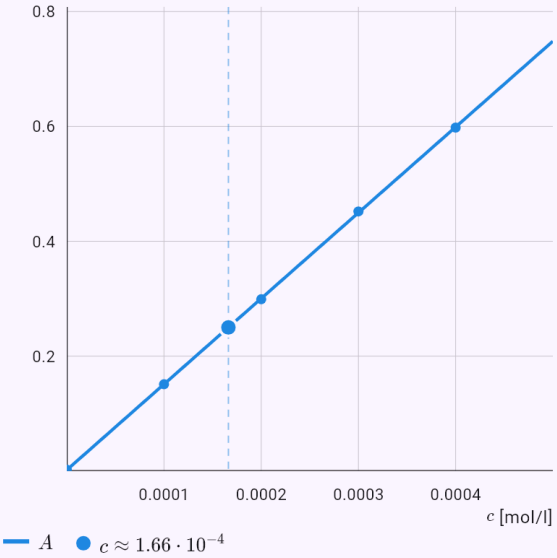
Result

$$c \approx 0.166\,\mathrm{mmol/l}$$

Steps

- Find c; given
  $c_i = (0;\ 0.1;\ 0.2;\ 0.3;\ 0.4)\,\mathrm{mmol/l}, \quad A_i = (0.002;\ 0.1$
- Converting the units
  $c_i = \left(0;\ 10^{-4};\ 2\cdot10^{-4};\ 3\cdot10^{-4};\ 4\cdot10^{-4}\right)\,\mathrm{mol/l}$
- A solution absorbs light in proportion to the absorbing substance's concentration and the layer's thickness: the absorbance A = ε·l·c (the Beer–Lambert law), where ε is the molar absorption coefficient and l the cuvette's length. A spectrophotometer measures the transmittance T – the share of light let through – and A = −log T. An unknown concentration comes from a known ε, from a comparison with a standard of known concentration, or from a calibration line measured on several standards (least squares). The law holds for dilute solutions. The model's equation: $A = \varepsilon\, l\, c$
- Least squares calibration line
  $A = a + b\,c: \quad b = \dfrac{n\sum c_i A_i - \sum c_i \sum A_i}{n\sum c_i^2 - \left(\sum c_i\right)^2} \approx 1493\,\mathrm{l/m}$
- Expressing the sought quantity: $c = \dfrac{A-a}{b}$
- Substituting: $c = \dfrac{0.25 - 0.0018}{1493} \approx 1.662\cdot10^{-4}\,\mathrm{mol/l}$
- Converting the units:
  $c \approx 1.662\cdot10^{-4}\,\mathrm{mol/l} = 0.1662\,\mathrm{mmol/l}$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

Carburizing steel – the time for 0.8 % carbon at the depth of 0.5 mm.

Input: `model(diffusion; find=t[h]; D=1.6e-11[m2_s]; cs=1.2[pct]; c0=0.25[pct]; cx=0.8[pct];`

`x=0.5[mm]; process=carburizing)`

Result

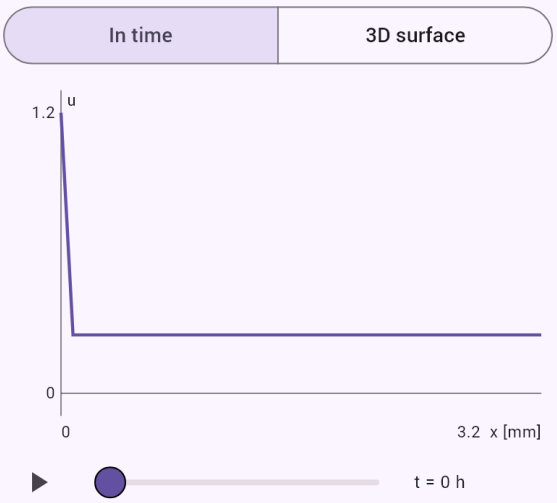
$$t \approx 7.047\,\text{h}$$

Steps

- Find t; given
  $D = 1.6 \cdot 10^{-11}\,\text{m}^2/\text{s}, \quad c_s = 1.2\,\%, \quad c_0 = 0.25\,\%, \quad c$
- Converting the units
  $c_s = 1.2\,\% = 0.012, \quad c_0 = 0.25\,\% = 0.0025, \quad c = 0.8$
- A substance diffuses from a surface held at the concentration c_s into a body with the initial concentration c_0 – e.g. carbon into steel in carburizing. The diffusion equation c_t = D·c_xx holds; for a body infinitely thick compared with the penetration depth it is solved by the Laplace transform in time: c = c_0 + (c_s − c_0)·erfc(x/(2√(Dt))) with the complementary error function. From it the concentration at a depth, the depth of a required concentration or the time needed follow. The model's equation
  $$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}, \quad c(0,t) = c_s, \quad c(x,0) = c_0$$
- Solution by the Laplace transform (a semi-infinite body):
  $$c(x,t) = c_0 + (c_s - c_0)\operatorname{erfc}\frac{x}{2\sqrt{Dt}}$$
- Inverse complementary error function
  $$\operatorname{erfc} z = \frac{c - c_0}{c_s - c_0} = \frac{0.008 - 0.0025}{0.012 - 0.0025} \approx 0.5789 \Rightarrow z \approx 0.$$
- Expressing the sought quantity: $t = \dfrac{x^2}{4Dz^2}$
- Substituting
  $$t = \frac{(5 \cdot 10^{-4})^2}{4 \cdot 1.6 \cdot 10^{-11} \cdot 0.3924^2} \approx 25\,370\,\text{s}$$

- The typical penetration depth is √(Dt):
  $\sqrt{Dt} \approx 6.371 \cdot 10^{-4}\,\text{m}$. Twice the depth takes four times the time.
- Converting the units: $t \approx 25\,370\,\text{s} = 7.047\,\text{h}$

Graph



The solution u at the time t chosen by the slider; the button plays it.

The carbon concentration at the depth of 0.5 mm after 7 hours.

Input: `model(diffusion; find=c[pct]; D=1.6e-11[m2_s]; cs=1.2[pct]; c0=0.25[pct]; x=0.5[mm];`

```
t=7[h]; process=carburizing)
```

Result

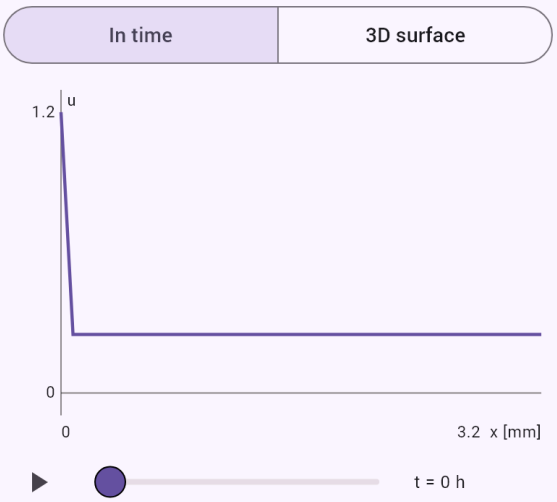
$c \approx 0.7988\,\%$

Steps

- Find c; given
  $D = 1.6 \cdot 10^{-11}\,\mathrm{m}^2/\mathrm{s}, \quad c_s = 1.2\,\%, \quad c_0 = 0.25\,\%, \quad x$
- Converting the units
  $c_s = 1.2\,\% = 0.012, \quad c_0 = 0.25\,\% = 0.0025, \quad x = 0.5$
- A substance diffuses from a surface held at the concentration c_s into a body with the initial concentration c_0 – e.g. carbon into steel in carburizing. The diffusion equation c_t = D·c_xx holds; for a body infinitely thick compared with the penetration depth it is solved by the Laplace transform in time: c = c_0 + (c_s − c_0)·erfc(x/(2√(Dt))) with the complementary error function. From it the concentration at a depth, the depth of a required concentration or the time needed follow. The model's equation
  $\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}, \quad c(0,t) = c_s, \quad c(x,0) = c_0$
- Solution by the Laplace transform (a semi-infinite body):
  $c(x,t) = c_0 + (c_s - c_0)\operatorname{erfc}\frac{x}{2\sqrt{Dt}}$
- Argument of the error function
  $z = \frac{x}{2\sqrt{Dt}} = \frac{5 \cdot 10^{-4}}{2\sqrt{1.6 \cdot 10^{-11} \cdot 25\,200}} \approx 0.3937, \quad \operatorname{erfc} z$
- Substituting:
  $c = 0.0025 + (0.012 - 0.0025) \cdot 0.5777 \approx 0.007988$

- The typical penetration depth is √(Dt):
  $\sqrt{Dt} \approx 6.35 \cdot 10^{-4}\,\mathrm{m}$. Twice the depth takes four times the time.
- Converting the units: $c \approx 0.007988 = 0.7988\,\%$

Graph



The solution u at the time t chosen by the slider; the button plays it.

Known limitations

- pH and titrations assume 25 °C (Kw = 10^{-14}) and dilute solutions – activity coefficients aren't computed.
- Titrations of a monoprotic acid or base by a strong titrant only; polyprotic acids so far only as the pH of a solution.
- A chemical equation isn't recognized from a photo yet – type it into the manual input or the form.

48. Electrical engineering

Electrical calculations are on the home screen in the section "Applications", field Electrical engineering: a series and a parallel RLC circuit with an alternating current, RC and RL filters, a resonant circuit, a voltage divider, a Wheatstone bridge and power factor correction. In the form you choose the connection and the quantity sought and fill in the values with units; the "?" icon explains where the formulas come from. Voltages and currents are RMS values, complex numbers are written with j as in electrical engineering.

For each circuit the app draws its diagram with the parts' values and, for an alternating current, the frequency response on a logarithmic axis – the impedance and the phase, for filters the gain in decibels (a Bode plot). The steps show the reactances, the impedance in the algebraic and polar form (50∠53.13° Ω), the phase shift, the currents, the parts' voltages and the power; they also explain why the parts' voltages add as phasors.

A series-parallel circuit can be entered as an expression, e.g. R1 + (L1 || (R2 + C1)) with the values R1=100 L1=10m C1=1u (prefixes m, u, n, k, M). The steps simplify the circuit group by group, then work out every part's voltage and current. Any circuit (with bridges and several sources too) is entered as a netlist as in SPICE – one part per line with its nodes, node 0 is the ground; the steps set up Kirchhoff's current law at the nodes, the system's matrix and solve it by Gaussian elimination. Without a frequency it's a direct current.

Examples

A series RLC circuit: real and reactive power, power factor.

Input: `model(acseries; find=P[W]; U=100[V]; R=30[ohm]; L=254.6[mH]; C=79.58[uF]; f=50[Hz];`

`parts=rlc)`

Result

$P \approx 120.1\,\mathrm{W}$
$Q \approx 160\,\mathrm{var}$
$S \approx 200\,\mathrm{VA}$
$\cos\varphi \approx 0.6$

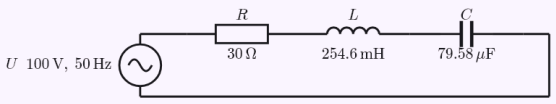
Steps

- Find P; given
  $U = 100\,\mathrm{V}, \quad R = 30\,\Omega, \quad L = 254.6\,\mathrm{mH}, \quad C = 79.5$
- Converting the units
  $L = 254.6\,\mathrm{mH} = 0.2546\,\mathrm{H}, \quad C = 79.58\,\mu\mathrm{F} = 7.958 \cdot 1$
- In an AC circuit, an inductor and a capacitor oppose the current with a frequency-dependent resistance – the reactance: the inductive X_L = ωL grows with the frequency, the capacitive X_C = 1/(ωC) falls (ω = 2πf). The inductor's voltage leads the current by 90°, the capacitor's lags by 90°, so they don't add arithmetically but as phasors – complex numbers with the imaginary unit j: the impedance Z = R + j(X_L − X_C), its magnitude Z = √(R² + (X_L − X_C)²) and the phase shift φ = arctan((X_L − X_C)/R). The current I = U/Z, the real power P = UI·cos φ. Voltages and currents are RMS values. The model's equation: $\hat{Z} = R + \mathrm{j}(X_L - X_C)$
- Angular frequency: $\omega = 2\pi f = 2\pi \cdot 50 \approx 314.2\,\mathrm{rad/s}$
- The inductor's inductive reactance:
  $X_L = \omega L = 314.2 \cdot 0.2546 \approx 79.98\,\Omega$
- The capacitor's capacitive reactance:
  $X_C = \dfrac{1}{\omega C} = \dfrac{1}{314.2 \cdot 7.958 \cdot 10^{-5}} \approx 40\,\Omega$
- The impedance's magnitude – the reactance and the resistance are perpendicular phasors
  $Z = \sqrt{R^2 + (X_L - X_C)^2} = \sqrt{30^2 + (79.98 - 40)^2} \approx 4$
- The impedance as a complex number (algebraic and polar form): $\hat{Z} \approx 30 + \mathrm{j}\,39.99\,\Omega = 49.99\angle 53.12^\circ\,\Omega$
- The current's phase shift behind the voltage:
  $\varphi = \arctan\dfrac{X_L - X_C}{R} \approx 53.12^\circ$
- The circuit is inductive – the current lags behind the voltage.
- Resonant frequency: $f_0 = \dfrac{1}{2\pi\sqrt{LC}} \approx 35.36\,\mathrm{Hz}$
- The current from Ohm's law: $I = \dfrac{U}{Z} = \dfrac{100}{49.99} \approx 2\,\mathrm{A}$
- Real, reactive and apparent power
  $P = UI\cos\varphi \approx 120.1\,\mathrm{W}$

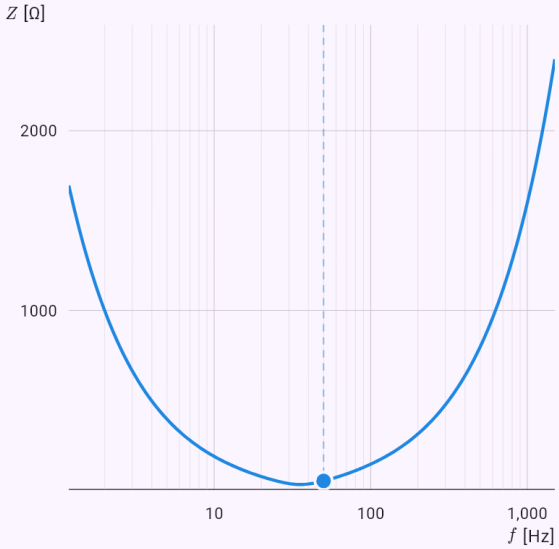
$Q = UI\sin\varphi \approx 160\,\mathrm{var}$
$S = UI \approx 200\,\mathrm{VA}$

- Power factor: $\cos\varphi = \dfrac{P}{S} \approx 0.6$

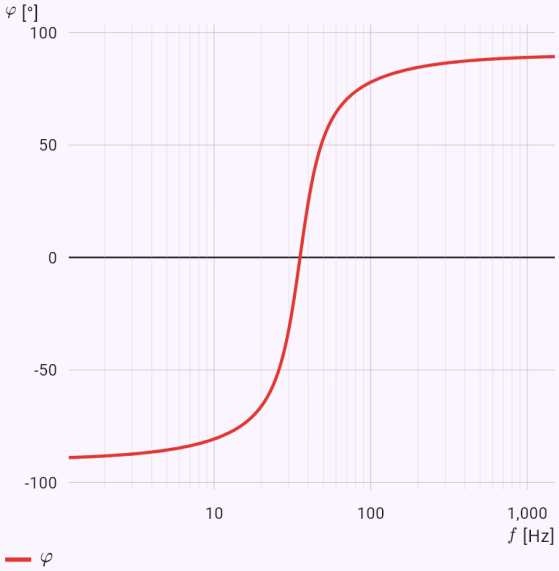
Circuit diagram



Frequency response



Z $Z \approx 50\,\Omega$



φ

The horizontal axis is logarithmic – each division is ten times the frequency. The dots are the given frequency and notable points.

A parallel RLC circuit: the branch currents and the total current.

Input: `model(acparallel; find=parts[mA]; U=10[V]; R=100[ohm]; L=159.2[mH]; C=31.83[uF];`

f=50[Hz]; parts=rlc)

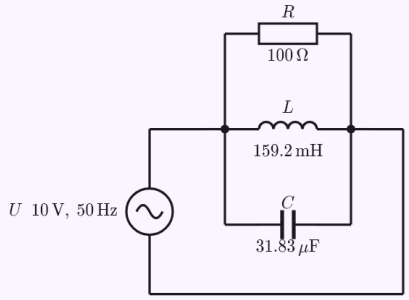
Result

$$I_R \approx 100\,\text{mA}$$
$$I_L \approx 199.9\,\text{mA}$$
$$I_C \approx 100\,\text{mA}$$
$$I \approx 141.4\,\text{mA}$$

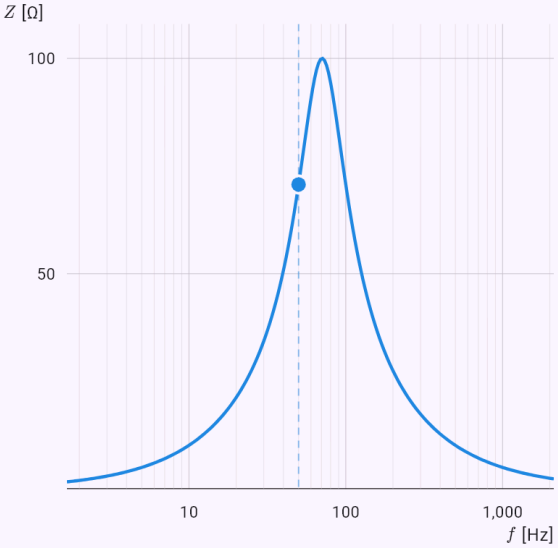
Steps

- Find I; given
  $U = 10\,\text{V}, \quad R = 100\,\Omega, \quad L = 159.2\,\text{mH}, \quad C = 31.8$
- Converting the units
  $L = 159.2\,\text{mH} = 0.1592\,\text{H}, \quad C = 31.83\,\mu\text{F} = 3.183 \cdot 1$
- In a parallel circuit all branches have the same voltage and the branch currents add as phasors. So instead of the impedances their reciprocals add – the admittances: Y = 1/R + j(ωC − 1/(ωL)). The resistor's current is in phase with the voltage, the inductor's lags by 90°, the capacitor's leads by 90°. The total current I = √(I_R² + (I_C − I_L)²) may be smaller than the branch currents – at resonance the inductor's and the capacitor's currents cancel. The model's equation
  $$\hat{Y} = \frac{1}{\hat{Z}} = \frac{1}{R} + \text{j}\left(\frac{1}{X_C} - \frac{1}{X_L}\right)$$
- Angular frequency: $\omega = 2\pi f = 2\pi \cdot 50 \approx 314.2\,\text{rad/s}$
- The inductor's inductive reactance:
  $X_L = \omega L = 314.2 \cdot 0.1592 \approx 50.01\,\Omega$
- The capacitor's capacitive reactance:
  $X_C = \frac{1}{\omega C} = \frac{1}{314.2 \cdot 3.183 \cdot 10^{-5}} \approx 100\,\Omega$
- Admittance – the branches' reciprocal impedances add as phasors
  $Y = \sqrt{\left(\frac{1}{100}\right)^2 + \left(\frac{1}{100} - \frac{1}{50.01}\right)^2} \approx 0.01414\,\text{S}, \quad \hat{Y}$
- The circuit's impedance: $Z = \frac{1}{Y} \approx 70.73\,\Omega$
- The impedance as a complex number (algebraic and polar form): $\hat{Z} \approx 50.03 + \text{j}\,50\,\Omega = 70.73\angle 44.98^\circ\,\Omega$
- The current's phase shift behind the voltage
  $\varphi = \arctan\left(R\left(\frac{1}{X_L} - \frac{1}{X_C}\right)\right) \approx 44.98^\circ$
- The circuit is inductive – the current lags behind the voltage.
- Resonant frequency: $f_0 = \frac{1}{2\pi\sqrt{LC}} \approx 70.7\,\text{Hz}$
- Branch currents – all branches have the same voltage
  $I_R = \frac{U}{R} = \frac{10}{100} \approx 0.1\,\text{A}$
  $I_L = \frac{U}{X_L} = \frac{10}{50.01} \approx 0.1999\,\text{A}$
  $I_C = \frac{U}{X_C} = \frac{10}{100} \approx 0.1\,\text{A}$
- The total current – the phasor sum of the branch currents: $I = \sqrt{I_R^2 + (I_C - I_L)^2} \approx 0.1414\,\text{A}$
- The branch currents add as phasors, not arithmetically – the inductor's and the capacitor's currents are in antiphase.

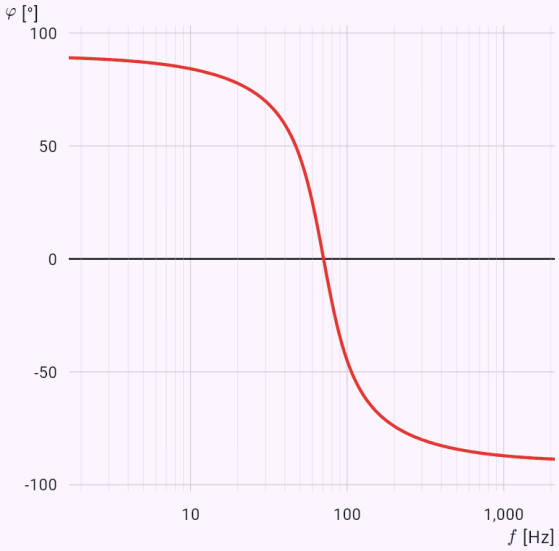
Circuit diagram



Frequency response



Z $Z \approx 70.7\,\Omega$



φ

The horizontal axis is logarithmic – each division is ten times the frequency. The dots are the given frequency and notable points.

An RC low-pass: the gain in decibels and the Bode plot.

Input: `model(filter; find=gain[one]; R=1[kohm]; C=100[nF]; f=1[kHz]; U1=5[V]; kind=rcLow)`

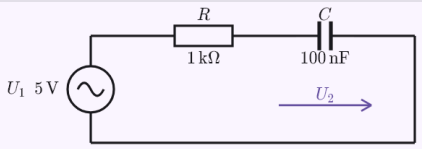
Result

$$|\hat{H}| \approx 0.8467$$
$$A \approx -1.45\,\text{dB}$$
$$\varphi \approx -32.14^\circ$$
$$U_2 \approx 4.234\,\text{V}$$

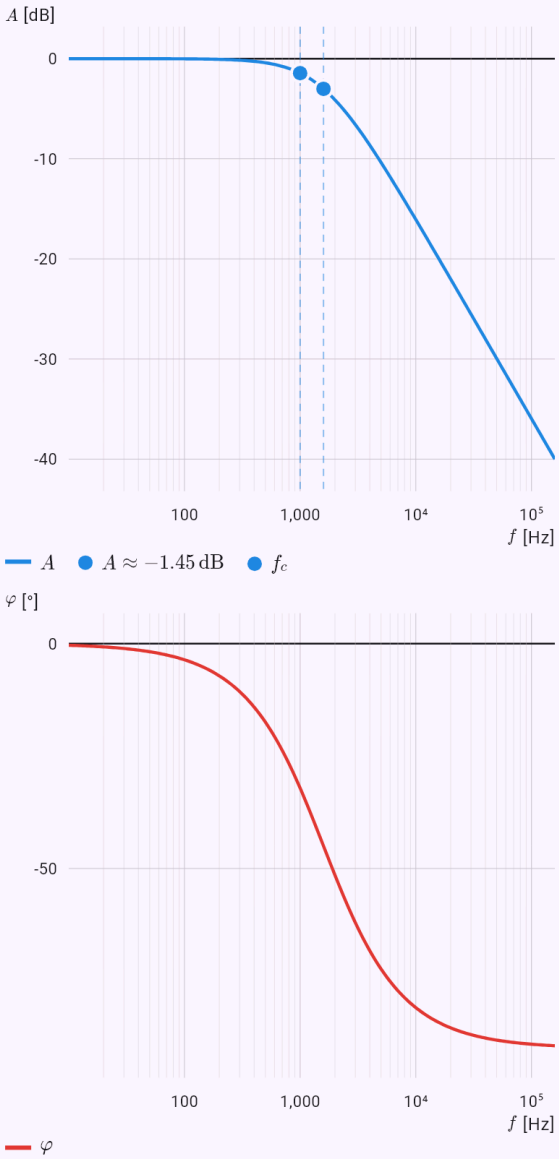
Steps

- Given
  $R = 1\,\text{k}\Omega, \quad C = 100\,\text{nF}, \quad f = 1\,\text{kHz}, \quad U_1 = 5\,\text{V}$
- Converting the units
  $R = 1\,\text{k}\Omega = 1000\,\Omega, \quad C = 100\,\text{nF} = 10^{-7}\,\text{F}, \quad f = 1$
- A first-order filter is a voltage divider of a resistor and a capacitor (or an inductor). A low-pass (RC with the output across the capacitor) passes low frequencies and suppresses high ones, a high-pass the other way round. The cutoff frequency f_c = 1/(2πRC), for RL f_c = R/(2πL): there the output voltage falls to 1/√2 ≈ 70.7 % (by 3 dB) and the phase shift is 45°. The transfer H = U_2/U_1 is a complex number; its magnitude in decibels A = 20·log|H| and its phase make up the Bode plot – above the cutoff a low-pass falls by 20 dB per decade. The model's equation
  $$f_c = \frac{1}{2\pi RC}, \quad \hat{H} = \frac{\hat{U}_2}{\hat{U}_1} = \frac{1}{1 + \mathrm{j}\frac{f}{f_c}}$$
- A low-pass: low frequencies pass, high ones are suppressed.
- Cutoff frequency:
  $$f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 1000 \cdot 10^{-7}} \approx 1592\,\text{Hz}$$
- At the cutoff frequency the gain is 1/√2 ≈ 0.707 (−3 dB) and the phase shift 45°.
- The frequency relative to the cutoff:
  $$\frac{f}{f_c} = \frac{1000}{1592} \approx 0.6283$$
- The gain's magnitude:
  $$|\hat{H}| = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^2}} \approx 0.8467$$
- The gain in decibels: $A = 20 \log |\hat{H}| \approx -1.45\,\text{dB}$
- The output's phase shift against the input:
  $$\varphi = -\arctan \frac{f}{f_c} \approx -32.14^\circ$$
- Output voltage: $U_2 = |\hat{H}|\, U_1 \approx 0.8467 \cdot 5 \approx 4.234\,\text{V}$

Circuit diagram



Frequency response



The horizontal axis is logarithmic – each division is ten times the frequency. The dots are the given frequency and notable points.

A series resonant circuit: quality factor and bandwidth.

Input: `model(resonance; find=Q[one]; L=10[mH]; C=100[nF]; R=10[ohm]; kind=series)`

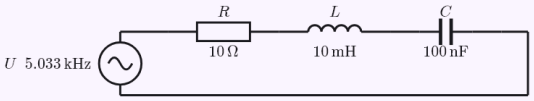
Result

$$Q \approx 31.62$$
$$B \approx 159.2\,\text{Hz}$$
$$f_1 \approx 4954\,\text{Hz}$$
$$f_2 \approx 5113\,\text{Hz}$$

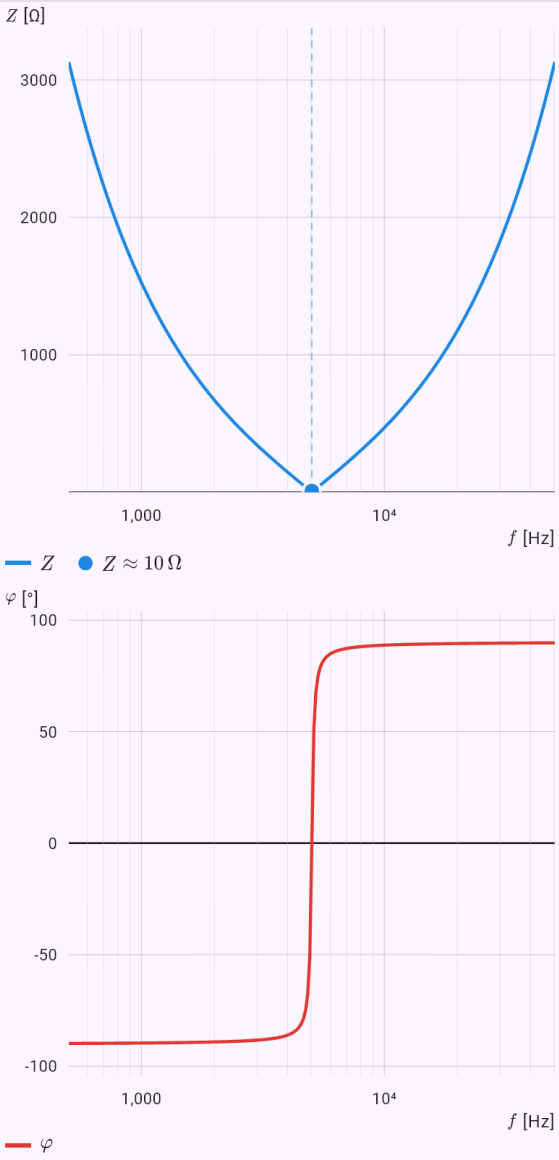
Steps

- Given: $L = 10\,\text{mH}, \quad C = 100\,\text{nF}, \quad R = 10\,\Omega$
- Converting the units:
  $L = 10\,\text{mH} = 0.01\,\text{H}, \quad C = 100\,\text{nF} = 10^{-7}\,\text{F}$
- At the resonant frequency $f_0 = 1/(2\pi\sqrt{LC})$ the inductor's and the capacitor's reactances are equal and cancel. A series circuit then has the lowest impedance (R alone) and the largest current, a parallel one the highest impedance. The sharpness of the resonance is the quality factor Q – for a series circuit $Q = (1/R)\sqrt{L/C}$ – and the bandwidth $B = f_0/Q$ between the frequencies where the current falls to $1/\sqrt{2}$ of its maximum. Tuning: the capacitance for a given frequency $C = 1/(4\pi^2 f_0^2 L)$.
  The model's equation: $f_0 = \dfrac{1}{2\pi\sqrt{LC}}, \quad Q = \dfrac{1}{R}\sqrt{\dfrac{L}{C}}$
- Substituting: $f_0 = \dfrac{1}{2\pi\sqrt{0.01 \cdot 10^{-7}}} \approx 5033\,\text{Hz}$
- At resonance the inductor's and the capacitor's reactances cancel – a series circuit has the lowest impedance and the largest current.
- Quality factor: $Q = \dfrac{1}{R}\sqrt{\dfrac{L}{C}} = \dfrac{1}{10}\sqrt{\dfrac{0.01}{10^{-7}}} \approx 31.62$
- Bandwidth: $B = \dfrac{f_0}{Q} = \dfrac{5033}{31.62} \approx 159.2\,\text{Hz}$
- The band's edges (a fall to $1/\sqrt{2}$)
  $f_{1,2} = f_0\left(\sqrt{1 + \dfrac{1}{4Q^2}} \mp \dfrac{1}{2Q}\right)$: $\quad f_1 \approx 4954\,\text{Hz}, \quad f_2 \approx$

Circuit diagram



Frequency response



The horizontal axis is logarithmic – each division is ten times the frequency. The dots are the given frequency and notable points.

A loaded voltage divider.

Input: `model(divider; find=U2[V]; U=12[V]; R1=1[kohm]; R2=2[kohm]; RL=2[kohm]; load=load)`

Result

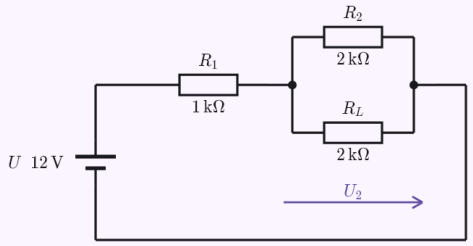
$$U_2 \approx 6\,\text{V}$$

Steps

- Find U_2; given
  $U = 12\,\text{V}, \quad R_1 = 1\,\text{k}\Omega, \quad R_2 = 2\,\text{k}\Omega, \quad R_L = 2\,\text{k}\Omega$
- Converting the units
  $R_1 = 1\,\text{k}\Omega = 1000\,\Omega, \quad R_2 = 2\,\text{k}\Omega = 2000\,\Omega, \quad R_L = 2$
- Two resistors in series divide the voltage in the ratio of their resistances: the same current I = U/(R₁ + R₂) flows through both, so U₂ = U·R₂/(R₁ + R₂). A load R_L is in parallel with R₂ and lowers the lower part's resistance to R₂R_L/(R₂ + R_L) – a loaded divider's output voltage is lower than with no load. The model's equation:
  $U_2 = U\dfrac{R_{2L}}{R_1 + R_{2L}}, \quad R_{2L} = \dfrac{R_2 R_L}{R_2 + R_L}$
- The load is in parallel with R₂
  $R_{2L} = \dfrac{R_2 R_L}{R_2 + R_L} = \dfrac{2000 \cdot 2000}{2000 + 2000} \approx 1000\,\Omega$
- Substituting: $U_2 = 12 \cdot \dfrac{1000}{1000 + 1000} \approx 6\,\text{V}$

- With no load it would be $U_{2,0} = U\dfrac{R_2}{R_1 + R_2} \approx 8\,\text{V}$ – the load lowers the output voltage.

Circuit diagram



Balancing a Wheatstone bridge.

Input: `model(bridge; find=R4[ohm]; R1=100[ohm]; R2=200[ohm]; R3=150[ohm])`

Result

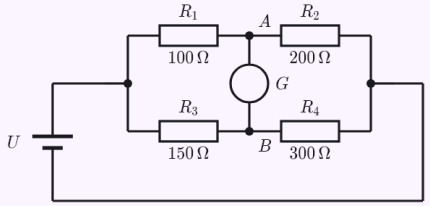
$$R_4 \approx 300\,\Omega$$

Steps

- Find R_4; given:
  $R_1 = 100\,\Omega, \quad R_2 = 200\,\Omega, \quad R_3 = 150\,\Omega$
- A Wheatstone bridge is two voltage dividers in parallel with a meter between their midpoints (A and B). The bridge is balanced – no current through the meter – when both dividers divide the voltage in the same ratio: R₁/R₂ = R₃/R₄. That's how an unknown resistance is measured precisely. An unbalanced bridge gives the voltage U_AB = U(R₂/(R₁ + R₂) − R₄/(R₃ + R₄)) – that's how strain gauges work, for example. The model's equation:
  $\dfrac{R_1}{R_2} = \dfrac{R_3}{R_4}$
- The bridge is balanced – A and B have the same voltage and no current flows through the meter.
- Expressing the sought quantity: $R_4 = \dfrac{R_2 R_3}{R_1}$

- Substituting: $R_4 = \dfrac{200 \cdot 150}{100} \approx 300\,\Omega$

Circuit diagram



The capacitor to correct the power factor from 0.7 to 0.95.

Input: `model(compensation; find=C[uF]; P=10[kW]; U=230[V]; f=50[Hz]; cos1=0.7[one]; cos2=0.95[one])`

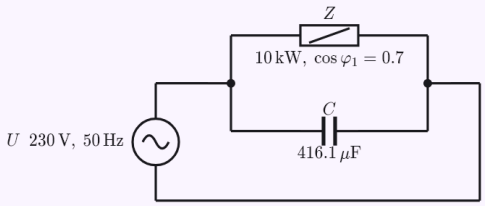
Result

$$C \approx 416.1\,\mu\text{F}$$

Steps

- Find C; given
  $P = 10\,\text{kW}, \quad U = 230\,\text{V}, \quad f = 50\,\text{Hz}, \quad \cos\varphi_1 = 0.7$
- Converting the units: $P = 10\,\text{kW} = 10\,000\,\text{W}$
- Loads with coils (motors, transformers) draw besides the real power P a reactive power Q = P·tan φ, which needlessly loads the line with a larger current. A capacitor in parallel supplies reactive power of the opposite sign Q_C = $Q_1 - Q_2$ and raises the power factor from cos φ_1 to cos φ_2. The capacitance needed C = Q_C/(ωU²), where ω = 2πf. The real power doesn't change, the current from the mains falls. The model's equation:
  $Q_C = P(\tan\varphi_1 - \tan\varphi_2), \quad C = \dfrac{Q_C}{\omega U^2}$
- Phase shifts before and after the correction
  $\varphi_1 = \arccos 0.7 \approx 45.57^\circ, \quad \varphi_2 = \arccos 0.95 \approx 18.19^\circ$
- Reactive powers – the capacitor supplies the difference
  $Q_1 = P\tan\varphi_1 \approx 10\,200\,\text{var}$
  $Q_2 = P\tan\varphi_2 \approx 3287\,\text{var}$
  $Q_C = Q_1 - Q_2 \approx 6915\,\text{var}$
- Angular frequency: $\omega = 2\pi f \approx 314.2\,\text{rad/s}$
- Expressing the sought quantity: $C = \dfrac{Q_C}{\omega U^2}$
- Substituting: $C = \dfrac{6915}{314.2 \cdot 230^2} \approx 4.161 \cdot 10^{-4}\,\text{F}$
- The current from the mains before and after the correction
  $I_1 = \dfrac{P}{U\cos\varphi_1} \approx 62.11\,\text{A}, \quad I_2 = \dfrac{P}{U\cos\varphi_2} \approx 45.77\,\text{A}$
- Converting the units: $C \approx 4.161 \cdot 10^{-4}\,\text{F} = 416.1\,\mu\text{F}$

Circuit diagram



A circuit as an expression – simplified, every part's current.

Input: `circuit(R1 + (L1 || C1); R1=10; L1=10m; C1=100u; U=10V; f=50Hz)`

Result

$$Z \approx 10.59\angle 19.22^\circ\,\Omega$$
$$I \approx 0.9443\,\mathrm{A}$$
$$\varphi \approx 19.22^\circ$$
$$P \approx 8.917\,\mathrm{W}$$

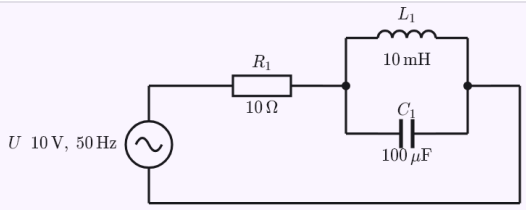
Steps

- The connection (+ in series, ∥ in parallel): $R_1 + (L_1 \parallel C_1)$; the given values $R_1 = 10\,\Omega, \quad L_1 = 10\,\mathrm{mH}, \quad C_1 = 100\,\mu\mathrm{F}, \quad U = 10\,\mathrm{V}$
- Angular frequency: $\omega = 2\pi f = 2\pi \cdot 50 \approx 314.2\,\mathrm{rad/s}$
- The impedances of the inductors and capacitors as complex numbers (j is the imaginary unit)
  $\hat{Z}_{L_1} = \mathrm{j}\omega L_1 = \mathrm{j} \cdot 314.2 \cdot 0.01 \approx \mathrm{j}\,3.142\,\Omega$
  $\hat{Z}_{C_1} = \dfrac{1}{\mathrm{j}\omega C_1} = -\mathrm{j}\dfrac{1}{314.2 \cdot 10^{-4}} \approx -\mathrm{j}\,31.83\,\Omega$
- A parallel group – the reciprocals of the impedances add
  $\hat{Z}_a = \dfrac{\hat{Z}_{L_1} \cdot \hat{Z}_{C_1}}{\hat{Z}_{L_1} + \hat{Z}_{C_1}} \approx \mathrm{j}\,3.486\,\Omega$
- A series group – the impedances add: $\hat{Z} = R_1 + \hat{Z}_a$
- The circuit's total impedance:
  $\hat{Z} = 10 + \mathrm{j}\,3.486\,\Omega = 10.59\angle 19.22^\circ\,\Omega$
- The source's current by Ohm's law (the source's voltage has the phase 0°)
  $\hat{I} = \dfrac{\hat{U}}{\hat{Z}} = \dfrac{10}{10.59\angle 19.22^\circ} \approx 0.9443\angle -19.22^\circ\,\mathrm{A}$
- The same current flows through all parts of the group $\hat{Z}$ – their voltages
  $\hat{U}_{R_1} = \hat{I} R_1 \approx 9.443\angle -19.22^\circ\,\mathrm{V}$
  $\hat{U}_a = \hat{I}\hat{Z}_a \approx 3.291\angle 70.78^\circ\,\mathrm{V}$
- All branches of the group $\hat{Z}_a$ have the same voltage – the branch currents
  $\hat{I}_{L_1} = \dfrac{\hat{U}_a}{\hat{Z}_{L_1}} \approx 1.048\angle -19.22^\circ\,\mathrm{A}$
  $\hat{I}_{C_1} = \dfrac{\hat{U}_a}{\hat{Z}_{C_1}} \approx 0.1034\angle 160.8^\circ\,\mathrm{A}$
- The parts – impedance, voltage and current (magnitude ∠ phase)

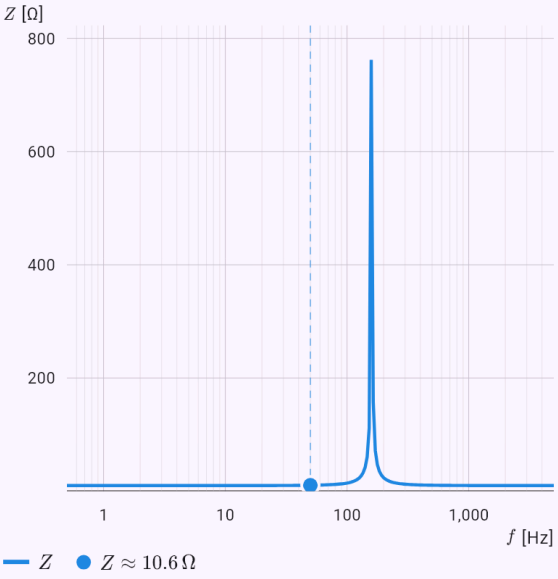
	$\hat{Z}\,[\Omega]$	$\hat{U}\,[\mathrm{V}]$	$\hat{I}\,[\mathrm{A}]$
R_1	10	$9.443\angle -19.22^\circ$	$0.9443\angle -19.22^\circ$
L_1	$\mathrm{j}\,3.142$	$3.291\angle 70.78^\circ$	$1.048\angle -19.22^\circ$
C_1	$-\mathrm{j}\,31.83$	$3.291\angle 70.78^\circ$	$0.1034\angle 160.8^\circ$

- The power supplied by the source
  $\hat{S} = \hat{U}\hat{I}^* \approx 8.917 + \mathrm{j}\,3.108, \quad P \approx 8.917\,\mathrm{W}, \quad Q \approx 3.1$

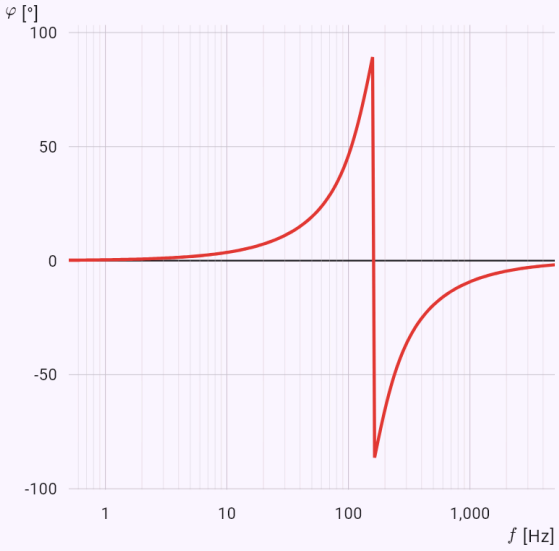
Circuit diagram



Frequency response



Z $Z \approx 10.6\,\Omega$



φ

The horizontal axis is logarithmic – each division is ten times the frequency. The dots are the given frequency and notable points.

An unbalanced bridge as a netlist – nodal analysis.

Input: `netlist(V1 1 0 12; R1 1 2 100; R2 2 0 200; R3 1 3 150; R4 3 0 250; R5 2 3 50)`

Result

$U_1 \approx 12\,\mathrm{V}$
$U_2 \approx 7.842\,\mathrm{V}$
$U_3 \approx 7.723\,\mathrm{V}$

Steps

- The parts, the nodes they connect and their values

	+	−	
V_1	1	0	$12\,\mathrm{V}$
R_1	1	2	$100\,\Omega$
R_2	2	0	$200\,\Omega$
R_3	1	3	$150\,\Omega$
R_4	3	0	$250\,\Omega$
R_5	2	3	$50\,\Omega$

- The unknowns are the nodes' voltages against the ground and the voltage sources' currents: $U_1,\ U_2,\ U_3,\ I_{V_1}$
- Kirchhoff's current law – the currents leaving each node add up to zero

$$1\text{: } I_{V_1} + \frac{U_1 - U_2}{R_1} + \frac{U_1 - U_3}{R_3} = 0$$
$$2\text{: } \frac{U_2 - U_1}{R_1} + \frac{U_2}{R_2} + \frac{U_2 - U_3}{R_5} = 0$$
$$3\text{: } \frac{U_3 - U_1}{R_3} + \frac{U_3}{R_4} + \frac{U_3 - U_2}{R_5} = 0$$

- The voltage sources set the voltage between their nodes: $U_1 = 12$
- The system of linear equations as a matrix for the unknowns $(U_1,\ U_2,\ U_3,\ I_{V_1})$

$$\left(\begin{array}{cccc|c} 0.01667 & -0.01 & -0.006667 & 1 & 0 \\ -0.01 & 0.035 & -0.02 & 0 & 0 \\ -0.006667 & -0.02 & 0.03067 & 0 & 0 \\ 1 & 0 & 0 & 0 & 12 \end{array}\right)$$

- Gaussian elimination gives the nodes' voltages
  $U_1 \approx 12\,\mathrm{V}$
  $U_2 \approx 7.842\,\mathrm{V}$
  $U_3 \approx 7.723\,\mathrm{V}$
- Each part's voltage and current (from node + to node −)

	$U\,[\mathrm{V}]$	$I\,[\mathrm{A}]$
V_1	12	0.0701
R_1	4.158	0.04158
R_2	7.842	0.03921
R_3	4.277	0.02851
R_4	7.723	0.03089
R_5	0.1188	0.002376

Known limitations

- The circuits are linear and in the steady state – the transients of RC, RL and RLC are the physics applications; diodes, transistors and operational amplifiers aren't there yet.
- The diagram is drawn for the forms and for a circuit as an expression; a netlist gets a table of parts and nodes. A circuit isn't recognized from a photo of its diagram.
- A netlist is solved either at a direct current or at one frequency – sources of different frequencies and superposition aren't there yet.

49. Biology and medicine

Calculations from biology, the lab and medicine are on the home screen in the section "Applications", fields Biology and Medicine: enzyme kinetics by Michaelis and Menten (with an inhibitor too), the Hardy–Weinberg equilibrium, dilutions and colony counts, the membrane potential by Nernst and Goldman, and drug dosing. Fill in the form as for the other applications; the "?" icon explains where the formulas come from.

From measured rates of an enzyme reaction it determines K_m and V_max twice: the school way from the Lineweaver–Burk line and more precisely by nonlinear regression; the graph shows the points and both curves. For the Hardy–Weinberg equilibrium it computes the allele frequencies, the expected counts and a χ^2 test from the genotypes' counts, for a recessive disease the share of carriers from its incidence.

Dilutions: the stock solution's volume by $c_1V_1 = c_2V_2$, a serial dilution and CFU/ml from a colony count. Dosing: a dose by body mass or body surface (Mosteller, Du Bois), an infusion's rate in ml/h and drops per minute and the creatinine clearance by Cockcroft and Gault. The dose calculations are only a guide and no substitute for a clinician's judgement.

Examples

K_m and V_max from measured rates – Lineweaver–Burk and nonlinear regression.

Input: `model(enzyme; find=params[one]; SS=0.5 1 2 4 8[mmol_l]; vv=22 31 52 64`

`81[umol_l_min]; input=table)`

Result

$V_{max} \approx 99.85\,\mu\text{mol}/(\text{l}\cdot\text{min})$

$K_m \approx 2.005\,\text{mmol/l}$

Steps

- Given

  $[\text{S}]_i = (0.5;\ 1;\ 2;\ 4;\ 8)\,\text{mmol/l}, \quad v_i = (22;\ 31;\ 52;\ 64$

- An enzyme E binds its substrate S and turns it into the product. The rate v grows with the substrate's concentration but saturates – at most it reaches the limiting rate V_max, when all enzyme molecules are occupied. The Michaelis constant K_m is the substrate's concentration at half that rate: v = V_max[S]/(K_m + [S]). From measured rates the constants come from the Lineweaver–Burk line (1/v against 1/[S], intercept 1/V_max, slope K_m/V_max) or more precisely by nonlinear regression. An inhibitor: a competitive one raises the apparent K_m, a noncompetitive one lowers V_max, an uncompetitive one lowers both – all by the factor α = 1 + [I]/K_i. The model's equation:

  $$v = \frac{V_{max}[\text{S}]}{K_m + [\text{S}]}$$

- Reciprocals for the Lineweaver–Burk plot

  $\frac{1}{[\text{S}]_i} \approx (2;\ 1;\ 0.5;\ 0.25;\ 0.125), \quad \frac{1}{v_i} \approx (0.04545;\ 0.03$

- Regression line of 1/v against 1/[S]

  $\frac{1}{v} = \frac{1}{V_{max}} + \frac{K_m}{V_{max}} \cdot \frac{1}{[\text{S}]}: \quad a \approx 0.01117, \quad b \approx 0.0178$

- The constants from the line (Lineweaver–Burk)

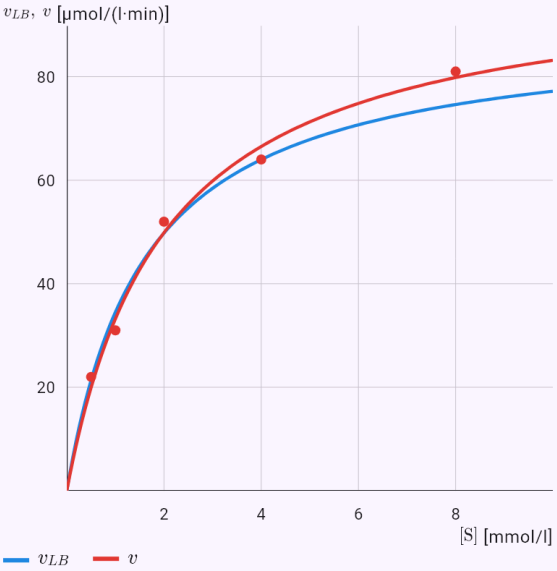
  $V_{max} = \frac{1}{a} \approx 89.49\,\mu\text{mol}/(\text{l}\cdot\text{min}), \quad K_m = \frac{b}{a} \approx 1.594$

- Nonlinear regression – least squares on the curve itself

  $\min \sum \left(v_i - \frac{V_{max}[\text{S}]_i}{K_m + [\text{S}]_i} \right)^2: \quad V_{max} \approx 99.85\,\mu\text{mol}/(\text{l}$

- The Lineweaver–Burk plot magnifies the errors at small concentrations (large 1/v) – so the result is the nonlinear regression's; the line is the school method and the starting estimate.

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

The reaction rate with a competitive inhibitor.

Input: `model(enzyme; find=v[umol_l_min]; Vmax=100[umol_l_min]; Km=2[mmol_l]; S=2[mmol_l];`

```
I=1[mmol_l]; Ki=1[mmol_l]; input=constants; inhibition=competitive)
```

Result

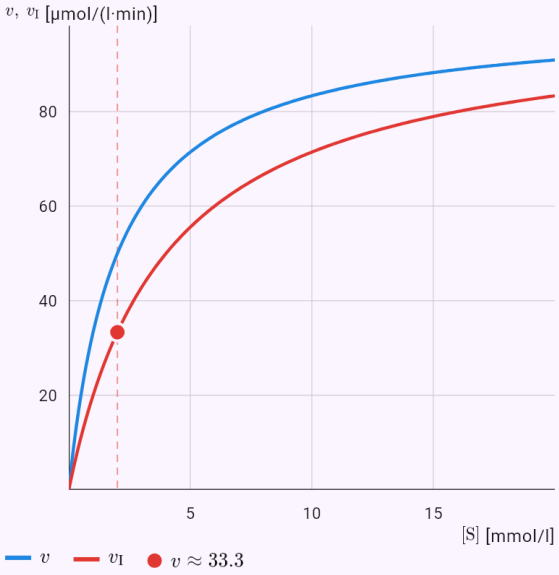
$$v \approx 33.33\,\mu\text{mol}/(\text{l}\cdot\text{min})$$

Steps

- Find v; given $V_{max} = 100\,\mu\text{mol}/(\text{l}\cdot\text{min}), \quad K_m = 2\,\text{mmol/l}, \quad [\text{S}] =$
- An enzyme E binds its substrate S and turns it into the product. The rate v grows with the substrate's concentration but saturates – at most it reaches the limiting rate V_max, when all enzyme molecules are occupied. The Michaelis constant K_m is the substrate's concentration at half that rate: v = V_max[S]/(K_m + [S]). From measured rates the constants come from the Lineweaver–Burk line (1/v against 1/[S], intercept 1/V_max, slope K_m/V_max) or more precisely by nonlinear regression. An inhibitor: a competitive one raises the apparent K_m, a noncompetitive one lowers V_max, an uncompetitive one lowers both – all by the factor α = 1 + [I]/K_i. The model's equation:
$$v = \frac{V_{max}[\text{S}]}{K_m + [\text{S}]}$$
- Inhibition factor: $\alpha = 1 + \frac{[\text{I}]}{K_i} = 1 + \frac{1}{1} \approx 2$
- Apparent constants with the inhibitor $K_m^{app} = \alpha K_m \approx 4\,\text{mmol/l}, \quad V_{max}^{app} = V_{max} = 100\,\mu\text{mol}/$
- A competitive inhibitor competes with the substrate for the active site – it raises the apparent K_m, V_max stays (a lot of substrate outcompetes it).
- Substituting
$$v = \frac{100 \cdot 2}{4 + 2} \approx 33.33\,\mu\text{mol}/(\text{l}\cdot\text{min})$$

- The enzyme's saturation: $\frac{v}{V_{max}} \approx 33.3\,\%$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

MN blood groups – allele frequencies and the χ² test of equilibrium.

Input: `model(hardyweinberg; find=frequencies[one]; AA=1787[one]; Aa=3039[one]; aa=1303[one];`

`input=counts)`

Result

$p \approx 0.5395$

$q \approx 0.4605$

$\chi^2 \approx 0.02698$

$p_{\chi^2} \approx 0.87$

Steps

- Given: $n_{AA} = 1787, \quad n_{Aa} = 3039, \quad n_{aa} = 1303$
- In a population where individuals mate at random and there's no selection, mutation or migration, the alleles' frequencies stay constant and the genotypes are in the ratio p² : 2pq : q², where p and q (p + q = 1) are the frequencies of the alleles A and a. From the genotypes' counts we compute p = (2n_AA + n_Aa)/(2n) and the expected counts; a χ² test (1 degree of freedom) shows whether the population departs from equilibrium. For a recessive trait we only know the share of homozygotes q², so q = √q² and the share of carriers 2pq. The model's equation: $p + q = 1, \quad p^2 + 2pq + q^2 = 1$
- Allele frequencies – each individual carries two
  $p = \frac{2n_{AA} + n_{Aa}}{2n} = \frac{2 \cdot 1787 + 3039}{2 \cdot 6129} \approx 0.5395, \quad q = 1$
- Expected counts at equilibrium
  $E_{AA} = np^2 \approx 1784, \quad E_{Aa} = 2npq \approx 3045, \quad E_{aa} = n($
- χ² goodness-of-fit test
  $\chi^2 = \sum \frac{(O-E)^2}{E} \approx 0.02698, \quad df = 1, \quad p \approx 0.87$
- p > 0.05 – the departure from Hardy–Weinberg equilibrium isn't significant.

The share of carriers of a recessive disease with an incidence of 1 : 2500.

Input: `model(hardyweinberg; find=frequencies[one]; f=0.04[pct]; input=recessive)`

Result

$p \approx 0.98$

$q \approx 0.02$

$2pq \approx 0.0392$

Steps

- Given: $f_{aa} = 0.04\,\%$
- Converting the units: $f_{aa} = 0.04\,\% = 4 \cdot 10^{-4}$
- In a population where individuals mate at random and there's no selection, mutation or migration, the alleles' frequencies stay constant and the genotypes are in the ratio p² : 2pq : q², where p and q (p + q = 1) are the frequencies of the alleles A and a. From the genotypes' counts we compute p = (2n_AA + n_Aa)/(2n) and the expected counts; a χ² test (1 degree of freedom) shows whether the population departs from equilibrium. For a recessive trait we only know the share of homozygotes q², so q = √q² and the share of carriers 2pq. The model's equation: $p + q = 1, \quad p^2 + 2pq + q^2 = 1$
- The recessive allele's frequency from the homozygotes' share
  $q^2 = f_{aa} \Rightarrow q = \sqrt{4 \cdot 10^{-4}} \approx 0.02, \quad p = 1 - q \approx 0.98$
- Genotype frequencies:
  $p^2 \approx 0.9604, \quad 2pq \approx 0.0392, \quad q^2 = 4 \cdot 10^{-4}$
- The carriers (heterozygotes) are $2pq \approx 3.92\,\% \approx \frac{1}{26}$

How much stock solution for 50 ml diluted tenfold.

Input: `model(dilution; find=V1[ml]; c1=10[mg_ml]; c2=1[mg_ml]; V2=50[ml]; mode=simple)`

Result

$$V_1 \approx 5\,\text{ml}$$

Steps

- Find V_1; given:
  $c_1 = 10\,\text{mg/ml}, \quad c_2 = 1\,\text{mg/ml}, \quad V_2 = 50\,\text{ml}$
- A dilution doesn't change the amount (mass) of the dissolved substance: $c_1V_1 = c_2V_2$. So the stock solution's volume $V_1 = c_2V_2/c_1$ is topped up with the solvent to V_2. In a serial dilution each step dilutes by the same factor d (say 1 : 10); after n steps the concentration is c_0/d^n. In microbiology, from the count N of colonies on a plate that got V_p ml of the dilution D, the number of living cells in the original sample is CFU/ml = N·D/V_p (reliable for 30 to 300 colonies). The model's equation:
  $c_1V_1 = c_2V_2$
- Expressing the sought quantity: $V_1 = \dfrac{c_2V_2}{c_1}$
- Substituting: $V_1 = \dfrac{1 \cdot 50}{10} \approx 5\,\text{ml}$
- The solvent's volume and the dilution:
  $V_2 - V_1 \approx 45\,\text{ml}, \quad \dfrac{c_1}{c_2} \approx 10$

Cells in a sample from the colonies on a plate.

Input: `model(dilution; find=cfu[cfu_ml]; N=150[one]; Vp=0.1[ml]; D=100000[one]; mode=cfu)`

Result

$$\text{CFU} \approx 1.5 \cdot 10^8\,\text{CFU/ml}$$

Steps

- Find CFU; given:
  $N = 150, \quad V_p = 0.1\,\text{ml}, \quad D = 100\,000$
- A dilution doesn't change the amount (mass) of the dissolved substance: $c_1V_1 = c_2V_2$. So the stock solution's volume $V_1 = c_2V_2/c_1$ is topped up with the solvent to V_2. In a serial dilution each step dilutes by the same factor d (say 1 : 10); after n steps the concentration is c_0/d^n. In microbiology, from the count N of colonies on a plate that got V_p ml of the dilution D, the number of living cells in the original sample is CFU/ml = N·D/V_p (reliable for 30 to 300 colonies). The model's equation:
  $\text{CFU/ml} = \dfrac{N \cdot D}{V_p}$
- Substituting:
  $\text{CFU/ml} = \dfrac{150 \cdot 100\,000}{0.1} \approx 1.5 \cdot 10^8\,\text{CFU/ml}$

A neuron's resting membrane potential (Goldman).

Input: `model(nernst; find=E[mV]; Ko=5[mmol_l]; Ki=140[mmol_l]; Nao=145[mmol_l]; Nai=15[mmol_l]; Clo=110[mmol_l]; Cli=10[mmol_l]; PK=1[one]; PNa=0.04[one]; PCl=0.45[one]; T=37[degC]; equation=goldman)`

Result

$$V_m \approx -67.3\,\mathrm{mV}$$

Steps

- Find V_m; given $[\mathrm{K^+}]_o = 5\,\mathrm{mmol/l}, \quad [\mathrm{K^+}]_i = 140\,\mathrm{mmol/l}, \quad [\mathrm{Na^+}]_o =$
- Converting the units: $T = 37\,^\circ\mathrm{C} = 310.1\,\mathrm{K}$
- An ion the membrane lets through distributes so that the electric force balances the diffusion down its concentration gradient. The equilibrium voltage is Nernst's equation E = (RT/zF)·ln(c_o/c_i), where c_o and c_i are the concentrations outside and inside the cell, z the ion's charge, R the gas and F Faraday's constant (at 37 °C RT/F ≈ 26.7 mV). The resting membrane potential, made by several ions, is Goldman's equation – the concentrations weighted by the relative permeabilities P_K, P_Na, P_Cl (for the anion $\mathrm{Cl^-}$ inside and outside swap). The model's equation

$$V_m = \frac{RT}{F} \ln \frac{P_K[\mathrm{K^+}]_o + P_{Na}[\mathrm{Na^+}]_o + P_{Cl}[\mathrm{Cl^-}]_i}{P_K[\mathrm{K^+}]_i + P_{Na}[\mathrm{Na^+}]_i + P_{Cl}[\mathrm{Cl^-}]_o}$$

- The constant RT/F at this temperature: $\frac{RT}{F} = \frac{8.314 \cdot 310.1}{96\,485} \approx 26.73\,\mathrm{mV}$
- Weighted concentrations outside (o) and inside (i) $\mathrm{o} : 1 \cdot 5 + 0.04 \cdot 145 + 0.45 \cdot 10 \approx 15.3, \quad \mathrm{i} : 1 \cdot 140 + 0$
- Substituting: $V_m = 26.73 \cdot \ln \frac{15.3}{190.1} \approx -67.34\,\mathrm{mV}$
- The resting potential is close to the equilibrium potential of the most permeable ion – at rest potassium's.

A cytostatic's dose by body surface.

Input: `model(dosage; find=dose[mg]; h=180[cm]; m=80[kg]; ds=100[mg_m2]; mode=bsa)`

Result

$$D \approx 200\,\mathrm{mg}$$

Steps

- Find D; given:
  $h = 180\,\mathrm{cm}, \quad m = 80\,\mathrm{kg}, \quad d_S = 100\,\mathrm{mg/m^2}$
- Only a guide – no substitute for a clinician's judgement, the package leaflet or local standards.
- A dose by body mass D = d·m (mg/kg × kg); a solution's known concentration gives the volume V = D/c. Cytostatics and children's doses are per body surface: Mosteller's formula S = √(height·mass/3600) (cm, kg, m²), Du Bois's for comparison. An infusion's rate in ml/h is the volume divided by the time, in drops per minute times the set's drop factor too (usually 20 drops/ml). Cockcroft and Gault's creatinine clearance estimates the kidneys' function: Cl = (140 − age)·mass/(72·S_cr) with creatinine in mg/dl (µmol/l divided by 88.4), for women times 0.85. The results are only a guide – no substitute for a clinician's judgement or the package leaflet. The model's equation: $S = \sqrt{\dfrac{h \cdot m}{3600}}, \quad D = d_S \cdot S$
- Body surface (Mosteller): $S = \sqrt{\dfrac{180 \cdot 80}{3600}} \approx 2\,\mathrm{m^2}$
- Du Bois for comparison:
  $S = 0.007184 \cdot h^{0.725} \cdot m^{0.425} \approx 2\,\mathrm{m^2}$
- Substituting: $D = 100 \cdot 2 \approx 200\,\mathrm{mg}$

A woman's creatinine clearance (Cockcroft–Gault).

Input: `model(dosage; find=crcl[ml_min]; age=60[yr]; m=72[kg]; scr=88.4[umol_l]; mode=crcl; sex=female)`

Result

$$Cl_{cr} \approx 68\,\mathrm{ml/min}$$

Steps

- Find Cl_{cr}; given
  $a = 60\,\mathrm{yr}, \quad m = 72\,\mathrm{kg}, \quad S_{cr} = 88.4\,\mu\mathrm{mol/l}$
- Only a guide – no substitute for a clinician's judgement, the package leaflet or local standards.
- A dose by body mass D = d·m (mg/kg × kg); a solution's known concentration gives the volume V = D/c. Cytostatics and children's doses are per body surface: Mosteller's formula S = √(height·mass/3600) (cm, kg, m²), Du Bois's for comparison. An infusion's rate in ml/h is the volume divided by the time, in drops per minute times the set's drop factor too (usually 20 drops/ml). Cockcroft and Gault's creatinine clearance estimates the kidneys' function: Cl = (140 − age)·mass/(72·S_cr) with creatinine in mg/dl (µmol/l divided by 88.4), for women times 0.85. The results are only a guide – no substitute for a clinician's judgement or the package leaflet. The model's equation: $Cl_{cr} = \dfrac{(140 - a)m}{72\,S_{cr}} \cdot 0.85$
- Creatinine in mg/dl (µmol/l divided by 88.4):
  $S_{cr} = \dfrac{88.4}{88.4} \approx 1\,\mathrm{mg/dl}$
- Substituting
  $Cl_{cr} = \dfrac{(140 - 60) \cdot 72}{72 \cdot 1} \cdot 0.85 \approx 68\,\mathrm{ml/min}$

Known limitations

- Enzyme kinetics only for one substrate in the steady state (Michaelis–Menten); allosteric enzymes (Hill's equation) aren't there yet.
- Hardy–Weinberg for one gene with two alleles; more alleles and X-linked genes aren't there yet.
- Dosing is only a guide – it doesn't account for drug interactions, age groups or adjustments for organ function other than the creatinine clearance.

50. Economics, finance and insurance

Financial and economic calculations are on the home screen in the section "Applications", field Economics: interest for days (30E/360, ACT/360, ACT/365), a bill of exchange and its discount, an annuity paid out of savings (temporary, deferred, a perpetuity), mixed interest, savings with an initial deposit and yearly crediting, a loan repaid by a constant principal, interest on a deposit (simple, compound, continuous, with tax and inflation), regular savings, a loan with its amortization schedule and APR, a bond, an investment (NPV, IRR, payback), currency conversion, the break-even point, profit maximization, market equilibrium, Leontief's model and Markov chains, and actuarial mathematics – life tables, life insurance, annuities, reserves and non-life insurance with bonus-malus.

Investing in P2P loans computes the portfolio value (the cash on the account and the outstanding principal) month by month: the deposits are invested in whole amounts per loan, the loans are repaid by an annuity, the platform takes an origination fee and a share of the interest, the losses are a yearly default rate; for comparison a closed formula with continuous reinvestment. Inversely it finds the monthly deposit or the time for a target amount.

Amounts can be in any currency – pick it in the form (the default is your phone's country's); the calculation doesn't depend on it. Currency conversion takes the Czech National Bank's daily rates (in Czech or on a phone set to Czechia) or the European Central Bank's reference rates (in the other languages) and converts any pair through a common currency, showing the rates' source and date.

Amounts in a currency are written by the custom of the app's language and the region in the phone's settings (2,431.50 USD; in Czech 2 431,50 Kč; in German 2.431,50, in Switzerland 2’431.50), and so is the rate table's date.

The steps show the formulas, the substitution and the intermediate results: the effective and the real rate, the amortization schedule, the flows' present value, marginal revenue and cost as derivatives, surpluses as integrals, the Leontief inverse and the stationary distribution by solving a system. Actuarial mathematics uses the commutation numbers; the reserve is shown prospectively and retrospectively.

Examples

A deposit with monthly compounding, tax on interest and inflation.

Input: `model(deposit; find=Kn[one]; K0=100000[one]; p=5[pct]; n=10[one]; d=15[pct];`

`infl=3[pct]; kind=compound; freq=monthly; cur=CZK)`

Result

$$K_n \approx 152{,}844.24\,\mathrm{CZK}$$

Steps

- Given:
  $K_0 = 100{,}000, \quad p = 5\,\%, \quad n = 10, \quad d = 15\,\%, \quad \pi = 3\,\%$
- Converting the units:
  $p = 5\,\% = 0.05, \quad d = 15\,\% = 0.15, \quad \pi = 3\,\% = 0.03$
- With simple interest the interest is always on the original deposit: K_n = K_0(1 + pn). With compound interest it's credited and earns interest too: K_n = K_0(1 + p/m)^(mn), m being the credits a year (annual, semiannual, quarterly, monthly). Continuous compounding is the limit m → ∞: K_n = K_0e^(pn). A tax on interest d lowers the rate to p(1 − d). The effective annual rate compares the frequencies. Discounting is the other way round – how much to deposit today. With inflation π the real value is K_n/(1 + π)^n. Mixed interest (how banks credit interest once a year) compounds the whole years and adds simple interest for the rest of the last one: K_n = K_0(1 + p)^k·(1 + p·f), n = k + f. The model's equation:
  $K_n = K_0\left(1 + \frac{p'}{m}\right)^{mn}$
- Interest credits a year: $m = 12$
- The rate after the tax on interest:
  $p' = p(1 - d) = 5\,\% \cdot (1 - 0.15) = 4.25\,\%$
- Effective annual rate: $p_{ef} = \left(1 + \frac{p'}{m}\right)^{m} - 1 \approx 4.334\,\%$
- Substituting
  $K_n = 100{,}000 \cdot \left(1 + \frac{0.0425}{12}\right)^{120} \approx 152{,}844.24\,\mathrm{CZK}$
- Interest in total: $K_n - K_0 \approx 52{,}844.24\,\mathrm{CZK}$
- Real value (in today's prices):
  $\frac{K_n}{(1+\pi)^n} \approx 113{,}730.47\,\mathrm{CZK}$
- Real interest rate: $r = \frac{1 + p_{ef}}{1 + \pi} - 1 \approx 1.295\,\%$
- The interest rate exceeds inflation – money's real value grows.

Monthly savings for 15 years.

Input: `model(savings; find=S[one]; a=2000[one]; p=3[pct]; n=15[one]; d=15[pct]; timing=due; freq=monthly; cur=CZK)`

Result

$$S \approx 438{,}907.24\,\mathrm{CZK}$$

Steps

- Given: $a = 2{,}000, \quad p = 3\,\%, \quad n = 15, \quad d = 15\,\%$
- Converting the units:
  $p = 3\,\% = 0.03, \quad d = 15\,\% = 0.15$
- With regular savings we deposit an amount a every period (month, year). Each deposit earns interest for a different time, so the amount saved is a geometric series' sum: S = a·q·(q^N − 1)/(q − 1) for deposits at the start of each period (an annuity due), without the first q for deposits at the end (an ordinary annuity); q = 1 + p/m and N = mn is the number of deposits. The equation also gives the deposit needed or the time. An optional initial deposit K_0 earns interest all the time – K_0·q^N is added. With interest credited once a year, the monthly deposits earn simple interest within the year – a year gives A = a(12 + p·13/2) for deposits at the start of the month, a(12 + p·11/2) at its end – and only these yearly amounts compound. The model's equation:
  $S = a\,q\,\frac{q^N - 1}{q - 1}, \quad q = 1 + \frac{p'}{m}, \quad N = mn$
- The rate after the tax on interest:
  $p' = p(1 - d) = 3\,\% \cdot (1 - 0.15) = 2.55\,\%$
- Accumulation factor per period:
  $q = 1 + \frac{0.0255}{12} \approx 1.002125$
- Substituting
  $S = 2{,}000 \cdot 1.002125 \cdot \frac{1.002125^{180} - 1}{0.002125} \approx 438{,}907.24\,\mathrm{C}$
- Deposited in total and the interest
  $N \cdot a = 180 \cdot 2{,}000 = 360{,}000\,\mathrm{CZK}, \quad S - Na \approx 78{,}9($

A mortgage of 3 million for 25 years – payment, schedule and APR.

Input: `model(loan; find=a[one]; D=3000000[one]; p=5[pct]; n=25[one]; F=15000[one]; freq=monthly; cur=CZK)`

Result

$$a \approx 17{,}537.70\,\mathrm{CZK}$$

$$\mathrm{RPSN} \approx 5.169\,\%$$

Steps

- Given:
  $D = 3{,}000{,}000, \quad p = 5\,\%, \quad n = 25, \quad F = 15{,}000$
- Converting the units: $p = 5\,\% = 0.05$
- An annuity loan is repaid by equal payments a. Each payment first pays the interest on the balance and the rest lowers the debt (the principal), so the interest's share falls over time. The present value of all payments equals the loan, which gives a = D·q^N(q − 1)/(q^N − 1), q = 1 + p/m. The amortization schedule shows each payment's interest, principal and the balance left. The APR includes the fees too – the rate at which the payments' present value equals the amount actually received. Repaid by a constant principal, the debt falls by the same M = D/N each period and the payment a_k = M + U_k falls with the interest – less interest is paid than with an annuity, but the first payments are higher. The schedule shows the first year and the last payment, or all of it. The model's equation:
  $$a = D\,\frac{q^N(q-1)}{q^N-1}, \quad q = 1 + \frac{p}{m}, \quad N = mn$$
- Interest per period:
  $$i = \frac{p}{m} = \frac{0.05}{12} \approx 0.00416667, \quad q = 1 + i$$
- Substituting
  $$a = 3{,}000{,}000 \cdot \frac{1.0041667^{300} \cdot 0.00416667}{1.0041667^{300} - 1} \approx 17{,}537.7$$
- Amortization schedule (U interest, M principal, D balance)

k	a	U_k	M_k	L
1	17,537.70	12,500	5,037.70	2,994,962.3
2	17,537.70	12,479.01	5,058.69	2,989,903.6
3	17,537.70	12,457.93	5,079.77	2,984,823.8
4	17,537.70	12,436.77	5,100.94	2,979,722.9
5	17,537.70	12,415.51	5,122.19	2,974,600.7
6	17,537.70	12,394.17	5,143.53	2,969,457.1
7	17,537.70	12,372.74	5,164.96	2,964,292.2
8	17,537.70	12,351.22	5,186.48	2,959,105.7
9	17,537.70	12,329.61	5,208.09	2,953,897.6
10	17,537.70	12,307.91	5,229.79	2,948,667.8
11	17,537.70	12,286.12	5,251.59	2,943,416.2
12	17,537.70	12,264.23	5,273.47	2,938,142.7
⋮	⋮	⋮	⋮	
300	17,537.70	72.77	17,464.93	

- Paid in total and the interest in it
  $$\sum a \approx 5{,}261{,}310.37\,\mathrm{CZK}, \quad \sum U_k \approx 2{,}261{,}310.37\,\mathrm{C}$$
- APR – the annual percentage rate with the fees
  $$D - F = \sum_{k=1}^{N} \frac{a}{(1+\mathrm{RPSN})^{k/m}} \Rightarrow \mathrm{RPSN} \approx 5.169\,\%$$
- › The method: bisection, 14 iterations

Saving 1,000 a month with an initial deposit and interest credited yearly (as taught at Czech schools).

Input: `model(savings; find=S[one]; a=1000[one]; K0=20000[one]; p=2[pct]; n=5[one];`

`timing=due; freq=monthly; credit=annual; cur=CZK)`

Result

$$S \approx 85{,}206.62\,\mathrm{CZK}$$

Steps

- Given: $a = 1{,}000, \quad K_0 = 20{,}000, \quad p = 2\,\%, \quad n = 5$
- Converting the units: $p = 2\,\% = 0.02$
- With regular savings we deposit an amount a every period (month, year). Each deposit earns interest for a different time, so the amount saved is a geometric series' sum: S = a·q·(q^N − 1)/(q − 1) for deposits at the start of each period (an annuity due), without the first q for deposits at the end (an ordinary annuity); q = 1 + p/m and N = mn is the number of deposits. The equation also gives the deposit needed or the time. An optional initial deposit K_0 earns interest all the time – K_0·q^N is added. With interest credited once a year, the monthly deposits earn simple interest within the year – a year gives A = a(12 + p·13/2) for deposits at the start of the month, a(12 + p·11/2) at its end – and only these yearly amounts compound. The model's equation

$$A = a\left(m + p'\,\frac{m+1}{2}\right), \quad S = A\,\frac{r^n - 1}{r - 1} + K_0 r^n, \quad r =$$

- A year's deposits with the interest within the year:
  $$A = a \cdot \left(12 + 0.02 \cdot \frac{13}{2}\right) = a \cdot 12.13$$
- Accumulation factor per period: $r = 1 + 0.02 = 1.02$
- Substituting
  $$S = 1{,}000 \cdot 12.13 \cdot \frac{1.02^5 - 1}{0.02} + 20{,}000 \cdot 1.02^5 \approx 85{,}20$$
- Deposited in total and the interest
  $$K_0 + N \cdot a = 20{,}000 + 60 \cdot 1{,}000 = 80{,}000\,\mathrm{CZK}, \quad S$$

Investing in P2P loans: 2,000 CZK a month for 10 years, 200 CZK per loan, with a fee and losses.

Input: `model(lending; find=V[one]; a=2000[one]; p=8[pct]; f=1[pct]; loss=2[pct]; L=200[one];`

`M=60[one]; n=10[one]; cur=CZK)`

Result

$$V \approx 318{,}974.12\,\text{CZK}$$

Steps

- Given

  $a = 2{,}000, \quad p = 8\,\%, \quad f = 1\,\%, \quad \lambda = 2\,\%, \quad L = 20$

- Converting the units:

  $p = 8\,\% = 0.08, \quad f = 1\,\% = 0.01, \quad \lambda = 2\,\% = 0.02$

- On a P2P lending platform the investor deposits an amount a every month. The cash is invested in whole multiples of the amount L per loan (e.g. 200 CZK), the rest waits on the account. The platform deducts the origination fee f from each investment, L(1 − f) goes into the loan. The loans are repaid by an annuity, equal monthly payments s = P₀·i/(1 − (1 + i)^(−M)) over M months, i = p/12; the platform keeps the share g of the interest. The default rate λ (a year) writes off part of the principal each month, λ_m = 1 − (1 − λ)^(1/12). The payments return to the account and are reinvested. The portfolio value V = H + P is the cash on the account H plus the outstanding principal in loans P. The calculation simulates month by month and, for comparison, gives a closed formula with continuous reinvestment: V ≈ 12a(1 − f)(e^(rn) − 1)/r, r = p(1 − g)(1 − f) − λ − f·24/(M + 1). The inverse problem finds the monthly deposit or the time needed for a target value. It is an estimate with a fixed return – the real outcome depends mainly on the losses. The return a platform quotes is often already after fees and losses; then do not enter them again. This is not investment advice. The model's equation

  $$V_k = H_k + P_k, \quad P' = P(1-\lambda_m) - (s - iP), \quad H' =$$

- Monthly interest, monthly losses and the annuity payment

  $$i = \frac{p}{12} = 0.00666667, \quad \lambda_m = 1 - (1-\lambda)^{1/12} \approx 0.001$$

- Substituting

  $$V_{120} = H_{120} + P_{120} = 10{,}735.19 + 308{,}238.93 \approx 318{,}97$$

- Month-by-month simulation – the state at the end of the years (k in months): deposited, principal in loans P, cash H, fees F, losses Z, value V

k	$\sum a$	P_k	H_k	F_k	
12	24,000	23,784.39	726.58	266	2
24	48,000	49,198.83	1,248.72	606	1,0
36	72,000	75,757.04	2,129.17	1,032	2,3
48	96,000	103,744.84	3,148.57	1,568	4,2
60	120,000	133,102.33	4,439.70	2,240	6,6
72	144,000	164,382.06	5,541.37	3,054	9,8
84	168,000	197,551.52	6,600.52	4,002	13,5
96	192,000	232,369.52	7,962.71	5,094	18,0
108	216,000	269,327.69	9,238.91	6,346	23,3
120	240,000	308,238.93	10,735.19	7,764	29,3

- Deposited, fees F, losses Z, net interest U and the gain

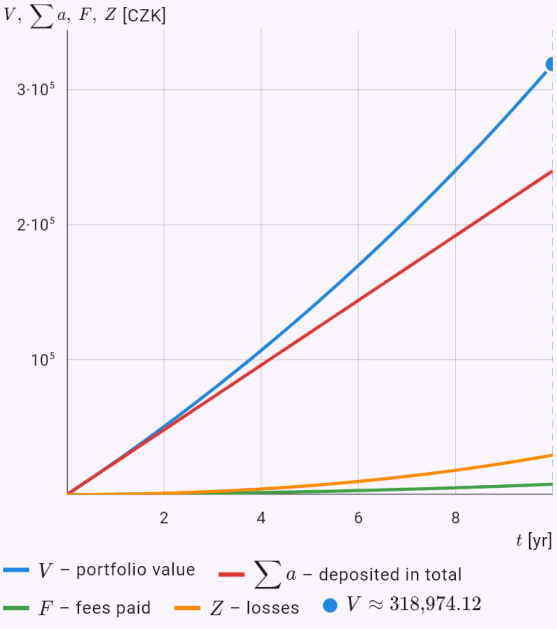
  $$\sum a = 240{,}000\,\text{CZK}, \quad F = 7{,}764\,\text{CZK}, \quad Z = 29{,}33$$

- For comparison: the closed formula with continuous reinvestment

  $$r = p(1-g)(1-f) - \lambda - f\,\frac{24}{M+1} \approx 5.527\,\%, \quad V \approx$$

- This is an estimate with a fixed return and a constant default rate – the reality depends mainly on the losses. The return a platform quotes is often already after fees and losses. This is not investment advice.

Graph



The model's quantities in time, computed from the solution; the answer is marked.

A loan repaid by a constant principal – falling payments compared with an annuity.

Input: `model(loan; find=a[one]; D=1200000[one]; p=5[pct]; n=10[one]; freq=monthly;`

`method=constant; schedule=short; cur=CZK)`

Result

$a_1 \approx 15{,}000\,\mathrm{CZK}$

$a_N \approx 10{,}041.67\,\mathrm{CZK}$

Steps

- Given: $D = 1{,}200{,}000, \quad p = 5\,\%, \quad n = 10$
- Converting the units: $p = 5\,\% = 0.05$
- An annuity loan is repaid by equal payments a. Each payment first pays the interest on the balance and the rest lowers the debt (the principal), so the interest's share falls over time. The present value of all payments equals the loan, which gives a = D·q^N(q − 1)/(q^N − 1), q = 1 + p/m. The amortization schedule shows each payment's interest, principal and the balance left. The APR includes the fees too – the rate at which the payments' present value equals the amount actually received. Repaid by a constant principal, the debt falls by the same M = D/N each period and the payment a_k = M + U_k falls with the interest – less interest is paid than with an annuity, but the first payments are higher. The schedule shows the first year and the last payment, or all of it. The model's equation

  $$M = \frac{D}{N}, \quad a_k = M + U_k, \quad U_k = D_{k-1}\, i, \quad i = \frac{p}{m},$$

- Interest per period: $i = \dfrac{p}{m} = \dfrac{0.05}{12} \approx 0.00416667$
- Substituting: $M = \dfrac{1{,}200{,}000}{120} \approx 10{,}000\,\mathrm{CZK}$
- The first and the last payment

  $a_1 = M + D\, i \approx 15{,}000\,\mathrm{CZK}, \quad a_N = M(1 + i) \approx 10{,}$
- Amortization schedule (U interest, M principal, D balance)

k	a_k	U_k	M_k	D_k
1	15,000	5,000	10,000	1,190,000
2	14,958.33	4,958.33	10,000	1,180,000
3	14,916.67	4,916.67	10,000	1,170,000
4	14,875	4,875	10,000	1,160,000
5	14,833.33	4,833.33	10,000	1,150,000
6	14,791.67	4,791.67	10,000	1,140,000
7	14,750	4,750	10,000	1,130,000
8	14,708.33	4,708.33	10,000	1,120,000
9	14,666.67	4,666.67	10,000	1,110,000
10	14,625	4,625	10,000	1,100,000
11	14,583.33	4,583.33	10,000	1,090,000
12	14,541.67	4,541.67	10,000	1,080,000
⋮	⋮	⋮	⋮	⋮
120	10,041.67	41.67	10,000	0

- Paid in total and the interest in it

  $\sum a_k \approx 1{,}502{,}500\,\mathrm{CZK}, \quad \sum U_k \approx 302{,}500\,\mathrm{CZK}$
- Compared with an annuity – the payment and the total interest:

  $a \approx 12{,}727.86\,\mathrm{CZK}, \quad \sum U_k \approx 327{,}343.42\,\mathrm{CZK}$
- Repaying by a constant principal saves $24{,}843.42\,\mathrm{CZK}$ of interest, but the first payments are higher.

Interest for days by 30E/360 with a 15 % tax.

Input: `model(daycount; find=Kn[one]; K0=50000[one]; p=2[pct]; start=2026-03-15; end=2026-09-`

`20; d=15[pct]; basis=e30; cur=CZK)`

Result

$$u' \approx 436.81\,\mathrm{CZK}$$
$$K_n \approx 50{,}436.81\,\mathrm{CZK}$$

Steps

- Given: $K_0 = 50{,}000, \quad p = 2\,\%, \quad d = 15\,\%$
- Converting the units: $p = 2\,\% = 0.02, \quad d = 15\,\% = 0.15$
- Short-term deposits earn simple interest for t days: u = $K_0 \cdot p \cdot t/B$. The days and the year's length B follow the day count convention: 30E/360 (Czech and German banks and schools) counts each month as 30 days (the 31st as the 30th) and the year as 360 days, ACT/360 (France, Spain, money markets) the actual days and a year of 360, ACT/365 (the UK) the actual days and a year of 365. The days are counted from the deposit's and the withdrawal's dates, or typed directly. The tax on interest is taken off the interest. The opposite – how much to deposit today for an amount – is discounting. The model's equation: $u = K_0 \cdot p \cdot \frac{t}{B}, \quad K_n = K_0 + u$

- Day count convention and the year's length: $30\mathrm{E}/360: \quad B = 360$
- Days (30E/360: a month of 30 days, the 31st as the 30th) $3/15/2026 \to 9/20/2026: \quad t = 360 \cdot (2026 - 2026) +$
- Substituting: $u = 50{,}000 \cdot 0.02 \cdot \frac{185}{360} \approx 513.89\,\mathrm{CZK}$
- The tax on the interest and the interest after tax $u \cdot d \approx 77.08\,\mathrm{CZK}, \quad u' = u - u \cdot d \approx 436.81\,\mathrm{CZK}$
- Interest in total: $K_n = K_0 + u' \approx 50{,}436.81\,\mathrm{CZK}$

A bill discounted – what the bank pays 90 days before maturity.

Input: `model(bill; find=payout[one]; N=100000[one]; dr=6[pct]; start=2026-01-10; end=2026-04-10; basis=act360; cur=CZK)`

Result

$$K \approx 98{,}500\,\mathrm{CZK}$$
$$D \approx 1{,}500\,\mathrm{CZK}$$

Steps

- Given: $N = 100{,}000, \quad d = 6\,\%$
- Converting the units: $d = 6\,\% = 0.06$
- A bill of exchange promises to pay its nominal N on the maturity date. A bank may buy it earlier (discount it) and pays it less the discount D = N·d·t/B, where d is the discount rate and t the days to maturity: K = N(1 − d·t/B). The discount is taken from the final amount, not the one paid, so the bank's real yield i = d/(1 − d·t/B) is higher than the discount rate. The usual convention is ACT/360. The model's equation: $D = N \cdot d \cdot \frac{t}{B}, \quad K = N - D = N\left(1 - d\,\frac{t}{B}\right)$
- Day count convention and the year's length: $\mathrm{ACT}/360: \quad B = 360$

- Days by the calendar: $1/10/2026 \to 4/10/2026: \quad t = 90$
- Substituting: $D = 100{,}000 \cdot 0.06 \cdot \frac{90}{360} \approx 1{,}500\,\mathrm{CZK}$
- Amount paid out: $K = N - D \approx 98{,}500\,\mathrm{CZK}$
- The bank's real yield: $i = \frac{d}{1 - d\,\frac{t}{B}} \approx 6.091\,\%$
- The discount is taken from the final amount – so the bank really earns more than the discount rate.

A monthly annuity from 2 million for 20 years, deferred by 5 years.

Input: `model(annuity; find=a[one]; PV=2000000[one]; p=3[pct]; n=20[one]; k=5[one];`

`kind=temporary; timing=ordinary; freq=monthly; cur=CZK)`

Result

$$a \approx 12{,}884.60\,\mathrm{CZK}$$

Steps

- Given: $K = 2{,}000{,}000, \quad p = 3\,\%, \quad n = 20, \quad k = 5$
- Converting the units: $p = 3\,\% = 0.03$
- An annuity pays a regular amount a out of a capital K, which keeps earning interest meanwhile. The present value of N payments is K = a·(1 − q^(−N))/(q − 1), q = 1 + p/m – the same formula as a loan's repayment. Payments at the start of the periods (an annuity due) multiply it by q. A deferred annuity starts paying after k years, the capital growing to K·q^(mk) meanwhile. A perpetuity pays the interest only and the capital stays: K = a/(q − 1). The result is the payment, the capital needed, or how long the payments last. The model's equation:
  $K = a\,\dfrac{1 - q^{-N}}{q - 1} \cdot q^{-mk}, \quad q = 1 + \dfrac{p'}{m}, \quad N = mn$
- Accumulation factor per period: $q = 1 + \dfrac{0.03}{12} \approx 1.0025$
- Discounted over the deferral:
  $q^{-mk} = 1.0025^{-60} \approx 0.860869$
- Expressing the sought quantity: $a = \dfrac{K(q-1)q^{mk}}{1 - q^{-N}}$
- Substituting: $a \approx 12{,}884.60\,\mathrm{CZK}$
- The capital when the payments start:
  $K\,q^{mk} \approx 2{,}323{,}233.56\,\mathrm{CZK}$

- The payments (U interest, K balance)

k	a	U_k	K_k
1	12,884.60	5,808.08	2,316,157.05
2	12,884.60	5,790.39	2,309,062.84
3	12,884.60	5,772.66	2,301,950.90
4	12,884.60	5,754.88	2,294,821.18
5	12,884.60	5,737.05	2,287,673.64
6	12,884.60	5,719.18	2,280,508.23
7	12,884.60	5,701.27	2,273,324.90
8	12,884.60	5,683.31	2,266,123.61
9	12,884.60	5,665.31	2,258,904.33
10	12,884.60	5,647.26	2,251,666.99
11	12,884.60	5,629.17	2,244,411.56
12	12,884.60	5,611.03	2,237,137.99
⋮	⋮	⋮	⋮
240	12,884.60	32.13	0

- Paid out in total and the interest in it
  $\sum a \approx 3{,}092{,}303.41\,\mathrm{CZK}, \quad \sum a - K \approx 1{,}092{,}303.4$

A bond's price and duration.

Input: `model(bond; find=price[one]; N=1000[one]; cp=5[pct]; n=5[one]; y=4[pct]; freq=annual; cur=EUR)`

Result

$$P \approx 1{,}044.52\,€$$

Steps

- Given: $N = 1{,}000, \quad c = 5\,\%, \quad n = 5, \quad y = 4\,\%$
- Converting the units: $c = 5\,\% = 0.05, \quad y = 4\,\% = 0.04$
- A bond pays a coupon C = c·N/m every period and its face value N at the end. Its price is the present value of these payments discounted by the market yield y. When the market yield is below the coupon rate, the bond sells above par (at a premium), otherwise below it. The yield to maturity is the rate at which the payments' present value equals the price – found numerically. The duration is the payments' weighted average time and measures the price's sensitivity to interest rates. The model's equation
  $P = \displaystyle\sum_{k=1}^{N} \frac{C}{\left(1 + \frac{y}{m}\right)^k} + \frac{N_{nom}}{\left(1 + \frac{y}{m}\right)^N}$

- Coupon per period: $C = \dfrac{c \cdot N_{nom}}{m} = 50\,€, \quad N = 5$
- Substituting
  $P = 50 \cdot \dfrac{1 - 0.961538^5}{0.04} + \dfrac{1{,}000}{1.04^5} \approx 222.59 + 821.93 =$
- The market yield is below the coupon – the bond sells above par.
- Duration (Macaulay and modified)
  $D = \dfrac{1}{P} \displaystyle\sum_k t_k\, PV_k \approx 4.557, \quad D_{mod} = \dfrac{D}{1 + \frac{y}{m}} \approx 4.382$

An investment's internal rate of return.

Input: `model(npv; find=irr[one]; CF=-1000 300 400 500 200[one]; r=10[pct]; cur=EUR)`

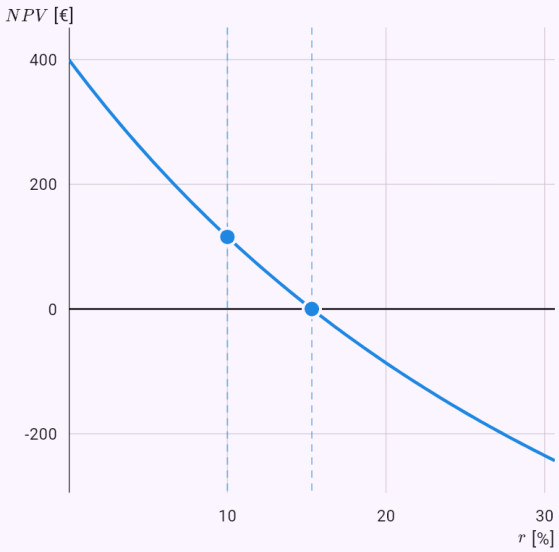
Result

$$IRR \approx 15.32\,\%$$

Steps

- Given:
  $CF_t = (-1{,}000;\ 300;\ 400;\ 500;\ 200), \quad r = 10\,\%$
- Converting the units: $r = 10\,\% = 0.1$
- The net present value NPV adds up a project's cash flows CF_0, CF_1, … discounted by the required return r: NPV = Σ CF_t/(1 + r)^t. A project with NPV > 0 earns more than the required return. The internal rate of return IRR is the rate at which NPV = 0 – a polynomial equation's root, found numerically. The payback period is when the (simple or discounted) running sum of the flows stops being negative. The model's equation:
  $$NPV = \sum_{t=0}^{n} \frac{CF_t}{(1+r)^t}$$
- Net present value
  $$NPV = -1{,}000 + \frac{300}{1.1^1} + \frac{400}{1.1^2} + \frac{500}{1.1^3} + \frac{200}{1.1^4} \approx 115.5$$
- NPV ≥ 0 – the project earns at least the required return.
- Solving the equation numerically (bisection):
  $NPV(IRR) = 0 \Rightarrow IRR \approx 15.32\,\%$
- › The method: bisection, 14 iterations

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

Euros to korunas by today's ČNB rate.

Input: `model(exchange; find=converted[one]; A=100[one]; from=EUR; to=CZK; source=cnb)`

Result

$$2{,}443.50\ \mathrm{CZK}$$

$$1\ € \approx 24.435\ \mathrm{CZK}$$

Steps

- Given: $A = 100$
- An exchange rate is the price of a unit of a foreign currency. The Czech National Bank publishes the koruna's rates every working day, the European Central Bank the euro's reference rates. An amount converts through a common currency: $A_2 = A_1 \cdot k_1 / k_2$, where k_1 and k_2 are both currencies' values in korunas (ČNB) or in euros (ECB) – a cross rate. Banks and exchange offices use their own buying and selling rates with a margin; the result is indicative. The model's equation: $A_2 = A_1 \cdot \frac{k_1}{k_2}$
- Exchange rate table: $\check{\mathrm{C}}\mathrm{NB},\ 11/27/2026$
- Rates: $1\ € = 24.435\ \mathrm{CZK}$
- Substituting: $A_2 = 100 \cdot \frac{24.435}{1} \approx 2{,}443.50\ \mathrm{CZK}$
- Banks and exchange offices use their own rates with a margin – the result is indicative.

The profit maximum from marginal revenue and cost.

Input: `model(profit; find=optimum[one]; a=100[one]; b=2[one]; F=200[one]; cv=20[one];`

e=0.5[one]; cur=CZK)

Result

$Q^* \approx 16$

$p^* \approx 68\,\mathrm{CZK}$

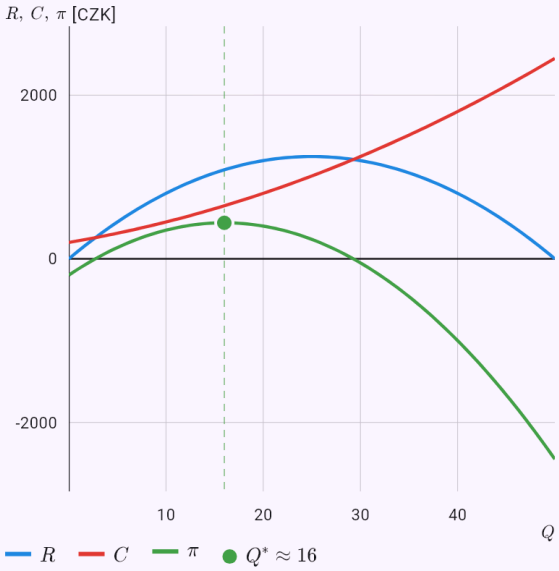
$\pi^* \approx 440\,\mathrm{CZK}$

Steps

- Given:
  $a = 100, \quad b = 2, \quad F = 200, \quad c = 20, \quad e = 0.5$
- A firm with a linear demand p = a − bQ has the revenue R = pQ and the costs C(Q) = F + cQ + eQ². The profit π = R − C is largest where its derivative is zero, i.e. the marginal revenue equals the marginal cost: MR = R'(Q) = MC = C'(Q). The second derivative confirms a maximum. The price elasticity of demand ε = (dQ/dp)·(p/Q) says by how many percent the quantity demanded changes when the price changes by 1 %; |ε| > 1 is an elastic demand. The model's equation:
  $p = a - bQ, \quad R = pQ, \quad \pi = R - C$
- Revenue and costs:
  $R(Q) = (100 - 2Q)Q, \quad C(Q) = 200 + 20Q + 0.5Q^2$
- Marginal revenue and marginal cost (derivatives):
  $MR = R'(Q) = 100 - 4Q, \quad MC = C'(Q) = 20 + 1Q$
- Substituting:
  $MR = MC: \quad 100 - 4Q = 20 + 1Q \Rightarrow Q^* = \frac{80}{5} \approx 16$
- The profit's second derivative is negative – a maximum:
  $\pi''(Q) = -5 < 0$
- The price and the profit at the optimum:
  $p^* = 100 - 2 \cdot 16 \approx 68\,\mathrm{CZK}, \quad \pi^* \approx 440\,\mathrm{CZK}$

- Price elasticity of demand: $\varepsilon = -\frac{p^*}{bQ^*} \approx -2.125$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

Market equilibrium, surpluses and a tax.

Input: `model(market; find=equilibrium[one]; a=100[one]; b=2[one]; cs=10[one]; d=3[one];`

`t=5[one]; cur=CZK)`

Result

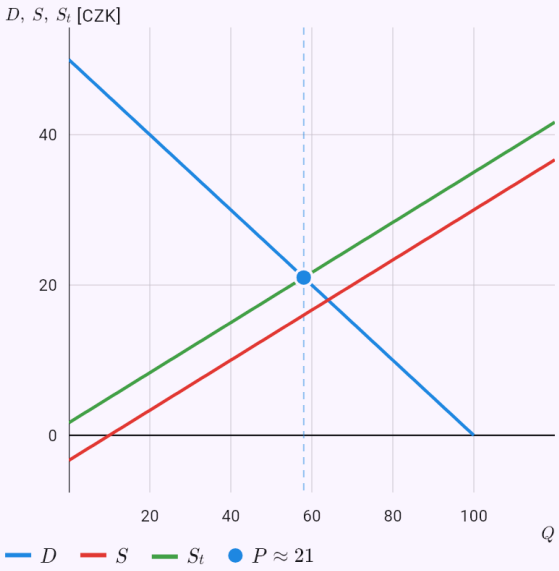
$$P \approx 21\,\text{CZK}$$
$$Q \approx 58$$
$$CS \approx 841\,\text{CZK}$$
$$PS \approx 560.67\,\text{CZK}$$

Steps

- Given: $a = 100, \quad b = 2, \quad c = 10, \quad d = 3, \quad t = 5$
- The demand Q_D = a − bP falls with the price, the supply Q_S = c + dP rises. The market is in equilibrium where they meet: P* = (a − c)/(b + d). The consumer surplus is the area between the demand curve and the price (what buyers would have paid on top), the producer surplus the area between the price and the supply curve – both integrals. A tax t per unit shifts the supply up by t: buyers pay more, sellers get less, less is traded and a deadweight loss arises. The model's equation: $Q_D = a - bP, \quad Q_S = c + d(P - t), \quad Q_D = Q_S$
- Substituting: $100 - 2P = 10 + 3P \Rightarrow P^* = \frac{90}{5} \approx 18, \quad Q^* \approx 64$
- With the tax: the buyers' and the sellers' price $P_D = \frac{a - c + dt}{b + d} \approx 21, \quad P_S = P_D - t \approx 16, \quad Q_t \approx 5$
- Tax revenue and deadweight loss $tQ_t \approx 290\,\text{CZK}, \quad \frac{1}{2}t(Q^* - Q_t) \approx 15\,\text{CZK}$
- Consumer surplus: $CS = \int_0^Q \left(\frac{a - Q}{b} - P_D\right) dQ \approx 841\,\text{CZK}$
- Producer surplus: $PS = \int_0^Q \left(P_S - \frac{Q - c}{d}\right) dQ \approx 560.67\,\text{CZK}$

- Demand's and supply's elasticity at the equilibrium $\varepsilon_D = -b\frac{P}{Q} \approx -0.7241, \quad \varepsilon_S = d\frac{P}{Q} \approx 0.8276$

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

Leontief's model of three sectors.

Input: `model(leontief; find=x[one]; A=0.2 0.3 0.1 0.4 0.1 0.2 0.1 0.2 0.3[one]; y=100 200`

```
150[one])
```

Result

$$x_1 \approx 349.3$$
$$x_2 \approx 465.8$$
$$x_3 \approx 397.3$$

Steps

- Given
  $A = (0.2;\ 0.3;\ 0.1;\ 0.4;\ 0.1;\ 0.2;\ 0.1;\ 0.2;\ 0.3), \quad y =$
- Leontief's model describes an economy of n sectors supplying each other. The element a_ij of the technical coefficients' matrix A says how much of sector i's output sector j uses per unit of its own output. The gross output x must cover the intermediate use Ax and the final demand y: x = Ax + y, so x = (I − A)⁻¹y. The Leontief inverse (I − A)⁻¹ shows how much each sector must produce per unit of final demand. The model's equation: $x = Ax + y \Rightarrow (I - A)x = y \Rightarrow x = (I - A)^{-1}y$
- Technical coefficients and final demand
  $$A = \begin{pmatrix} 0.2 & 0.3 & 0.1 \\ 0.4 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.3 \end{pmatrix}, \quad y = (100;\ 200;\ 150)$$
- The matrix I − A
  $$I - A = \begin{pmatrix} 0.8 & -0.3 & -0.1 \\ -0.4 & 0.9 & -0.2 \\ -0.1 & -0.2 & 0.7 \end{pmatrix}$$
- Leontief inverse
  $$(I - A)^{-1} \approx \begin{pmatrix} 1.616 & 0.6301 & 0.411 \\ 0.8219 & 1.507 & 0.5479 \\ 0.4658 & 0.5205 & 1.644 \end{pmatrix}$$
- Substituting: $x = (I - A)^{-1}y \approx (349.3;\ 465.8;\ 397.3)$
- Inter-sector flows
  $$x_{ij} = a_{ij}x_j: \quad \begin{pmatrix} 69.86 & 139.7 & 39.73 \\ 139.7 & 46.58 & 79.45 \\ 34.93 & 93.15 & 119.2 \end{pmatrix}$$

Three brands' long-run customer shares.

Input: `model(markov; find=stationary[one]; P=0.8 0.15 0.05 0.1 0.8 0.1 0.05 0.15 0.8[one])`

Result

$$\pi \approx (0.2857;\ 0.4286;\ 0.2857)$$

Steps

- Given:
  $P = (0.8;\ 0.15;\ 0.05;\ 0.1;\ 0.8;\ 0.1;\ 0.05;\ 0.15;\ 0.8)$
- A Markov chain moves between states with probabilities p_ij that depend only on the present state (not on the history). The transition matrix P's rows add up to 1. The distribution after one step is $\pi_1 = \pi_0 P$, after n steps $\pi_n = \pi_0 P^n$. The stationary distribution satisfies $\pi P = \pi$ and shows the states' long-run shares – brands' customers, bonus-malus classes, a machine's states. The model's equation: $\pi_{k+1} = \pi_k P, \quad \pi_n = \pi_0 P^n, \quad \pi P = \pi$
- Transition matrix
  $$P = \begin{pmatrix} 0.8 & 0.15 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.15 & 0.8 \end{pmatrix}$$
- The system for the stationary distribution
  $$\begin{aligned} \pi_1 &= 0.8\,\pi_1 + 0.1\,\pi_2 + 0.05\,\pi_3 \\ \pi_2 &= 0.15\,\pi_1 + 0.8\,\pi_2 + 0.15\,\pi_3 \\ \pi_3 &= 0.05\,\pi_1 + 0.1\,\pi_2 + 0.8\,\pi_3 \end{aligned}$$
  $$\pi_1 + \pi_2 + \pi_3 = 1$$
- In the long run the distribution settles at the stationary one whatever the start (for a regular chain).

A woman's life expectancy at 65.

Input: `model(mortality; find=expectancy[one]; x=65[one]; table=female)`

Result

$$e_{65} \approx 20.57$$

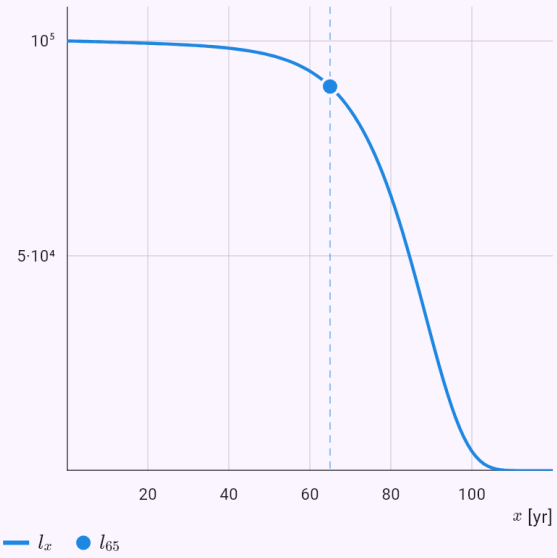
Steps

- Given: $x = 65$
- A model table (Gompertz–Makeham) roughly like the Czech population's – for precise results enter official q_x.
- A life table gives how many of 100 000 newborns (l_x) live to the age x. The probability that a person aged x lives n more years is ${}_np_x = l_{x+n}/l_x$, of dying ${}_nq_x = 1 - {}_np_x$. The life expectancy e_x is the average number of years a person aged x still lives. The model table by Gompertz–Makeham's law $\mu(x) = A + B \cdot c^x$ is set roughly like the Czech population; precise calculations use official tables – enter your own q_x. The model's equation

$$ {}_np_x = \frac{l_{x+n}}{l_x}, \quad {}_nq_x = 1 - {}_np_x, \quad e_x \approx \frac{1}{l_x}\sum_{k\geq 1} l_{x+k} + \frac{1}{2} $$

- Survivors from the table: $l_{65} \approx 89\,397.8$
- Substituting: $e_{65} \approx 20.57$
- On average they live to 85.57

Graph



The graph computed from the model's solution; the dots are the values given or the answer.

An endowment's annual premium.

Input: `model(lifeinsurance; find=premium[one]; x=40[one]; n=20[one]; i=2.5[pct]; S=1000000[one]; product=endowment; payment=annual; table=male; cur=CZK)`

Result

$$P \approx 40{,}227.05\ \text{CZK}$$

Steps

- Given: $x = 40, \quad n = 20, \quad i = 2.5\,\%, \quad S = 1{,}000{,}000$
- Converting the units: $i = 2.5\,\% = 0.025$
- A model table (Gompertz–Makeham) roughly like the Czech population's – for precise results enter official q_x.
- The net premium comes from the equivalence principle: the premiums' present value equals the future benefits' present value. With the commutation numbers $D_x = v^x l_x$, N_x, C_x and M_x ($v = 1/(1+i)$, i the technical interest rate) the net single premium of a term insurance for n years is $(M_x - M_{x+n})/D_x$, of a pure endowment D_{x+n}/D_x, an endowment their sum; an annuity $\ddot{a} = (N_x - N_{x+n})/D_x$. The annual premium divides by the annuity $\ddot{a}$. The reserve in year t is the future benefits' minus the future premiums' present value (prospectively) – and comes out the same as the premiums received minus the benefits paid, accumulated (retrospectively). The model's equation

$$D_x = v^x l_x, \quad N_x = \sum_{k\geq x} D_k, \quad C_x = v^{x+1} d_x, \quad M_x = \sum_{k\geq}$$

- Commutation numbers

$$D_{40} \approx 35\,866.1, \quad N_{40} \approx 883\,077, \quad M_{40} \approx 14\,327.6,$$

- Net single premium

$$A_{x:\overline{20|}} \approx 0.622543, \quad S \cdot A_{x:\overline{20|}} \approx 622{,}543.08\ \text{CZK}$$

- Annual premium

$$\ddot{a}_{x:\overline{20|}} \approx 15.4757, \quad P = \frac{S \cdot A_{x:\overline{20|}}}{\ddot{a}_{x:\overline{20|}}} \approx 40{,}227.05\ \text{CZK}$$

A bonus-malus system and the average premium level.

Input: `model(nonlife; find=stationary[one]; levels=100 90 80 70 60 50[one]; q=10[pct]; mode=bonusmalus; rule=reset; cur=CZK)`

Result

$$\bar{b} \approx 63.14$$

Steps

- Given: $b = (100;\ 90;\ 80;\ 70;\ 60;\ 50), \quad q = 10\,\%$
- Converting the units: $q = 10\,\% = 0.1$
- In non-life insurance (cars, property) a policyholder's expected annual claims are E[S] = λ·m, λ being the mean number of claims a year and m the mean claim. The risk premium gets a loading θ for costs and safety. A bonus-malus system changes the premium by the claims history: a year without a claim moves to a better class, a claim back. The classes make a Markov chain; its stationary distribution gives the classes' long-run shares and the average premium level. The model's equation: $\pi P = \pi, \quad \bar{b} = \sum_i \pi_i b_i$
- A year without a claim moves a class down (a discount), a claim back to the first class.
- Transition matrix

$$P = \begin{pmatrix} 0.1 & 0.9 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0.9 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0.9 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0.9 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 0.9 \\ 0.1 & 0 & 0 & 0 & 0 & 0.9 \end{pmatrix}$$

- The system for the stationary distribution

$$\begin{aligned} \pi_1 &= 0.1\,\pi_1 + 0.1\,\pi_2 + 0.1\,\pi_3 \\ \pi_2 &= 0.9\,\pi_1 \\ \pi_3 &= 0.9\,\pi_2 \\ \pi_4 &= 0.9\,\pi_3 \\ \pi_5 &= 0.9\,\pi_4 \\ \pi_6 &= 0.9\,\pi_5 + 0.9\,\pi_6 \end{aligned}$$

$$\pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 + \pi_6 = 1$$

- The classes' long-run shares: $\pi \approx (0.1;\ 0.09;\ 0.081;\ 0.0729;\ 0.0656;\ 0.5905)$
- Substituting: $\bar{b} = \sum_i \pi_i b_i \approx 63.14$

Known limitations

- The life tables are model ones (Gompertz–Makeham, roughly like the Czech population); for precise results enter your own q_x from official tables.
- The rates are reference rates – banks and exchange offices use their own rates and fees.
- Microeconomics with a linear demand and quadratic costs; general functions given by a formula aren't there yet.
- The conventions don't move dates for holidays – interest is for the days between the dates given; a loan has no extra repayments or rate changes.
- Investing in P2P loans is an estimate with a fixed return and a constant default rate; it ignores debt recovery, early repayment, a secondary market and taxes. This is not investment advice.

51. Linear programming, transportation, assignment and matrix games

Optimization problems are on the home screen in the "Applications" section, field Logistics and networks (linear programming also in Economics): linear and integer programming, the transportation problem and the assignment problem. A linear program can also be typed or photographed, e.g. max 3x + 2y; x + y <= 4; x + 3y <= 6; x, y >= 0 – the constraints on separate lines or after a semicolon, min instead of max, and the word integer makes it an integer program.

The steps show the standard form with slack and artificial variables and the simplex tableaux in exact fractions with the pivot marked. Constraints ≥ and = are handled by the two-phase method by default; a chip selects the big M method. The optimum comes with the dual program, the constraints' shadow prices and a sensitivity analysis (the ranges of the objective's coefficients and of the right sides). With two variables the app draws the feasible region, the vertices with the objective's values and the level line through the optimum. An integer program is solved by branch and bound, and you go through the branching tree with the arrows as a graph.

The transportation problem is given by the matrix of unit costs, the suppliers' supplies and the customers' demands; an unbalanced problem gets a dummy supplier or customer. The starting solution comes from the north-west corner, the least cost or Vogel's approximation method, the optimum from the MODI method with potentials and closed loops; the transportation plan is also shown as a graph. The assignment problem is solved by the Hungarian method – row and column reduction, covering the zeros with lines and choosing independent zeros; it handles a maximum and a rectangular matrix too.

Two-player matrix games (the Logistics and networks field, or the notation game(3 -1 2; -2 4 1) – the rows are player A's strategies, the columns player B's): a zero-sum game first looks for a saddle point (maximin = minimax), then crosses out dominated strategies and computes the mixed strategies by a formula for a 2 × 2 game, graphically for 2 × n and m × 2 (the lines' lower and upper envelope) and by linear programming with simplex tableaux for a larger one – the dual problem gives the other player's strategy. The result is both players' optimal strategies and the value of the game in exact fractions. A bimatrix game (game(2 0; 0 1 | 1 0; 0 2), A's payoffs | B's payoffs) shows the best responses and the Nash equilibria in pure strategies, for a 2 × 2 game the mixed equilibrium too.

Examples

A production plan by the simplex method with shadow prices and sensitivity.

Input: `lp(max 3x1 + 5x2; x1 <= 4; 2x2 <= 12; 3x1 + 2x2 <= 18)`

Result

$$x_1 = 2$$
$$x_2 = 6$$
$$z_{max} = 36$$

Steps

- The linear program

  $\max z = 3x_1 + 5x_2$
  $x_1 \le 4$
  $2x_2 \le 12$
  $3x_1 + 2x_2 \le 18$
  $x_1, x_2 \ge 0$

- No non-negativity conditions are given – we assume $x_1, x_2 \ge 0$
- The standard form: a ≤ constraint gets a slack variable s added, a ≥ one gets it subtracted plus an artificial variable a, an equality only an artificial one. The starting basis consists of the s and a

  $x_1 + s_1 = 4$
  $2x_2 + s_2 = 12$
  $3x_1 + 2x_2 + s_3 = 18$

- Simplex tableau 1: x_2 enters (the most negative $z_j - c_j$), s_2 leaves (the smallest ratio b/a); the pivot is boxed.

	x_1	x_2	s_1	s_2	s_3	b	b/a
s_1	1	0	1	0	0	4	–
s_2	0	$\boxed{2}$	0	1	0	12	6
s_3	3	2	0	0	1	18	9
z	−3	−5	0	0	0	0	

- Simplex tableau 2: x_1 enters (the most negative $z_j - c_j$), s_3 leaves (the smallest ratio b/a); the pivot is boxed.

	x_1	x_2	s_1	s_2	s_3	b	b/a
s_1	1	0	1	0	0	4	4
x_2	0	1	0	$\frac{1}{2}$	0	6	–
s_3	$\boxed{3}$	0	0	−1	1	6	2
z	−3	0	0	$\frac{5}{2}$	0	30	

- Tableau 3: no $z_j - c_j$ is negative – the solution is optimal.

	x_1	x_2	s_1	s_2	s_3	b
s_1	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2
x_2	0	1	0	$\frac{1}{2}$	0	6
x_1	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2
z	0	0	0	$\frac{3}{2}$	1	36

- The optimal solution read from the last tableau (the non-basic variables are zero): $x_1 = 2,\ x_2 = 6,\quad z_{max} = 36$
- The dual program – a dual variable for each constraint, a constraint for each variable

  $\min w = 4y_1 + 12y_2 + 18y_3$
  $y_1 + 3y_3 \ge 3$
  $2y_2 + 2y_3 \ge 5$
  $y_1, y_2, y_3 \ge 0$

- The shadow prices (the optimal dual variables) from the last tableau: how much the optimum changes when a constraint's right side grows by 1. A constraint with slack ($b_i - a_i x > 0$) has the shadow price 0.

	$b_i - a_i x$	y_i
1	2	0
2	0	$\frac{3}{2}$
3	0	1

- The dual's optimum equals the primal's:
  $w = 12 \cdot \frac{3}{2} + 18 \cdot 1 = 36 = z$
- Sensitivity analysis – the ranges of the objective's coefficients in which the same solution stays optimal

	c_j	
x_1	3	$\langle 0;\ \frac{15}{2} \rangle$
x_2	5	$\langle 2;\ \infty)$

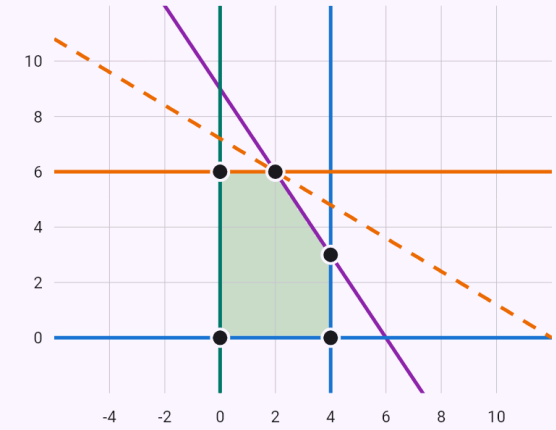
- The ranges of the right sides in which the same basis stays optimal (the shadow prices hold only there)

	b_i	
1	4	$\langle 2;\ \infty)$
2	12	$\langle 6;\ 18 \rangle$
3	18	$\langle 12;\ 24 \rangle$

- The graphical solution: the feasible region is a polygon and the optimum lies in a vertex. The objective's values at the vertices (★ the optimum)

$[x_1;\ x_2]$	z	
$[0;\ 0]$	0	
$[0;\ 6]$	30	
$[2;\ 6]$	36	★
$[4;\ 0]$	12	
$[4;\ 3]$	27	

Graph



$x \leq 4 \ \Leftrightarrow \ x \in (-\infty,\ 4]$
$2y \leq 12 \ \Leftrightarrow \ y \in (-\infty,\ 6]$
$3x + 2y \leq 18 \ \Leftrightarrow \ y \leq 9 - \frac{3x}{2}$
$x \geq 0 \ \Leftrightarrow \ x \in [0,\ \infty)$
$y \geq 0 \ \Leftrightarrow \ y \in [0,\ \infty)$
$z = 36 \ \Leftrightarrow \ 3x + 5y = 36$

The solutions are the curves' intersections in the shaded region (a full dot); a hollow dot fails the inequalities.

Intersections of the graphs:

● $(0, 0)$ ● $(0, 6)$ ● $(2, 6)$ ● $(4, 0)$ ● $(4, 3)$

A typed program – the graphical solution for two variables.

Input: `max 3x + 2y; x + y <= 4; x + 3y <= 6; x, y >= 0`

Result

$$x = 4$$
$$y = 0$$
$$z_{max} = 12$$

Steps

- The linear program

  $\max z = 3x + 2y$
  $x + y \le 4$
  $x + 3y \le 6$
  $x, y \ge 0$

- The standard form: a ≤ constraint gets a slack variable s added, a ≥ one gets it subtracted plus an artificial variable a, an equality only an artificial one. The starting basis consists of the s and a

  $x + y + s_1 = 4$
  $x + 3y + s_2 = 6$

- Simplex tableau 1: x enters (the most negative $z_j - c_j$), s_1 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	b	b/a
s_1	$\boxed{1}$	1	1	0	4	4
s_2	1	3	0	1	6	6
z	−3	−2	0	0	0	

- Tableau 2: no $z_j - c_j$ is negative – the solution is optimal.

	x	y	s_1	s_2	b
x	1	1	1	0	4
s_2	0	2	−1	1	2
z	0	1	3	0	12

- The optimal solution read from the last tableau (the non-basic variables are zero): $x = 4,\ y = 0,\quad z_{max} = 12$

- The dual program – a dual variable for each constraint, a constraint for each variable

  $\min v = 4u_1 + 6u_2$
  $u_1 + u_2 \ge 3$
  $u_1 + 3u_2 \ge 2$
  $u_1, u_2 \ge 0$

- The shadow prices (the optimal dual variables) from the last tableau: how much the optimum changes when a constraint's right side grows by 1. A constraint with slack ($b_i - a_i x > 0$) has the shadow price 0.

	$b_i - a_i x$	u_i
1	0	3
2	2	0

- The dual's optimum equals the primal's:
  $v = 4 \cdot 3 = 12 = z$

- Sensitivity analysis – the ranges of the objective's coefficients in which the same solution stays optimal

	c_j	
x	3	$\langle 2;\ \infty)$
y	2	$(-\infty;\ 3\rangle$

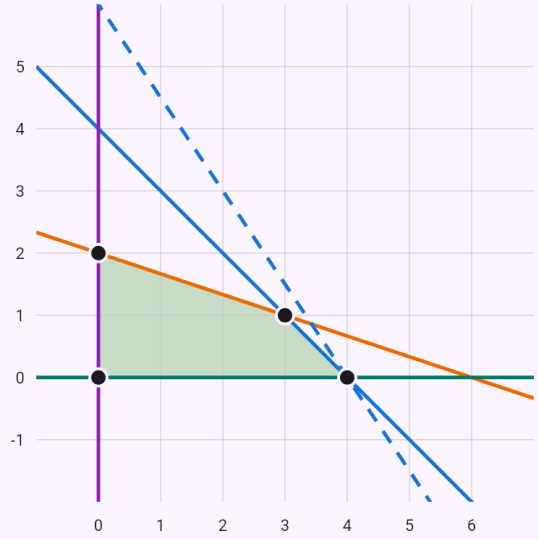
- The ranges of the right sides in which the same basis stays optimal (the shadow prices hold only there)

	b_i	
1	4	$\langle 0;\ 6\rangle$
2	6	$\langle 4;\ \infty)$

- The graphical solution: the feasible region is a polygon and the optimum lies in a vertex. The objective's values at the vertices (★ the optimum)

$[x;\ y]$	z	
$[0;\ 0]$	0	
$[0;\ 2]$	4	
$[3;\ 1]$	11	
$[4;\ 0]$	12	★

Graph



$x + y \le 4 \ \Leftrightarrow\ y \le 4 - x$
$x + 3y \le 6 \ \Leftrightarrow\ y \le 2 - \frac{x}{3}$
$x \ge 0 \ \Leftrightarrow\ x \in [0,\ \infty)$
$y \ge 0 \ \Leftrightarrow\ y \in [0,\ \infty)$
$z = 12 \ \Leftrightarrow\ 3x + 2y = 12$

The solutions are the curves' intersections in the shaded region (a full dot); a hollow dot fails the inequalities.

Intersections of the graphs:

● $(0, 0)$ ● $(0, 2)$ ● $(3, 1)$ ● $(4, 0)$

A minimum with ≥ constraints by the two-phase method.

Input: `lp(min 2x + 3y; x + y >= 4; x + 3y >= 6)`

Result

$$x = 3$$
$$y = 1$$
$$z_{min} = 9$$

Steps

- The linear program

  $\min z = 2x + 3y$

  $x + y \geq 4$

  $x + 3y \geq 6$

  $x, y \geq 0$

- No non-negativity conditions are given – we assume $x, y \geq 0$
- A minimum is found as the maximum of the opposite function: $\min z = -\max(-z), \quad -z = -2x - 3y$
- The standard form: a ≤ constraint gets a slack variable s added, a ≥ one gets it subtracted plus an artificial variable a, an equality only an artificial one. The starting basis consists of the s and a

  $x + y - s_1 + a_1 = 4$

  $x + 3y - s_2 + a_2 = 6$

- Phase one: the artificial variables aren't a real solution, so we drive them to zero first – we minimize their sum $\min w = a_1 + a_2$
- Simplex tableau 1: y enters (the most negative $z_j - c_j$), a_2 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	a_1	a_2	b	b/a
a_1	1	1	−1	0	1	0	4	4
a_2	1	$\boxed{3}$	0	−1	0	1	6	2
w	−2	−4	1	1	0	0	−10	

- Simplex tableau 2: x enters (the most negative $z_j - c_j$), a_1 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	a_1	a_2	b	b/a
a_1	$\boxed{\frac{2}{3}}$	0	−1	$\frac{1}{3}$	1	$-\frac{1}{3}$	2	3
y	$\frac{1}{3}$	1	0	$-\frac{1}{3}$	0	$\frac{1}{3}$	2	6
w	$-\frac{2}{3}$	0	1	$-\frac{1}{3}$	0	$\frac{4}{3}$	−2	

- Phase one ends with w = 0 – the artificial variables are zero and are dropped. Phase two continues with the real objective.
- Tableau 3: no $z_j - c_j$ is negative – the solution is optimal.

	x	y	s_1	s_2	b
x	1	0	$-\frac{3}{2}$	$\frac{1}{2}$	3
y	0	1	$\frac{1}{2}$	$-\frac{1}{2}$	1
z	0	0	$\frac{3}{2}$	$\frac{1}{2}$	−9

- The optimal solution read from the last tableau (the non-basic variables are zero): $x = 3,\ y = 1, \quad z_{min} = 9$
- The dual program – a dual variable for each constraint, a constraint for each variable

  $\max v = 4u_1 + 6u_2$

  $u_1 + u_2 \leq 2$

  $u_1 + 3u_2 \leq 3$

  $u_1, u_2 \geq 0$

- The shadow prices (the optimal dual variables) from the last tableau: how much the optimum changes when a constraint's right side grows by 1. A constraint with slack ($b_i - a_i x > 0$) has the shadow price 0.

	$b_i - a_i x$	u_i
1	0	$\frac{3}{2}$
2	0	$\frac{1}{2}$

- The dual's optimum equals the primal's:
  $v = 4 \cdot \frac{3}{2} + 6 \cdot \frac{1}{2} = 9 = z$
- Sensitivity analysis – the ranges of the objective's coefficients in which the same solution stays optimal

	c_j	
x	2	$\langle 1;\ 3 \rangle$
y	3	$\langle 2;\ 6 \rangle$

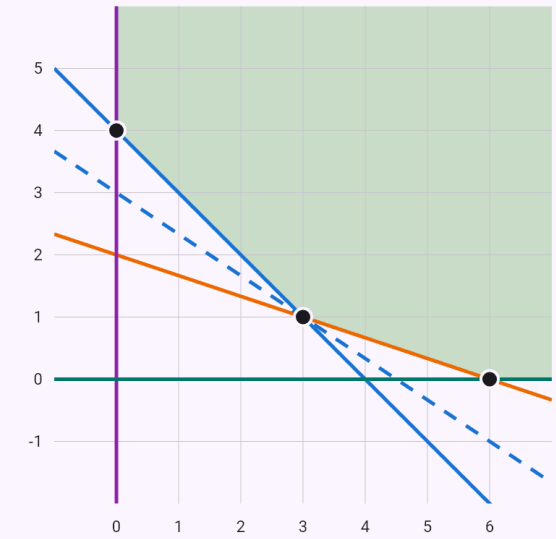
- The ranges of the right sides in which the same basis stays optimal (the shadow prices hold only there)

	b_i	
1	4	$\langle 2;\ 6 \rangle$
2	6	$\langle 4;\ 12 \rangle$

- The graphical solution: the feasible region is a polygon and the optimum lies in a vertex. The objective's values at the vertices (★ the optimum)

$[x;\ y]$	z	
$[0;\ 4]$	12	
$[3;\ 1]$	9	★
$[6;\ 0]$	12	

Graph



$x + y \geq 4 \ \Leftrightarrow \ y \geq 4 - x$
$x + 3y \geq 6 \ \Leftrightarrow \ y \geq 2 - \frac{x}{3}$
$x \geq 0 \ \Leftrightarrow \ x \in [0,\ \infty)$
$y \geq 0 \ \Leftrightarrow \ y \in [0,\ \infty)$
$z = 9 \ \Leftrightarrow \ 2x + 3y = 9$

The solutions are the curves' intersections in the shaded region (a full dot); a hollow dot fails the inequalities.

Intersections of the graphs:

● $(0, 4)$ ● $(3, 1)$ ● $(6, 0)$

The same program by the big M method.

Input: `lp(min 2x + 3y; x + y >= 4; x + 3y >= 6; method=bigm)`

Result

$$x = 3$$
$$y = 1$$
$$z_{min} = 9$$

Steps

- The linear program
  $\min z = 2x + 3y$
  $x + y \geq 4$
  $x + 3y \geq 6$
  $x, y \geq 0$
- No non-negativity conditions are given – we assume $x, y \geq 0$
- A minimum is found as the maximum of the opposite function: $\min z = -\max(-z), \quad -z = -2x - 3y$
- The standard form: a ≤ constraint gets a slack variable s added, a ≥ one gets it subtracted plus an artificial variable a, an equality only an artificial one. The starting basis consists of the s and a
  $x + y - s_1 + a_1 = 4$
  $x + 3y - s_2 + a_2 = 6$
- The big M method: the artificial variables enter the objective with a large penalty −M so that the simplex method pushes them out of the basis:
  $\max z' = -2x - 3y - Ma_1 - Ma_2$
- Simplex tableau 1: y enters (the most negative $z_j - c_j$), a_2 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	a_1	a_2	b
a_1	1	1	−1	0	1	0	4
a_2	1	$\boxed{3}$	0	−1	0	1	6
z	$-2M+2$	$-4M+3$	M	M	0	0	−10

- Simplex tableau 2: x enters (the most negative $z_j - c_j$), a_1 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	a_1	a_2
a_1	$\boxed{\frac{2}{3}}$	0	−1	$\frac{1}{3}$	1	$-\frac{1}{3}$
y	$\frac{1}{3}$	1	0	$-\frac{1}{3}$	0	$\frac{1}{3}$
z	$-\frac{2}{3}M+1$	0	M	$-\frac{1}{3}M+1$	0	$\frac{4}{3}M-1$

- Tableau 3: no $z_j - c_j$ is negative – the solution is optimal.

	x	y	s_1	s_2	a_1	a_2	b
x	1	0	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{3}{2}$	$-\frac{1}{2}$	3
y	0	1	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	1
z	0	0	$\frac{3}{2}$	$\frac{1}{2}$	$M-\frac{3}{2}$	$M-\frac{1}{2}$	−9

- The optimal solution read from the last tableau (the non-basic variables are zero): $x = 3,\ y = 1, \quad z_{min} = 9$
- The dual program – a dual variable for each constraint, a constraint for each variable
  $\max v = 4u_1 + 6u_2$
  $u_1 + u_2 \leq 2$
  $u_1 + 3u_2 \leq 3$
  $u_1, u_2 \geq 0$
- The shadow prices (the optimal dual variables) from the last tableau: how much the optimum changes when a constraint's right side grows by 1. A constraint with slack ($b_i - a_i x > 0$) has the shadow price 0.

	$b_i - a_i x$	u_i
1	0	$\frac{3}{2}$
2	0	$\frac{1}{2}$

- The dual's optimum equals the primal's:
  $v = 4 \cdot \frac{3}{2} + 6 \cdot \frac{1}{2} = 9 = z$
- Sensitivity analysis – the ranges of the objective's coefficients in which the same solution stays optimal

	c_j	
x	2	$\langle 3;\ 1 \rangle$
y	3	$\langle 6;\ 2 \rangle$

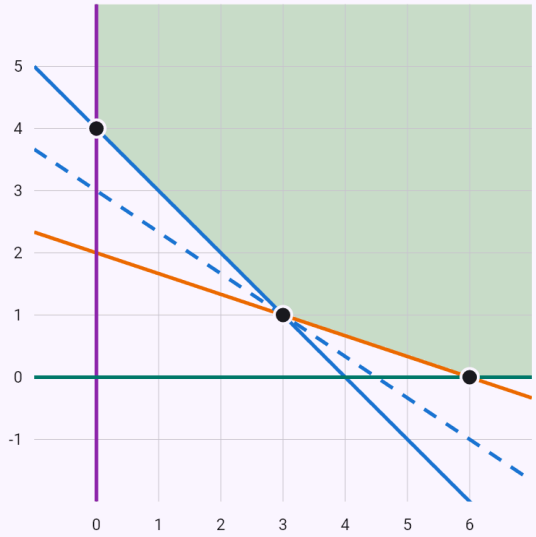
- The ranges of the right sides in which the same basis stays optimal (the shadow prices hold only there)

	b_i	
1	4	$\langle 2;\ 6 \rangle$
2	6	$\langle 4;\ 12 \rangle$

- The graphical solution: the feasible region is a polygon and the optimum lies in a vertex. The objective's values at the vertices (★ the optimum)

$[x;\ y]$	z	
$[0;\ 4]$	12	
$[3;\ 1]$	9	★
$[6;\ 0]$	12	

Graph



$x + y \geq 4 \ \Leftrightarrow \ y \geq 4 - x$
$x + 3y \geq 6 \ \Leftrightarrow \ y \geq 2 - \frac{x}{3}$
$x \geq 0 \ \Leftrightarrow \ x \in [0,\ \infty)$
$y \geq 0 \ \Leftrightarrow \ y \in [0,\ \infty)$
$z = 9 \ \Leftrightarrow \ 2x + 3y = 9$

The solutions are the curves' intersections in the shaded region (a full dot); a hollow dot fails the inequalities.

Intersections of the graphs:

● $(0, 4)$ ● $(3, 1)$ ● $(6, 0)$

An integer solution by branch and bound.

Input: `lp(max 5x + 4y; 6x + 4y <= 24; x + 2y <= 6; integer)`

Result

$$x = 4$$
$$y = 0$$
$$z_{max} = 20$$

Steps

- The linear program
  $\max z = 5x + 4y$
  $6x + 4y \le 24$
  $x + 2y \le 6$
  $x, y \ge 0,\ x, y \in \mathbb{Z}$
- No non-negativity conditions are given – we assume $x, y \ge 0$
- The standard form: a ≤ constraint gets a slack variable s added, a ≥ one gets it subtracted plus an artificial variable a, an equality only an artificial one. The starting basis consists of the s and a
  $6x + 4y + s_1 = 24$
  $x + 2y + s_2 = 6$
- Simplex tableau 1: x enters (the most negative $z_j - c_j$), s_1 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	b	b/a
s_1	$\boxed{6}$	4	1	0	24	4
s_2	1	2	0	1	6	6
z	-5	-4	0	0	0	

- Simplex tableau 2: y enters (the most negative $z_j - c_j$), s_2 leaves (the smallest ratio b/a); the pivot is boxed.

	x	y	s_1	s_2	b	b/a
x	1	$\frac{2}{3}$	$\frac{1}{6}$	0	4	6
s_2	0	$\boxed{\frac{4}{3}}$	$-\frac{1}{6}$	1	2	$\frac{3}{2}$
z	0	$-\frac{2}{3}$	$\frac{5}{6}$	0	20	

- Tableau 3: no $z_j - c_j$ is negative – the solution is optimal.

	x	y	s_1	s_2	b
x	1	0	$\frac{1}{4}$	$-\frac{1}{2}$	3
y	0	1	$-\frac{1}{8}$	$\frac{3}{4}$	$\frac{3}{2}$
z	0	0	$\frac{3}{4}$	$\frac{1}{2}$	21

- The optimal solution read from the last tableau (the non-basic variables are zero): $x = 3,\ y = \frac{3}{2},\quad z_{max} = 21$

- The program is an integer one – we use branch and bound. The solution without the integrality condition above is the tree's root P_0.
- The tree's nodes (depth first, the ≤ branch first): a fractional variable splits into two branches; × – no feasible solution, z ≤ z* – the branch can't beat the best solution found (pruned), ✓ – an integer solution.

		x, y	z	
P_0	–	$3,\ \frac{3}{2}$	21	$y \le 1,\ y \ge 2$
P_1	$y \le 1$	$\frac{10}{3},\ 1$	$\frac{62}{3}$	$x \le 3,\ x \ge 4$
P_2	$x \le 3$	$3,\ 1$	19	✓ $z^* = 19$
P_3	$x \ge 4$	$4,\ 0$	20	✓ $z^* = 20$
P_4	$y \ge 2$	$2,\ 2$	18	$z \le z^*$

- The best integer solution: $x = 4,\ y = 0,\quad z_{max} = 20$

Graph



Start

$P_0:\ x = 3,\ y = \frac{3}{2}$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

A transportation problem – Vogel's method and MODI.

Input: `transport(19 30 50 10; 70 30 40 60; 40 8 70 20; supply=7 9 18; demand=5 8 7 14;`

```
method=vogel)
```

Result

$$x_{11} = 5,\ x_{14} = 2,\ x_{22} = 2$$
$$x_{23} = 7,\ x_{32} = 6,\ x_{34} = 12$$
$$z_{min} = 743$$

Steps

- The unit costs, the suppliers' supplies a_i and the customers' demands b_j

	B_1	B_2	B_3	B_4	a_i
A_1	19	30	50	10	7
A_2	70	30	40	60	9
A_3	40	8	70	20	18
b_j	5	8	7	14	

- Supply equals demand – the problem is balanced:
  $\sum a_i = 34, \quad \sum b_j = 34$
- Vogel's approximation: in each step we compute the penalties – the difference of the two lowest costs in each row and column – and in the row or column with the largest penalty (boxed) we allocate to the cheapest cell.

k	A_1	A_2	A_3	B_1	B_2	B_3	B_4	
1	9	10	12	21	$\boxed{22}$	10	10	$(3,2)$
2	9	20	20	$\boxed{21}$		10	10	$(1,1)$
3	40	20	$\boxed{50}$			10	10	$(3,4)$
4	40	20				10	$\boxed{50}$	$(1,4)$
5		20				40	$\boxed{60}$	$(2,4)$
6		$\boxed{40}$				40		$(2,3)$

- The starting solution (the small number is the cost, the bold one the amount) – the cost
  $z = 19 \cdot 5 + 10 \cdot 2 + 40 \cdot 7 + 60 \cdot 2 + 8 \cdot 8 + 20 \cdot 10 = 779$

	B_1	B_2	B_3	B_4	a_i
A_1	$^{19}\mathbf{5}$	$^{30}.$	$^{50}.$	$^{10}\mathbf{2}$	7
A_2	$^{70}.$	$^{30}.$	$^{40}\mathbf{7}$	$^{60}\mathbf{2}$	9
A_3	$^{40}.$	$^{8}\mathbf{8}$	$^{70}.$	$^{20}\mathbf{10}$	18
b_j	5	8	7	14	

- MODI iteration 1: the potentials from $u_i + v_j = c_{ij}$ in the occupied cells ($u_1 = 0$), in the empty cells $d_{ij} = c_{ij} - u_i - v_j$ (in brackets). The most negative d = -18 is in the cell $(2,2)$ – it is occupied. The closed loop $(2,2)^+ \to (3,2)^- \to (3,4)^+ \to (2,4)^-$, we shift θ = 2 (the smallest amount in the cells marked −).

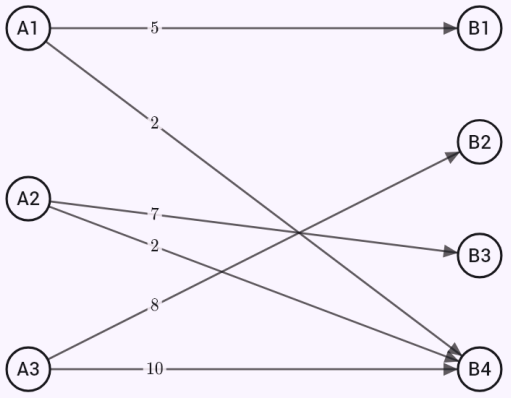
	B_1	B_2	B_3	B_4	a_i	u_i
A_1	$^{19}\mathbf{5}$	$^{30}(32)$	$^{50}(60)$	$^{10}\mathbf{2}$	7	0
A_2	$^{70}(1)$	$^{30}\boxed{-18}$	$^{40}\mathbf{7}$	$^{60}\mathbf{2}$	9	50
A_3	$^{40}(11)$	$^{8}\mathbf{8}$	$^{70}(70)$	$^{20}\mathbf{10}$	18	10
b_j	5	8	7	14		
v_j	19	-2	-10	10		

- No d_{ij} is negative – the transportation plan is optimal.

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	$^{19}\mathbf{5}$	$^{30}(32)$	$^{50}(42)$	$^{10}\mathbf{2}$	7	0
A_2	$^{70}(19)$	$^{30}\mathbf{2}$	$^{40}\mathbf{7}$	$^{60}(18)$	9	32
A_3	$^{40}(11)$	$^{8}\mathbf{6}$	$^{70}(52)$	$^{20}\mathbf{12}$	18	10
b_j	5	8	7	14		
v_j	19	-2	8	10		

- The total cost:
  $z = 19 \cdot 5 + 10 \cdot 2 + 30 \cdot 2 + 40 \cdot 7 + 8 \cdot 6 + 20 \cdot 12 = 743$

Graph



Start

$\sum$=779

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

An unbalanced transportation problem with a dummy customer.

Input: `transport(4 6 8; 5 7 3; supply=50 60; demand=30 40 20; method=northwest)`

Result

$x_{11} = 30,\ x_{12} = 20,\ x_{22} = 20$

$x_{23} = 20$

$z_{min} = 440$

Steps

- The unit costs, the suppliers' supplies a_i and the customers' demands b_j

	B_1	B_2	B_3	a_i
A_1	4	6	8	50
A_2	5	7	3	60
b_j	30	40	20	

- Supply exceeds demand ($\sum a_i = 110, \quad \sum b_j = 90$) – we add a dummy customer B_4^* with the demand 20 and zero costs (goods left at the suppliers).

- The starting solution by the north-west corner: from the top left cell we allocate as much as possible and move right or down (the small number is the cost, the bold one the amount), the cost
$z = 4 \cdot 30 + 6 \cdot 20 + 7 \cdot 20 + 3 \cdot 20 = 440$

	B_1	B_2	B_3	B_4^*	a_i
A_1	$^{4}\mathbf{30}$	$^{6}\mathbf{20}$	$^{8}.$	$^{0}.$	50
A_2	$^{5}.$	$^{7}\mathbf{20}$	$^{3}\mathbf{20}$	$^{0}\mathbf{20}$	60
b_j	30	40	20	20	

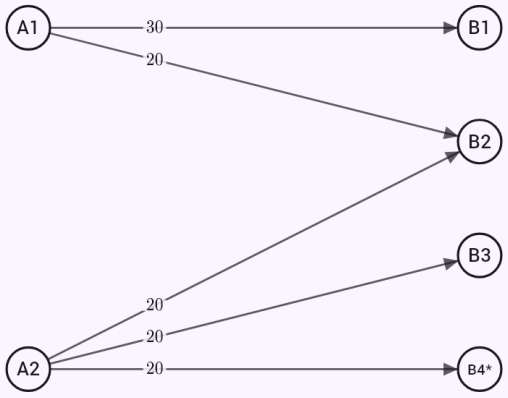
- No d_{ij} is negative – the transportation plan is optimal.

	B_1	B_2	B_3	B_4^*	a_i	u_i
A_1	$^{4}\mathbf{30}$	$^{6}\mathbf{20}$	$^{8}(6)$	$^{0}(1)$	50	0
A_2	$^{5}(0)$	$^{7}\mathbf{20}$	$^{3}\mathbf{20}$	$^{0}\mathbf{20}$	60	1
b_j	30	40	20	20		
v_j	4	6	2	-1		

- A d_{ij} is zero – there is another optimal solution with the same cost.

- The total cost:
$z = 4 \cdot 30 + 6 \cdot 20 + 7 \cdot 20 + 3 \cdot 20 = 440$

Graph



Start

$\sum = 440$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

Assigning workers to tasks at the least cost.

Input: `assignment(9 2 7 8; 6 4 3 7; 5 8 1 8; 7 6 9 4; goal=min)`

Result

$$R_1 \to S_2,\ R_2 \to S_1,\ R_3 \to S_3$$
$$R_4 \to S_4$$
$$z_{min} = 13$$

Steps

- The costs – the rows are workers (machines), the columns tasks

	S_1	S_2	S_3	S_4
R_1	9	2	7	8
R_2	6	4	3	7
R_3	5	8	1	8
R_4	7	6	9	4

- Each row's minimum $(2,\ 3,\ 1,\ 4)$ is subtracted

	S_1	S_2	S_3	S_4
R_1	7	0	5	6
R_2	3	1	0	4
R_3	4	7	0	7
R_4	3	2	5	0

- Each column's minimum $(3,\ 0,\ 0,\ 0)$ is subtracted

	S_1	S_2	S_3	S_4
R_1	4	0	5	6
R_2	0	1	0	4
R_3	1	7	0	7
R_4	0	2	5	0

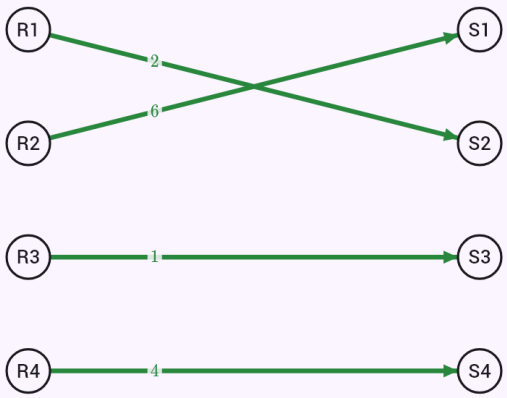
- 4 lines are needed to cover the zeros – we choose independent zeros, one in each row and column (boxed).

	S_1	S_2	S_3	S_4
R_1	4	$\boxed{0}$	5	6
R_2	$\boxed{0}$	1	0	4
R_3	1	7	$\boxed{0}$	7
R_4	0	2	5	$\boxed{0}$

- The assignment with the original costs – total $z_{min} = 2 + 6 + 1 + 4 = 13$

		c
R_1	S_2	2
R_2	S_1	6
R_3	S_3	1
R_4	S_4	4

Graph



An assignment with the largest profit.

Input: `assignment(9 2 7 8; 6 4 3 7; 5 8 1 8; 7 6 9 4; goal=max)`

Result

$$R_1 \to S_1,\ R_2 \to S_4,\ R_3 \to S_2$$
$$R_4 \to S_3$$
$$z_{max} = 33$$

Steps

- The costs – the rows are workers (machines), the columns tasks

	S_1	S_2	S_3	S_4
R_1	9	2	7	8
R_2	6	4	3	7
R_3	5	8	1	8
R_4	7	6	9	4

- We look for a maximum – we turn it into a minimum by subtracting each entry from the largest one, 9.

	S_1	S_2	S_3	S_4
R_1	0	7	2	1
R_2	3	5	6	2
R_3	4	1	8	1
R_4	2	3	0	5

- Each row's minimum $(0,\ 2,\ 1,\ 0)$ is subtracted

	S_1	S_2	S_3	S_4
R_1	0	7	2	1
R_2	1	3	4	0
R_3	3	0	7	0
R_4	2	3	0	5

- Each column's minimum $(0,\ 0,\ 0,\ 0)$ is subtracted

	S_1	S_2	S_3	S_4
R_1	0	7	2	1
R_2	1	3	4	0
R_3	3	0	7	0
R_4	2	3	0	5

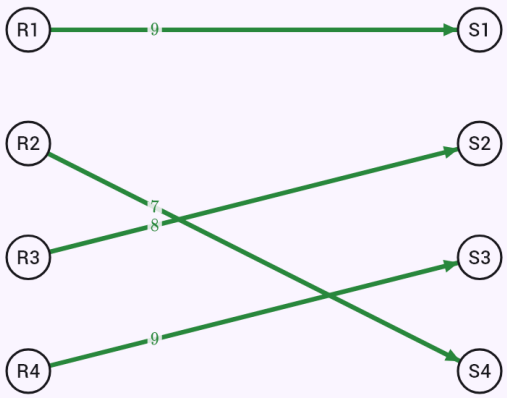
- 4 lines are needed to cover the zeros – we choose independent zeros, one in each row and column (boxed).

	S_1	S_2	S_3	S_4
R_1	$\boxed{0}$	7	2	1
R_2	1	3	4	$\boxed{0}$
R_3	3	$\boxed{0}$	7	0
R_4	2	3	$\boxed{0}$	5

- The assignment with the original costs – total $z_{max} = 9 + 7 + 8 + 9 = 33$

		c
R_1	S_1	9
R_2	S_4	7
R_3	S_2	8
R_4	S_3	9

Graph



A game with a saddle point – maximin equals minimax.

Input: `game(4 2 5; 3 1 6; 6 0 2)`

Result

$$v = 2, \quad (A_1, B_2)$$

Steps

- The payoff matrix (3×3): the rows are player A's strategies, the columns player B's, the numbers A's wins (B's losses). A wants as much as possible, B as little as possible.

	B_1	B_2	B_3
A_1	4	2	5
A_2	3	1	6
A_3	6	0	2

- We look for a saddle point: each row's minimum (what A can guarantee), each column's maximum (B loses no more). $\max\limits_i \min\limits_j a_{ij} = 2, \quad \min\limits_j \max\limits_i a_{ij} = 2$ – maximin equals minimax, the game has a saddle point.

	B_1	B_2	B_3	min
A_1	4	2	5	2
A_2	3	1	6	1
A_3	6	0	2	0
max	6	2	6	

- The optimal pure strategies (saddle points): (A_1, B_2); the value of the game $v = 2$

A 2 × 2 game without a saddle point – mixed strategies by the formula.

Input: `game(3 -1; -2 4)`

Result

$$v = 1, \quad p^* = \left(\frac{3}{5}, \frac{2}{5}\right), \quad q^* = \left(\frac{1}{2}, \frac{1}{2}\right)$$

Steps

- The payoff matrix (2×2): the rows are player A's strategies, the columns player B's, the numbers A's wins (B's losses). A wants as much as possible, B as little as possible.

	B_1	B_2
A_1	3	-1
A_2	-2	4

- We look for a saddle point: each row's minimum (what A can guarantee), each column's maximum (B loses no more).
$\max\limits_i \min\limits_j a_{ij} = -1, \quad \min\limits_j \max\limits_i a_{ij} = 3, \quad -1 < 3$ – there is no saddle point. The players will use mixed strategies – each pure strategy with some probability.

	B_1	B_2	min
A_1	3	-1	-1
A_2	-2	4	-2
max	3	4	

- A 2 × 2 game without a saddle point (strategies A_1, A_2 and B_1, B_2): A plays the first row with probability p so that the expected win is the same against both columns, B the first column with probability q likewise. For the matrix (a b; c d)

$$p = \frac{d-c}{a-b-c+d} = \frac{4-(-2)}{3-(-1)-(-2)+4} = \frac{3}{5}$$
$$q = \frac{d-b}{a-b-c+d} = \frac{4-(-1)}{3-(-1)-(-2)+4} = \frac{1}{2}$$
$$v = \frac{ad-bc}{a-b-c+d} = \frac{3\cdot 4-(-1)\cdot(-2)}{3-(-1)-(-2)+4} = 1$$

- The optimal mixed strategies: $p^* = \left(\frac{3}{5}, \frac{2}{5}\right)$, $q^* = \left(\frac{1}{2}, \frac{1}{2}\right)$; the value of the game $v = 1$

Dominated strategies crossed out one by one.

Input: `game(3 -1 4; -2 4 5; 1 -2 3)`

Result

$$v = 1, \quad p^* = \left(\frac{3}{5},\ \frac{2}{5},\ 0\right), \quad q^* = \left(\frac{1}{2}\right.$$

Steps

- The payoff matrix (3×3): the rows are player A's strategies, the columns player B's, the numbers A's wins (B's losses). A wants as much as possible, B as little as possible.

	B_1	B_2	B_3
A_1	3	-1	4
A_2	-2	4	5
A_3	1	-2	3

- We look for a saddle point: each row's minimum (what A can guarantee), each column's maximum (B loses no more).
$\max\limits_i \min\limits_j a_{ij} = -1, \quad \min\limits_j \max\limits_i a_{ij} = 3, \quad -1 < 3$ – there is no saddle point. The players will use mixed strategies – each pure strategy with some probability.

	B_1	B_2	B_3	min
A_1	3	-1	4	-1
A_2	-2	4	5	-2
A_3	1	-2	3	-2
max	3	4	5	

- Strategy A_3 is dominated by A_1 – that gives player A at least as much in every column. A won't choose it, we cross it out

	B_1	B_2	B_3
A_1	3	-1	4
A_2	-2	4	5

- Strategy B_3 is dominated by B_1 – with that one player B loses no more in any row. B won't choose it, we cross it out

	B_1	B_2
A_1	3	-1
A_2	-2	4

- A 2×2 game without a saddle point (strategies A_1, A_2 and B_1, B_2): A plays the first row with probability p so that the expected win is the same against both columns, B the first column with probability q likewise. For the matrix (a b; c d)

$$p = \frac{d-c}{a-b-c+d} = \frac{4-(-2)}{3-(-1)-(-2)+4} = \frac{3}{5}$$
$$q = \frac{d-b}{a-b-c+d} = \frac{4-(-1)}{3-(-1)-(-2)+4} = \frac{1}{2}$$
$$v = \frac{ad-bc}{a-b-c+d} = \frac{3\cdot 4-(-1)\cdot(-2)}{3-(-1)-(-2)+4} = 1$$

- The optimal mixed strategies: $p^* = \left(\frac{3}{5},\ \frac{2}{5},\ 0\right)$, $q^* = \left(\frac{1}{2},\ \frac{1}{2},\ 0\right)$; the value of the game $v = 1$

A 2 × 4 game graphically – the lower envelope's maximum.

Input: `game(2 3 1 5; 4 1 6 0)`

Result

$$v = \frac{17}{7}, \quad p^* = \left(\frac{5}{7}, \frac{2}{7}\right), \quad q^* = \left(0,\right.$$

Steps

- The payoff matrix (2×4): the rows are player A's strategies, the columns player B's, the numbers A's wins (B's losses). A wants as much as possible, B as little as possible.

	B_1	B_2	B_3	B_4
A_1	2	3	1	5
A_2	4	1	6	0

- We look for a saddle point: each row's minimum (what A can guarantee), each column's maximum (B loses no more). $\max_i \min_j a_{ij} = 1, \quad \min_j \max_i a_{ij} = 3, \quad 1 < 3$ – there is no saddle point. The players will use mixed strategies – each pure strategy with some probability.

	B_1	B_2	B_3	B_4	min
A_1	2	3	1	5	1
A_2	4	1	6	0	0
max	4	3	6	5	

- The graphical method: A plays the first row with probability p, the second with 1 − p. A's expected win against each of B's strategies is a line in p; B picks the lowest, so A looks for the highest point of the lines' lower envelope – the maximum $p^* = \frac{5}{7}, \quad v = \frac{17}{7}$. The lines

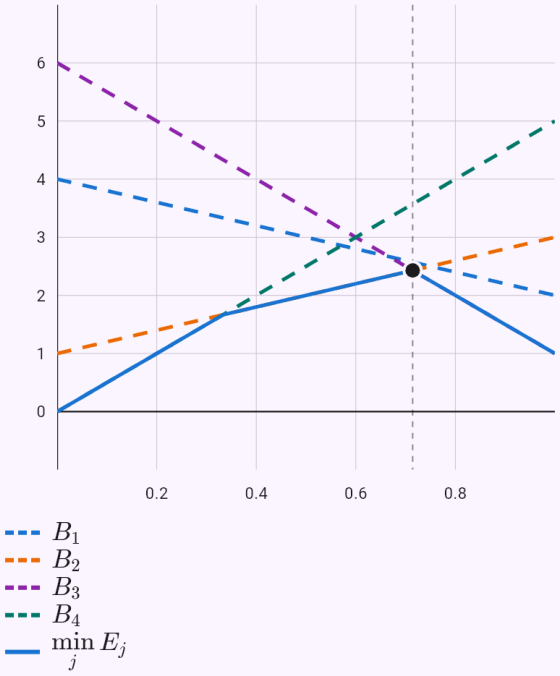
$$B_1: \; E = 2p + 4(1-p) = -2p + 4$$
$$B_2: \; E = 3p + (1-p) = 2p + 1$$
$$B_3: \; E = p + 6(1-p) = -5p + 6$$
$$B_4: \; E = 5p + 0(1-p) = 5p$$

- The optimum is where the lines B_2, B_3 cross. B mixes them with probabilities w and 1 − w so that A doesn't care which row to play:

$$2w + (-5)(1-w) = 0 \;\Rightarrow\; w = \frac{5}{7}$$

- The optimal mixed strategies: $p^* = \left(\frac{5}{7}, \frac{2}{7}\right)$, $q^* = \left(0, \frac{5}{7}, \frac{2}{7}, 0\right)$; the value of the game $v = \frac{17}{7}$

Graph



The lines are player A's expected wins against B's strategies as p changes; the highlighted lower envelope is what A can guarantee, the point is its maximum.

Rock, paper, scissors by linear programming – each strategy with probability 1/3.

Input: `game(0 -1 1; 1 0 -1; -1 1 0)`

Result

$$v = 0, \quad p^* = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \quad q^* = \left($$

Steps

- The payoff matrix (3×3): the rows are player A's strategies, the columns player B's, the numbers A's wins (B's losses). A wants as much as possible, B as little as possible.

	B_1	B_2	B_3
A_1	0	-1	1
A_2	1	0	-1
A_3	-1	1	0

- We look for a saddle point: each row's minimum (what A can guarantee), each column's maximum (B loses no more).
$\max_i \min_j a_{ij} = -1, \quad \min_j \max_i a_{ij} = 1, \quad -1 < 1$ – there is no saddle point. The players will use mixed strategies – each pure strategy with some probability.

	B_1	B_2	B_3	min
A_1	0	-1	1	-1
A_2	1	0	-1	-1
A_3	-1	1	0	-1
max	1	1	1	

- We add k = 2 to every entry to make them positive – the optimal strategies don't change, the value rises by k.
- A larger game becomes a linear program. Player B: max Σ y□ subject to Σ□ a'□□·y□ ≤ 1, y□ ≥ 0, where $y_j = \frac{q_j}{v'}, \quad v' = v + k$. Its dual problem belongs to player A.
- The linear program

$$\max z = y_1 + y_2 + y_3$$
$$2y_1 + y_2 + 3y_3 \le 1$$
$$3y_1 + 2y_2 + y_3 \le 1$$
$$y_1 + 3y_2 + 2y_3 \le 1$$
$$y_1, y_2, y_3 \ge 0$$

- The standard form: a ≤ constraint gets a slack variable s added, a ≥ one gets it subtracted plus an artificial variable a, an equality only an artificial one. The starting basis consists of the s and a

$$2y_1 + y_2 + 3y_3 + s_1 = 1$$
$$3y_1 + 2y_2 + y_3 + s_2 = 1$$
$$y_1 + 3y_2 + 2y_3 + s_3 = 1$$

- Simplex tableau 1: y_1 enters (the most negative $z_j - c_j$), s_2 leaves (the smallest ratio b/a); the pivot is boxed.

	y_1	y_2	y_3	s_1	s_2	s_3	b	b/a
s_1	2	1	3	1	0	0	1	$\frac{1}{2}$
s_2	$\boxed{3}$	2	1	0	1	0	1	$\frac{1}{3}$
s_3	1	3	2	0	0	1	1	1
z	-1	-1	-1	0	0	0	0	

- Simplex tableau 2: y_3 enters (the most negative $z_j - c_j$), s_1 leaves (the smallest ratio b/a); the pivot is boxed.

	y_1	y_2	y_3	s_1	s_2	s_3	b	b/a
s_1	0	$-\frac{1}{3}$	$\boxed{\frac{7}{3}}$	1	$-\frac{2}{3}$	0	$\frac{1}{3}$	$\frac{1}{7}$
y_1	1	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	0	$\frac{1}{3}$	1
s_3	0	$\frac{7}{3}$	$\frac{5}{3}$	0	$-\frac{1}{3}$	1	$\frac{2}{3}$	$\frac{2}{5}$
z	0	$-\frac{1}{3}$	$-\frac{2}{3}$	0	$\frac{1}{3}$	0	$\frac{1}{3}$	

- Simplex tableau 3: y_2 enters (the most negative $z_j - c_j$), s_3 leaves (the smallest ratio b/a); the pivot is boxed.

	y_1	y_2	y_3	s_1	s_2	s_3	b	b/a
y_3	0	$-\frac{1}{7}$	1	$\frac{3}{7}$	$-\frac{2}{7}$	0	$\frac{1}{7}$	–
y_1	1	$\frac{5}{7}$	0	$-\frac{1}{7}$	$\frac{3}{7}$	0	$\frac{2}{7}$	$\frac{2}{5}$
s_3	0	$\boxed{\frac{18}{7}}$	0	$-\frac{5}{7}$	$\frac{1}{7}$	1	$\frac{3}{7}$	$\frac{1}{6}$
z	0	$-\frac{3}{7}$	0	$\frac{2}{7}$	$\frac{1}{7}$	0	$\frac{3}{7}$	

- Tableau 4: no $z_j - c_j$ is negative – the solution is optimal.

	y_1	y_2	y_3	s_1	s_2	s_3	b
y_3	0	0	1	$\frac{7}{18}$	$-\frac{5}{18}$	$\frac{1}{18}$	$\frac{1}{6}$
y_1	1	0	0	$\frac{1}{18}$	$\frac{7}{18}$	$-\frac{5}{18}$	$\frac{1}{6}$
y_2	0	1	0	$-\frac{5}{18}$	$\frac{1}{18}$	$\frac{7}{18}$	$\frac{1}{6}$
z	0	0	0	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{2}$

- The optimal solution read from the last tableau (the non-basic variables are zero)

$$y_1 = \frac{1}{6},\ y_2 = \frac{1}{6},\ y_3 = \frac{1}{6}, \quad z_{max} = \frac{1}{2}$$

- The dual program – a dual variable for each constraint, a constraint for each variable

$$\min v = u_1 + u_2 + u_3$$
$$2u_1 + 3u_2 + u_3 \geq 1$$
$$u_1 + 2u_2 + 3u_3 \geq 1$$
$$3u_1 + u_2 + 2u_3 \geq 1$$
$$u_1, u_2, u_3 \geq 0$$

- The shadow prices (the optimal dual variables) from the last tableau: how much the optimum changes when a constraint's right side grows by 1. A constraint with slack ($b_i - a_i x > 0$) has the shadow price 0.

	$b_i - a_i x$	u_i
1	0	$\frac{1}{6}$
2	0	$\frac{1}{6}$
3	0	$\frac{1}{6}$

- The dual's optimum equals the primal's:

$$v = 1 \cdot \frac{1}{6} + 1 \cdot \frac{1}{6} + 1 \cdot \frac{1}{6} = \frac{1}{2} = z$$

- Sensitivity analysis – the ranges of the objective's coefficients in which the same solution stays optimal

	c_j	
y_1	1	$\langle \frac{4}{7};\ \frac{8}{5} \rangle$
y_2	1	$\langle \frac{4}{7};\ \frac{8}{5} \rangle$
y_3	1	$\langle \frac{4}{7};\ \frac{8}{5} \rangle$

- The ranges of the right sides in which the same basis stays optimal (the shadow prices hold only there)

	b_i	
1	1	$\langle \frac{4}{7};\ \frac{8}{5} \rangle$
2	1	$\langle \frac{4}{7};\ \frac{8}{5} \rangle$
3	1	$\langle \frac{4}{7};\ \frac{8}{5} \rangle$

- From the optimum: $v' = \frac{1}{\sum y_j} = 2, \quad v = v' - k = 0.$
  The strategies are the optimal y (for player A the shadow prices) multiplied by v'.
- The optimal mixed strategies: $p^* = \left(\frac{1}{3},\ \frac{1}{3},\ \frac{1}{3}\right)$, $q^* = \left(\frac{1}{3},\ \frac{1}{3},\ \frac{1}{3}\right)$; the value of the game $v = 0$

A bimatrix game (the battle of the sexes) – two pure and one mixed Nash equilibrium.

Input: `game(2 0; 0 1 | 1 0; 0 2)`

Result

$$(A_1, B_1); \quad (A_2, B_2); \quad p^* = \left(\frac{2}{3},\ \frac{1}{3}\right)$$

Steps

- A bimatrix game (2 × 2): each cell holds (A's payoff, B's payoff), and each player wants their own as large as possible.

	B_1	B_2
A_1	(2, 1)	(0, 0)
A_2	(0, 0)	(1, 2)

- Best responses: in each column we underline A's largest payoff, in each row we overline B's largest. The cells with both (boxed) are the Nash equilibria in pure strategies: (A_1, B_1), (A_2, B_2).

	B_1	B_2
A_1	$\boxed{(\underline{2},\ \overline{1})}$	(0, 0)
A_2	(0, 0)	$\boxed{(\underline{1},\ \overline{2})}$

- The mixed equilibrium: A chooses p so that B doesn't care which column to play, B chooses q so that A doesn't care which row to play (the expected payoffs $E_A = \frac{2}{3}, \quad E_B = \frac{2}{3}$)

$$p \cdot 1 + (1-p) \cdot 0 = p \cdot 0 + (1-p) \cdot 2 \ \Rightarrow\ p = \frac{2}{3}$$
$$q \cdot 2 + (1-q) \cdot 0 = q \cdot 0 + (1-q) \cdot 1 \ \Rightarrow\ q = \frac{1}{3}$$

Known limitations

- At most 12 variables and 15 constraints; integer programs up to 6 variables. All variables must be non-negative – free variables aren't supported yet.
- Transportation and assignment problems up to 8 × 8.
- Matrix games up to 6 × 6; a bimatrix game's mixed Nash equilibrium only for 2 × 2 games.

52. Graphs, networks and logistics

Problems on graphs and networks are on the home screen in the "Applications" section, field Logistics and networks: the shortest path (Dijkstra's, Bellman–Ford and Floyd–Warshall algorithms), the minimum spanning tree (Kruskal, Prim), the maximum flow and minimum cut, currency arbitrage as a search for a negative cycle and a project's critical path (CPM, PERT).

You enter the graph in a multi-line field, one edge per line – A-B 4 is an undirected edge of weight 4, A->B 4 a directed one, for a flow A->B 3/7 a given flow and capacity; without a number an edge has weight 1. Instead of the edges you can paste a matrix of weights row by row (-, ∞ or 0 means no edge), optionally with a line of the node names above it. Nodes can be named by a letter or a word (PHA, Boston). A project's activities are given by their name, duration and predecessors (D 5 B C), for PERT with three estimates (B 3/4/11 A).

For an arbitrage you can choose today's rate table of the ČNB or the ECB instead of your own rates – just list the currencies (CZK EUR USD GBP). With one table an arbitrage never comes out (the steps show why); with both tables at once each pair of currencies takes the better rate and a cycle above 1 can come out. The “Fill in the rates to edit” button puts one table's rates into the Exchange rates field – edit them, for example, by an exchange office's buying and selling rates.

The steps show a table of the iterations as in textbooks – Dijkstra's permanent and temporary labels, the Bellman–Ford passes, the Floyd–Warshall matrix after each node, Kruskal's sorted edges, the augmenting paths of a flow, the CPM forward and backward pass. Below the result is the graph, which you go through step by step with the arrows: red is the step's change, blue what is already decided, green the result.

Examples

The shortest path by Dijkstra's algorithm.

Input: `shortestpath(A-B 4; A-C 2; B-C 1; B-D 5; C-D 8; C-E 10; D-E 2; D-F 6; E-F 2; from=A;`

`to=F; method=dijkstra)`

Result

$$d(\mathrm{A},\mathrm{F}) = 12$$

$$\mathrm{A} \to \mathrm{C} \to \mathrm{B} \to \mathrm{D} \to \mathrm{E} \to \mathrm{F}$$

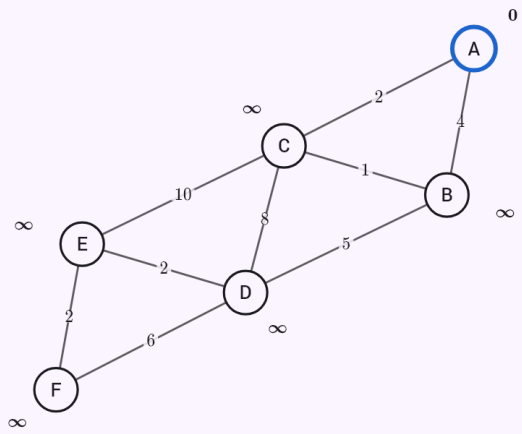
Steps

- An undirected graph with 6 nodes and 9 edges.
- Dijkstra's algorithm: in each step the node with the smallest temporary label becomes permanent (boxed) and its neighbours' labels are lowered if a shorter path leads through it (bold); the subscript is the predecessor.

		A	B	C	D	E	F
0		0	∞	∞	∞	∞	∞
1	A	$\boxed{0}$	$\mathbf{4_A}$	$\mathbf{2_A}$	∞	∞	∞
2	C		$\mathbf{3_C}$	$\boxed{2_A}$	$\mathbf{10_C}$	$\mathbf{12_C}$	∞
3	B		$\boxed{3_C}$		$\mathbf{8_B}$	12_C	∞
4	D				$\boxed{8_B}$	$\mathbf{10_D}$	$\mathbf{14_D}$
5	E					$\boxed{10_D}$	$\mathbf{12_E}$
6	F						$\boxed{12_E}$

- The path read backwards along the predecessors: $\mathrm{A} \to \mathrm{C} \to \mathrm{B} \to \mathrm{D} \to \mathrm{E} \to \mathrm{F}$, its length $2 + 1 + 5 + 2 + 2 = 12$

Graph



Start
$d(\mathrm{A}) = 0$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

A negative edge weight – the Bellman–Ford algorithm.

Input: `shortestpath(S->A 4; S->B 5; B->A -2; A->C 3; C->D 2; B->D 6; from=S; to=D; method=bellmanford)`

Result

$$d(\mathrm{S},\mathrm{D}) = 8$$

$$\mathrm{S} \to \mathrm{B} \to \mathrm{A} \to \mathrm{C} \to \mathrm{D}$$

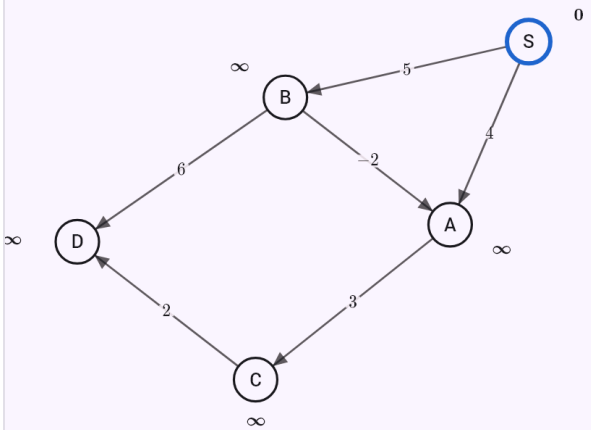
Steps

- A directed graph with 5 nodes and 6 edges.
- The Bellman–Ford algorithm: in each pass we try to shorten the path through every edge, $d(v) = \min\left(d(v), d(u) + w(u,v)\right)$; the changed labels are bold, the subscript is the predecessor.

	S	A	B	C	D
0	0	∞	∞	∞	∞
1	0	$\mathbf{3_B}$	$\mathbf{5_S}$	$\mathbf{6_A}$	$\mathbf{8_C}$

- In pass 2 no label changed any more – the distances are final.
- The path read backwards along the predecessors: $\mathrm{S} \to \mathrm{B} \to \mathrm{A} \to \mathrm{C} \to \mathrm{D}$, its length $5 - 2 + 3 + 2 = 8$

Graph



Start
$d(\mathrm{S}) = 0$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

The distances between all pairs of nodes.

Input: `shortestpath(A->B 3; B->C -1; A->C 4; C->A 2; C->D 2; D->B 1; method=floyd)`

Result

$$D = \begin{array}{c|cccc} & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{A} & 0 & 3 & 2 & 4 \\ \mathrm{B} & 1 & 0 & -1 & 1 \\ \mathrm{C} & 2 & 3 & 0 & 2 \\ \mathrm{D} & 2 & 1 & 0 & 0 \end{array}$$

Steps

- A directed graph with 4 nodes and 6 edges.
- The Floyd–Warshall algorithm starts with the matrix of the edges' weights (∞ – no edge)

$$D^{(0)} = \begin{array}{c|cccc} & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{A} & 0 & 3 & 4 & \infty \\ \mathrm{B} & \infty & 0 & -1 & \infty \\ \mathrm{C} & 2 & \infty & 0 & 2 \\ \mathrm{D} & \infty & 1 & \infty & 0 \end{array}$$

- We allow the node A as a stop: $d(i,j) = \min\left(d(i,j), d(i,k) + d(k,j)\right)$; the shortened distances are bold.

$$D^{(1)} = \begin{array}{c|cccc} & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{A} & 0 & 3 & 4 & \infty \\ \mathrm{B} & \infty & 0 & -1 & \infty \\ \mathrm{C} & 2 & \mathbf{5} & 0 & 2 \\ \mathrm{D} & \infty & 1 & \infty & 0 \end{array}$$

- We allow the node B as a stop: $d(i,j) = \min\left(d(i,j), d(i,k) + d(k,j)\right)$; the shortened distances are bold.

$$D^{(2)} = \begin{array}{c|cccc} & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{A} & 0 & 3 & \mathbf{2} & \infty \\ \mathrm{B} & \infty & 0 & -1 & \infty \\ \mathrm{C} & 2 & 5 & 0 & 2 \\ \mathrm{D} & \infty & 1 & \mathbf{0} & 0 \end{array}$$

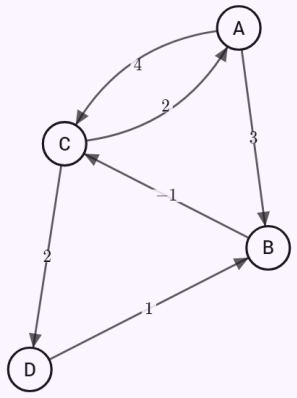
- We allow the node C as a stop: $d(i,j) = \min\left(d(i,j), d(i,k) + d(k,j)\right)$; the shortened distances are bold.

$$D^{(3)} = \begin{array}{c|cccc} & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{A} & 0 & 3 & 2 & \mathbf{4} \\ \mathrm{B} & \mathbf{1} & 0 & -1 & \mathbf{1} \\ \mathrm{C} & 2 & 5 & 0 & 2 \\ \mathrm{D} & \mathbf{2} & 1 & 0 & 0 \end{array}$$

- We allow the node D as a stop: $d(i,j) = \min\left(d(i,j), d(i,k) + d(k,j)\right)$; the shortened distances are bold.

$$D^{(4)} = \begin{array}{c|cccc} & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{A} & 0 & 3 & 2 & 4 \\ \mathrm{B} & 1 & 0 & -1 & 1 \\ \mathrm{C} & 2 & \mathbf{3} & 0 & 2 \\ \mathrm{D} & 2 & 1 & 0 & 0 \end{array}$$

Graph



Start
$D^{(0)}$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

The minimum spanning tree by Kruskal's algorithm.

Input: `spanningtree(A-B 7; A-D 5; B-C 8; B-D 9; B-E 7; C-E 5; D-E 15; D-F 6; E-F 8; E-G 9;`

`F-G 11; method=kruskal)`

Result

$w = 39$

$\mathrm{AB},\ \mathrm{AD},\ \mathrm{BE}$
$\mathrm{CE},\ \mathrm{DF},\ \mathrm{EG}$

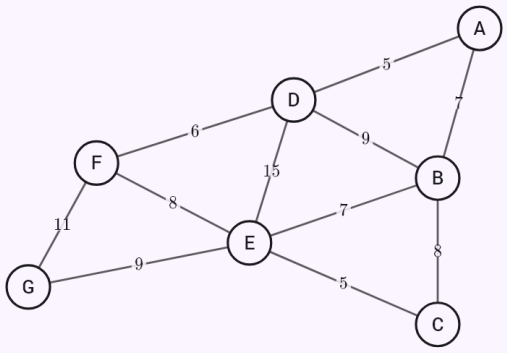
Steps

- An undirected graph with 7 nodes and 11 edges.
- Kruskal's algorithm: the edges sorted from the cheapest, each added (✓) unless it closes a cycle (×).

	w	
AD	5	✓
CE	5	✓
DF	6	✓
AB	7	✓
BE	7	✓
BC	8	×
EF	8	×
BD	9	×
EG	9	✓

- The total weight of the tree's edges:
  $7 + 5 + 7 + 5 + 6 + 9 = 39$

Graph



Start

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

The shortest network joining regional capitals by Prim's algorithm.

Input: `spanningtree(PHA-PLZ 93; PHA-LBC 110; PHA-HK 115; PHA-JI 128; PHA-BRN 205; LBC-HK 100; HK-OL 140; JI-BRN 88; BRN-OL 78; OL-OVA 100; BRN-OVA 170; PLZ-JI 210; start=PHA; method=prim)`

Result

$w = 697$

$\mathrm{PHA{-}PLZ},\ \mathrm{PHA{-}LBC},\ \mathrm{PHA{-}JI}$
$\mathrm{LBC{-}HK},\ \mathrm{JI{-}BRN},\ \mathrm{BRN{-}OL}$
$\mathrm{OL{-}OVA}$

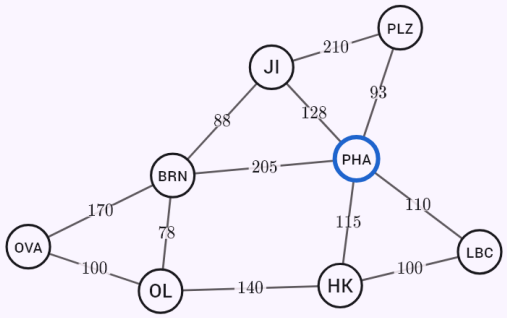
Steps

- An undirected graph with 8 nodes and 12 edges.
- Prim's algorithm from the node PHA: in each step we attach the cheapest edge leading from the tree to a node outside it.

		w
1	$\mathrm{PHA{-}PLZ}$	93
2	$\mathrm{PHA{-}LBC}$	110
3	$\mathrm{LBC{-}HK}$	100
4	$\mathrm{PHA{-}JI}$	128
5	$\mathrm{JI{-}BRN}$	88
6	$\mathrm{BRN{-}OL}$	78
7	$\mathrm{OL{-}OVA}$	100

- The total weight of the tree's edges:
  $93 + 110 + 128 + 100 + 88 + 78 + 100 = 697$

Graph



Start
PHA

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

The maximum flow and the minimum cut.

Input: `maxflow(S->A 16; S->B 13; B->A 4; A->C 12; C->B 9; B->D 14; D->C 7; C->T 20; D->T 4; from=S; to=T)`

Result

$f_{max} = 23$

$S = \{\mathrm{S}, \mathrm{A}, \mathrm{B}, \mathrm{D}\}$

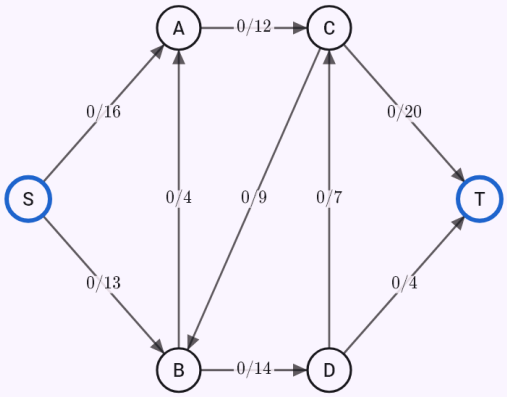
Steps

- We look for the maximum flow from the source S to the sink T, starting with a flow of value 0
- Augmenting path 1 (the shortest in the residual network): $\mathrm{S} \to \mathrm{A} \to \mathrm{C} \to \mathrm{T}$. The flow grows by the smallest residual capacity $\min(16, 12, 20) = 12$; its value is now 12
- Augmenting path 2 (the shortest in the residual network): $\mathrm{S} \to \mathrm{B} \to \mathrm{D} \to \mathrm{T}$. The flow grows by the smallest residual capacity $\min(13, 14, 4) = 4$; its value is now 16
- Augmenting path 3 (the shortest in the residual network): $\mathrm{S} \to \mathrm{B} \to \mathrm{D} \to \mathrm{C} \to \mathrm{T}$. The flow grows by the smallest residual capacity $\min(9, 10, 7, 8) = 7$; its value is now 23
- No more augmenting path – in the residual network the source reaches only the nodes $S = \{\mathrm{S}, \mathrm{A}, \mathrm{B}, \mathrm{D}\}$
- The flow along the edges (f) and their capacities (c)

	f	c
$\mathrm{S} \to \mathrm{A}$	12	16
$\mathrm{S} \to \mathrm{B}$	11	13
$\mathrm{B} \to \mathrm{A}$	0	4
$\mathrm{A} \to \mathrm{C}$	12	12
$\mathrm{C} \to \mathrm{B}$	0	9
$\mathrm{B} \to \mathrm{D}$	11	14
$\mathrm{D} \to \mathrm{C}$	7	7
$\mathrm{C} \to \mathrm{T}$	19	20
$\mathrm{D} \to \mathrm{T}$	4	4

- The edges leaving the set S form the minimum cut: $\mathrm{A} \to \mathrm{C}$, $\mathrm{D} \to \mathrm{C}$, $\mathrm{D} \to \mathrm{T}$. They are all full and their total capacity $12 + 7 + 4 = 23$ equals the maximum flow.

Graph



Start
$f = 0$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

A profitable cycle of exchanges of three currencies.

Input: `arbitrage(USD->EUR 0.9; EUR->GBP 0.8; GBP->USD 1.4; EUR->USD 1.1; GBP->EUR 1.24; USD-`

>GBP 0.7)

Result

$$\mathrm{USD} \to \mathrm{EUR} \to \mathrm{GBP} \to \mathrm{USD}$$

$$\prod r \approx 1.008 \; (+0.8\,\%)$$

Steps

- The rates and the edges' weights $w = -\ln r$

	r	$w = -\ln r$
$\mathrm{USD} \to \mathrm{EUR}$	0.9	0.10536
$\mathrm{EUR} \to \mathrm{GBP}$	0.8	0.22314
$\mathrm{GBP} \to \mathrm{USD}$	1.4	−0.33647
$\mathrm{EUR} \to \mathrm{USD}$	1.1	−0.09531
$\mathrm{GBP} \to \mathrm{EUR}$	1.24	−0.21511
$\mathrm{USD} \to \mathrm{GBP}$	0.7	0.35667

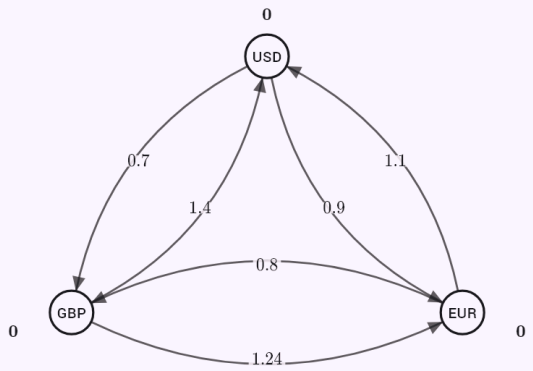
- A cycle makes a profit when the product of its rates is above 1 – in logarithms a negative cycle of the weights $r_1 r_2 \cdots r_k > 1 \iff \ln r_1 + \cdots + \ln r_k > 0 \iff (-\ln$

- The Bellman–Ford algorithm starts from all the currencies at once (all labels 0); the changed labels are bold, the subscript is the predecessor.

	USD	EUR	GBP
0	0	0	0
1	$\mathbf{-0.336472_{GBP}}$	$\mathbf{-0.215111_{GBP}}$	0
2	$\mathbf{-0.34444_{GBP}}$	$\mathbf{-0.231112_{USD}}$	$\mathbf{-0.007968}$
3	$\mathbf{-0.352409_{GBP}}$	$\mathbf{-0.23908_{USD}}$	$\mathbf{-0.015936}$

- The labels still change in the n-th pass – the predecessors lead to the negative cycle $\mathrm{USD} \to \mathrm{EUR} \to \mathrm{GBP} \to \mathrm{USD}$. The product of the rates $0.9 \cdot 0.8 \cdot 1.4 \approx 1.008$, a profit of $0.8\,\%$: 1000 units become 1008 after one round.

Graph



Start
$d = 0$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

Consistent rates as a matrix – no arbitrage.

Input: `arbitrage(CZK EUR USD; 1 0.04 0.0432; 25 1 1.08; 23.148148 0.925926 1)`

Result

No arbitrage – no cycle of exchanges makes a profit.

Steps

- The rates and the edges' weights $w = -\ln r$

	r	$w = -\ln r$
CZK → EUR	0.04	3.2189
CZK → USD	0.0432	3.1419
EUR → CZK	25	−3.2189
EUR → USD	1.08	−0.076961
USD → CZK	23.1481	−3.1419
USD → EUR	0.925926	0.076961

- A cycle makes a profit when the product of its rates is above 1 – in logarithms a negative cycle of the weights $r_1 r_2 \cdots r_k > 1 \iff \ln r_1 + \cdots + \ln r_k > 0 \iff (-\ln$

- The Bellman–Ford algorithm starts from all the currencies at once (all labels 0); the changed labels are bold, the subscript is the predecessor.

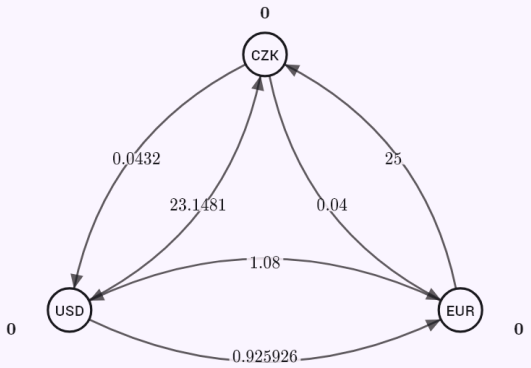
	CZK	EUR	USD
0	0	0	0
1	$\mathbf{-3.218876_{EUR}}$	0	$\mathbf{-0.076961_{EUR}}$

- In pass 2 no label changed any more – the distances are final.

- Even in the n-th pass no label changed – there is no negative cycle and so no arbitrage. The potentials p = −d are the logarithms of consistent "prices" of the currencies: every rate satisfies $\ln r \le p$(target) − p(source).

	$p = -d$
CZK	3.2189
EUR	0
USD	0.076961

Graph



Start
$d = 0$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

The rates of one table (ČNB) – never an arbitrage.

Input: `arbitrage(source=cnb; CZK EUR USD GBP)`

Result

No arbitrage – no cycle of exchanges makes a profit.

Steps

- Rates from the rate table: $\check{\mathrm{C}}\mathrm{NB},\ 11/27/2026$
- The table gives the value of one unit of each currency in the base currency CZK (k) – and its logarithm p = ln k

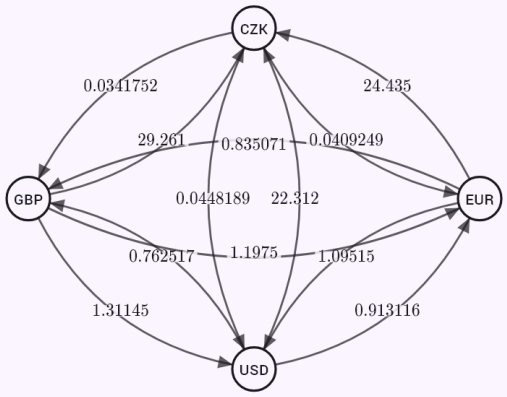
	k	$p = \ln k$
CZK	1	0
EUR	$1\,€ = 24.435\,\mathrm{CZK}$	3.196
USD	$1\,\$ = 22.312\,\mathrm{CZK}$	3.1051
GBP	$1\,\mathrm{GBP} = 29.261\,\mathrm{CZK}$	3.3763

- The rate between two currencies is a cross rate through the base currency: $r_{ij} = \dfrac{k_i}{k_j}$
- On every cycle the values k cancel and the product of the rates is exactly 1. The rates of one table ($\check{\mathrm{C}}\mathrm{NB}$) are consistent, so they never give an arbitrage – the logarithms p = ln k are the currencies' "prices" that show it.

$$r_{ij}\, r_{jl} \cdots r_{mi} = \frac{k_i}{k_j} \cdot \frac{k_j}{k_l} \cdots \frac{k_m}{k_i} = 1$$

- The ČNB and the ECB publish only mid (reference) rates, with no buying and selling rates. With an exchange office's rates (buying and selling) every cycle loses.

Graph



The ČNB's and the ECB's rates together – a tiny “gain” between reference rates.

Input: `arbitrage(source=cnb+ecb; CZK EUR USD GBP CHF)`

Result

$$\mathrm{USD} \to \mathrm{CHF} \to \mathrm{USD}$$

$$\prod r \approx 1.0009332 \; (+0.09332\,\%)$$

Steps

- Rates from the rate table:
  $\check{\mathrm{C}}\mathrm{NB},\ 11/27/2026;\quad \mathrm{ECB},\ 11/27/2026$
- For each pair of currencies we take the better of the two rates – the ČNB's or the ECB's (the table's last column). The other one wouldn't improve any cycle.
- The rates and the edges' weights $w = -\ln r$

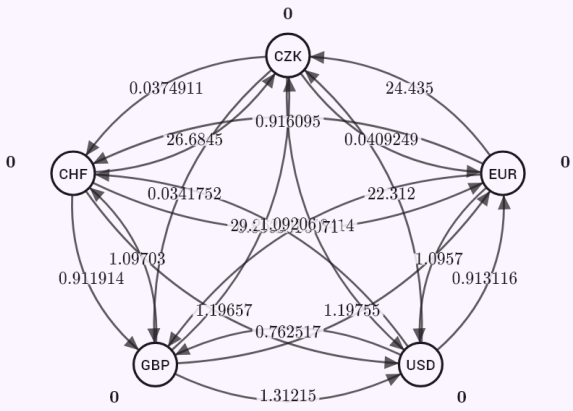
	r	$w = -\ln r$	
CZK → EUR	0.0409249	3.196	ČNB
CZK → USD	0.0448414	3.1046	ECB
CZK → GBP	0.0341752	3.3763	ČNB
CZK → CHF	0.0374911	3.2837	ČNB
EUR → CZK	24.435	−3.196	ČNB
EUR → USD	1.0957	−0.091393	ECB
EUR → GBP	0.835071	0.18024	ČNB
EUR → CHF	0.916095	0.087635	ČNB
USD → CZK	22.312	−3.1051	ČNB
USD → EUR	0.913116	0.090892	ČNB
USD → GBP	0.762517	0.27113	ČNB
USD → CHF	0.836501	0.17853	ČNB
GBP → CZK	29.2621	−3.3763	ECB
GBP → EUR	1.19755	−0.18028	ECB
GBP → USD	1.31215	−0.27167	ECB
GBP → CHF	1.09703	−0.092604	ČNB
CHF → CZK	26.6845	−3.2841	ECB
CHF → EUR	1.09206	−0.088066	ECB
CHF → USD	1.19657	−0.17946	ECB
CHF → GBP	0.911914	0.092209	ECB

- A cycle makes a profit when the product of its rates is above 1 – in logarithms a negative cycle of the weights
  $r_1 r_2 \cdots r_k > 1 \iff \ln r_1 + \cdots + \ln r_k > 0 \iff (-\ln$
- The Bellman–Ford algorithm starts from all the currencies at once (all labels 0); the changed labels are bold, the subscript is the predecessor.

	CZK	EUR	USD
0	0	0	0
1	$\mathbf{-3.376687_{CHF}}$	$\mathbf{-0.18067_{CHF}}$	$\mathbf{-0.272064}$
2	$\mathbf{-3.37762_{CHF}}$	$\mathbf{-0.181603_{CHF}}$	$\mathbf{-0.272996}$
3	$\mathbf{-3.378552_{CHF}}$	$\mathbf{-0.182536_{CHF}}$	$\mathbf{-0.273929}$
4	$\mathbf{-3.379485_{CHF}}$	$\mathbf{-0.183468_{CHF}}$	$\mathbf{-0.274862}$
5	$\mathbf{-3.380418_{CHF}}$	$\mathbf{-0.184401_{CHF}}$	$\mathbf{-0.275795}$

- The labels still change in the n-th pass – the predecessors lead to the negative cycle $\mathrm{USD} \to \mathrm{CHF} \to \mathrm{USD}$. The product of the rates $0.836501 \cdot 1.19657 \approx 1.0009332$, a profit of $0.09332\,\%$: 1000 units become 1000.933 after one round.
- The cycle only comes out between the reference rates of two banks, which are fixed at different times and rounded differently. Nobody trades at these rates – the gain can't be had in practice.

Graph



Start
$d = 0$

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

The critical path and the activities' slack.

Input: `cpm(A 3; B 4 A; C 2 A; D 5 B C; E 3 C; F 2 D E)`

Result

$T = 14$

$\mathrm{A} \to \mathrm{B} \to \mathrm{D} \to \mathrm{F}$

Steps

- The activities, their durations and predecessors

	t	
A	3	–
B	4	A
C	2	A
D	5	B, C
E	3	C
F	2	D, E

- The forward pass: an activity starts at the earliest when all its predecessors finish (ES = the largest EF of the predecessors) and finishes at EF = ES + t.

	ES	$\mathrm{EF} = \mathrm{ES} + t$
A	0	3
B	3	7
C	3	5
D	$\max(7, 5) = 7$	12
E	5	8
F	$\max(12, 8) = 12$	14

- The project's duration is the latest of the earliest finishes: $T = 14$
- The backward pass: an activity must finish at the latest when its successors must start (LF = the smallest LS of the successors) and start at the latest at LS = LF − t.

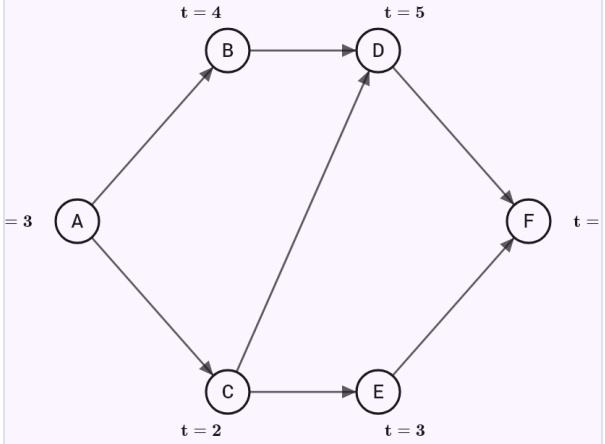
	LF	$\mathrm{LS} = \mathrm{LF} - t$
F	14	12
E	12	9
D	12	7
C	$\min(7, 9) = 7$	5
B	7	3
A	$\min(3, 5) = 3$	0

- The slack of each activity; the critical activities (★) have none

	$R = \mathrm{LS} - \mathrm{ES}$	
A	0	★
B	0	★
C	2	
D	0	★
E	4	
F	0	★

- The critical path – a delay of any of its activities delays the whole project: $\mathrm{A} \to \mathrm{B} \to \mathrm{D} \to \mathrm{F}$

Graph



Start

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

PERT – the probability of finishing the project by a deadline.

Input: `cpm(A 2/3/4; B 3/4/11 A; C 1/2/3 A; D 4/5/6 B C; T=15)`

Result

$T = 13$

$\mathrm{A} \to \mathrm{B} \to \mathrm{D}$

$P(T \le 15) \approx 0.9214$

Steps

- The activities, their durations and predecessors

	t	
A	3	–
B	5	A
C	2	A
D	5	B, C

- PERT: the expected durations and variances from the three estimates

	a	m	b	$t = \frac{a+4m+b}{6}$	$\sigma^2 = \left(\frac{b-a}{6}\right)^2$
A	2	3	4	3	0.1111
B	3	4	11	5	1.778
C	1	2	3	2	0.1111
D	4	5	6	5	0.1111

- The forward pass: an activity starts at the earliest when all its predecessors finish (ES = the largest EF of the predecessors) and finishes at EF = ES + t.

	ES	$\mathrm{EF} = \mathrm{ES} + t$
A	0	3
B	3	8
C	3	5
D	$\max(8, 5) = 8$	13

- The project's duration is the latest of the earliest finishes: $T = 13$
- The backward pass: an activity must finish at the latest when its successors must start (LF = the smallest LS of the successors) and start at the latest at LS = LF − t.

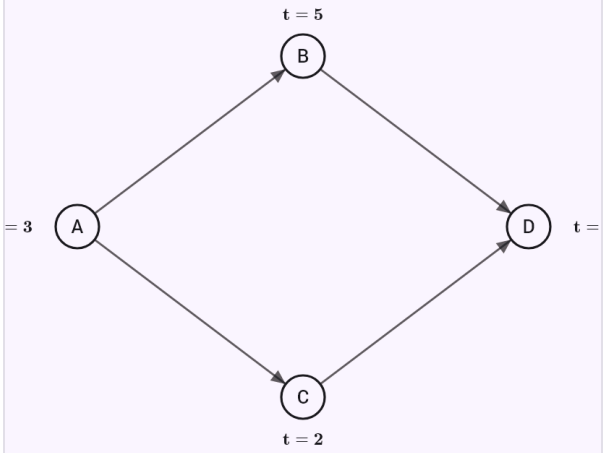
	LF	$\mathrm{LS} = \mathrm{LF} - t$
D	13	8
C	8	6
B	8	3
A	$\min(3, 6) = 3$	0

- The slack of each activity; the critical activities (★) have none

	$R = \mathrm{LS} - \mathrm{ES}$	
A	0	★
B	0	★
C	3	
D	0	★

- The critical path – a delay of any of its activities delays the whole project: $\mathrm{A} \to \mathrm{B} \to \mathrm{D}$
- The variance of the project's duration is the sum of the variances on the critical path: $\sigma_T^2 = 0.1111 + 1.778 + 0.1111 = 2, \quad \sigma_T \approx 1.414$
- The duration is approximately normally distributed – the probability of finishing by the deadline: $P(T \le 15) = \Phi\left(\frac{15-13}{1.414}\right) = \Phi(1.41) \approx 0.9214$

Graph



Start

Use the arrows to go through the algorithm step by step. Red is this step's change, blue what is already decided, green the result. Next to the nodes are their labels (distance, earliest and latest times).

Known limitations

- A graph may have at most 30 nodes and 200 edges; the Floyd–Warshall matrices node by node are shown up to 6 nodes, for larger graphs only the final matrix.
- The arbitrage search finds one profitable cycle (not necessarily the most profitable one); differences below 0.0001 % count as rounding of the rates.

- The ČNB and the ECB publish only mid (reference) rates, which nobody trades at; the “gain” from both tables at once is in hundredths of a percent and can't be had in practice.
- With both tables only the currencies both banks publish (neither publishes the Russian rouble).
- Vehicle routing, the knapsack problem, inventory and queues are coming; linear programming, transportation and assignment have a chapter of their own.

53. Psychometrics

Psychometric computations for interpreting test results (after the textbook "Clinical psychometrics – from a number to an interpretation") are in the "Statistics and psychometrics" form – the "Applications" section of the home screen, the Psychology field. Pick the task, the scale and what is sought and fill in the measured values; the scale prefills the norm's mean and standard deviation (IQ 100; 15, T 50; 10, scaled 10; 3, z 0; 1, sten 5.5; 2, stanin 5; 2), "Custom" lets you enter another test's norm. The "?" icon explains where the formula comes from.

Help right in the app: the "?" icon of each task in the form explains what it computes and where the formula comes from. The result shows the steps one by one, and the technical terms in them (Cohen's kappa, the ROC curve, the confidence interval, the p-value, the percentile, the true score, the predictive values …) are underlined with dots – a tap on a term shows its short explanation in plain words, often with an example. The notes at the end of the steps say how to interpret the result and what to watch out for.

Converting standard scores: $z = (X - M)/SD$ and from it T, IQ, the scaled score, sten and stanin (rounded to a whole number within the scale) and the percentile. Score and percentile: the percentile rank $PR = 100 \cdot \Phi(z)$ assuming the normal distribution, and back the score of a given percentile.

The standard error of measurement $SEM = SD \cdot \sqrt{(1 - r)}$ from the reliability r. The confidence interval around the observed score $X \pm z \cdot SEM$ at 80, 90, 95 or 99 % ($z = 1.28; 1.645; 1.96; 2.576$). The estimated true score $T' = M + r(X - M)$ – regression to the mean – with the interval around T' and the standard error of estimation $SEE = SD \cdot \sqrt{(r(1 - r))}$.

The significance of a difference of two scores of one scale (WAIS indexes, say): the standard error of the difference $SE_diff = SD \cdot \sqrt{(2 - r_{11} - r_{22})}$ and the critical difference $z \cdot SE_diff$. A test's predictive values by Bayes' theorem: PPV and NPV from the sensitivity, specificity and prevalence – or from a 2×2 table (true and false positives and negatives), optionally with the prevalence of the population the test is used in – with the likelihood ratios LR^+ and LR^-.

The Spearman–Brown formula (in the same form) gives the reliability of a test lengthened k times, $r' = k \cdot r / (1 + (k - 1) \cdot r)$, or the other way round how many times to lengthen a test for a target reliability – given the current number of items, also the items needed. Bonferroni's correction gives the stricter level $\alpha' = \alpha/m$ for m comparisons, with the probability of a false significant result without a correction and Šidák's correction.

Reliability and agreement from measured data come from the chips of data(…) or of the grid's “As data” (pasted from Excel, a CSV file): Cronbach's alpha – items in columns, respondents in rows – with an item analysis (the item's correlation with the rest of the test, alpha if the item is deleted); the ICC – raters in columns, subjects in rows – all six forms by Shrout and Fleiss from an analysis-of-variance table;

Cohen's kappa from two columns of ratings (categories as numbers) or from a square agreement table; the ROC curve and AUC from a column of scores and a column of the state 0/1 (1 = the condition) by the Mann–Whitney U, with the cut-off by Youden's index and a graph of the ROC curve. Alpha, the ICC, kappa and the AUC come with a confidence interval (90, 95 or 99 % by a chip): alpha by Feldt, the ICC from the F distribution, kappa from the asymptotic standard error, the AUC by Hanley and McNeil. For ordered categories kappa is computed with linear or quadratic weights too. The ROC curve can also come from two columns of two groups' scores (the first positive, of any lengths), with the Mann–Whitney test added to the AUC. Three columns – two tests' scores and the state – are compared by DeLong's test, both curves in one graph.

The steps end with an interpretive note: percentiles are ordinal (not added or averaged), converting to a percentile holds only for a normally distributed score, a significant difference isn't necessarily unusual (the discrepancy's base rate), the predictive values depend on the prevalence. The graph shows the normal curve with the score, the interval or the critical region marked.

Examples

IQ 85 in all scales: z = −1, T = 40, scaled score 7.

Input: `model(scores; find=all[one]; X=85[one]; M=100[one]; SD=15[one]; scale=iq)`

Result

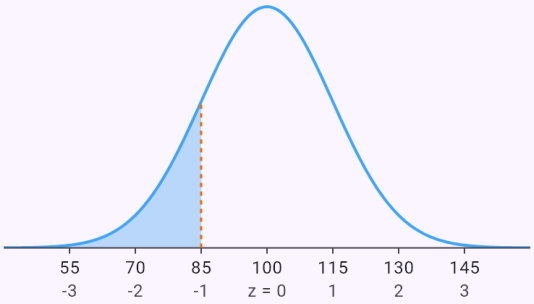
$$z = -1$$
$$T = 40$$
$$IQ = 85$$
$$SS = 7$$
$$\text{sten} = 4$$
$$\text{stanin} = 3$$
$$PR \approx 16$$

Steps

- Given: $X = 85, \quad M = 100, \quad SD = 15$
- Standard scores say how far a performance lies from the norm's mean, in standard deviations. First z = (X − M)/SD, then the other scale: T = 50 + 10z, IQ = 100 + 15z, scaled score = 10 + 3z, sten = 5.5 + 2z (1–10), stanin = 5 + 2z (1–9). The percentile is the percentage of the population scoring lower – valid only for a normally distributed score. The model's equation: $z = \frac{X - M}{SD}, \quad X' = M' + z \cdot SD'$
- Expressing the sought quantity: $z = \frac{X - M}{SD}$
- Substituting: $z = \frac{85 - 100}{15} = -1$
- Substituting: $T = 50 + 10z = 50 + 10 \cdot (-1) = 40$
- Substituting: $IQ = 100 + 15z = 100 + 15 \cdot (-1) = 85$
- Substituting: $SS = 10 + 3z = 10 + 3 \cdot (-1) = 7$
- Substituting: $\text{sten} = 5.5 + 2z = 5.5 + 2 \cdot (-1) = 3.5 \Rightarrow \text{sten} = 4$
- Substituting: $\text{stanin} = 5 + 2z = 5 + 2 \cdot (-1) = 3 \Rightarrow \text{stanin} = 3$
- Stens and stanins are whole numbers – the value is rounded and kept in the scale's range (1–10, 1–9).
- Substituting: $PR = 100 \cdot \Phi(z) = 100 \cdot \Phi(-1) \approx 15.87$
- Converting to a percentile holds only for a normally distributed score. For skewed tests (TMT's times, say) read the percentile off the norm table.
- Percentiles are an ordinal scale – their steps aren't equal (in the middle the 50th and 60th percentile are close in score, at the ends far apart). Percentiles are therefore not added, averaged or subtracted.

Graph

Normal density, the shaded area is the probability sought



$PR \approx 16 \quad X = 85$

IQ 130 is the 98th percentile.

Input: `model(percentile; find=PR[one]; X=130[one]; M=100[one]; SD=15[one]; scale=iq)`

Result

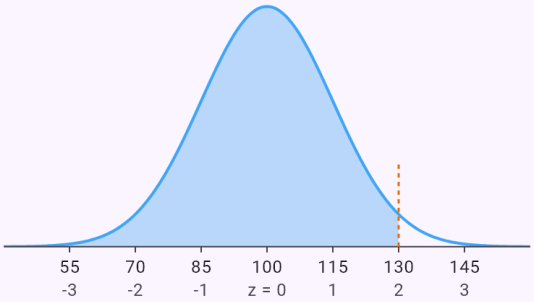
$$PR \approx 98$$

Steps

- Find PR; given: $X = 130, \quad M = 100, \quad SD = 15$
- The percentile (percentile rank) tells what percentage of the norm group scored lower. Assuming the normal distribution, it follows from the z-score by the distribution function: PR = 100·Φ(z); a score from a percentile by the quantile z = Φ⁻¹(PR/100). The model's equation: $PR = 100 \cdot \Phi\left(\frac{X - M}{SD}\right)$
- Expressing the sought quantity: $z = \frac{X - M}{SD}$
- Substituting: $z = \frac{130 - 100}{15} = 2$
- Substituting: $PR = 100 \cdot \Phi(2) \approx 97.72$
- Converting to a percentile holds only for a normally distributed score. For skewed tests (TMT's times, say) read the percentile off the norm table.
- Percentiles are an ordinal scale – their steps aren't equal (in the middle the 50th and 60th percentile are close in score, at the ends far apart). Percentiles are therefore not added, averaged or subtracted.

Graph

Normal density, the shaded area is the probability sought



$PR \approx 98 \quad X = 130$

The T-score of the 16th percentile.

Input: `model(percentile; find=X[one]; PR=16[pct]; M=50[one]; SD=10[one]; scale=t)`

Result

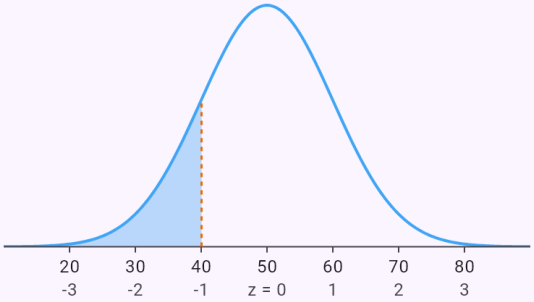
$$X \approx 40.1$$

Steps

- Find X; given: $PR = 16\,\%, \quad M = 50, \quad SD = 10$
- Converting the units: $PR = 16\,\% = 0.16$
- The percentile (percentile rank) tells what percentage of the norm group scored lower. Assuming the normal distribution, it follows from the z-score by the distribution function: PR = 100·Φ(z); a score from a percentile by the quantile z = Φ⁻¹(PR/100). The model's equation: $PR = 100 \cdot \Phi\left(\frac{X - M}{SD}\right)$
- Expressing the sought quantity: $z = \Phi^{-1}\left(\frac{PR}{100}\right), \quad X = M + z \cdot SD$
- Substituting $z = \Phi^{-1}(0.16) \approx -0.9945, \quad X = 50 + (-0.9945) \cdot 10$
- Converting to a percentile holds only for a normally distributed score. For skewed tests (TMT's times, say) read the percentile off the norm table.
- Percentiles are an ordinal scale – their steps aren't equal (in the middle the 50th and 60th percentile are close in score, at the ends far apart). Percentiles are therefore not added, averaged or subtracted.

Graph

Normal density, the shaded area is the probability sought



$PR = 16 \quad X \approx 40.1$

The standard error of measurement of a test with reliability 0.97.

Input: `model(sem; find=SEM[one]; SD=15[one]; r=0.97[one]; scale=iq)`

Result

$$SEM \approx 2.6$$

Steps

- Find SEM; given: $SD = 15, \quad r_{xx} = 0.97$
- No measurement is exact: the observed score X = the true score T + an error E. The standard error of measurement SEM = SD·√(1 − r) is how much the observed score typically differs from the true one; r is the test's reliability (test-retest, Cronbach's α). The higher the reliability, the smaller the error. The model's equation: $SEM = SD\sqrt{1 - r_{xx}}$
- The standard error of measurement: $SEM = 15\sqrt{1 - 0.97} \approx 2.598$

The 95% confidence interval around IQ 95.

Input: `model(interval; find=CI[one]; X=95[one]; SD=15[one]; r=0.97[one]; scale=iq;`

`level=95)`

Result

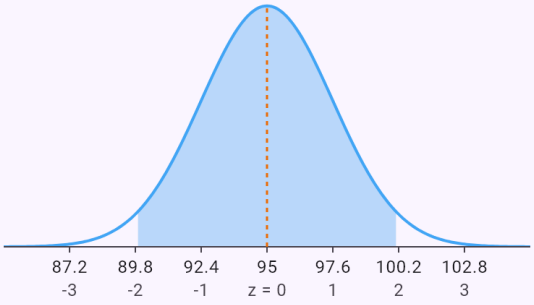
$$95\,\%:\ [89.9,\ 100.1]$$

Steps

- Find X; given: $X = 95,\quad SD = 15,\quad r_{xx} = 0.97$
- Around the observed score we mark the band where the true score lies with a given confidence: X ± z·SEM, with z = 1.28 (80 %), 1.645 (90 %), 1.96 (95 %) or 2.576 (99 %). A score is to be read as an interval, not an exact number. The model's equation
  $X \pm z_{1-\alpha/2} \cdot SEM$
  $SEM = SD\sqrt{1 - r_{xx}}$
- The standard error of measurement:
  $SEM = 15\sqrt{1 - 0.97} \approx 2.598$
- The standard normal critical value:
  $z_{1-\alpha/2} = z_{0.975} = 1.96 \quad (95\,\%)$
- Substituting:
  $95 \pm 1.96 \cdot 2.6 = 95 \pm 5.09 \Rightarrow [89.91,\ 100.1]$
- The interval is built around the observed score. More accurate (for extreme scores especially) is the interval around the estimated true score – the model "Estimated true score".

Graph

Normal density, the shaded area is the probability sought



$95\,\%:\ [89.9,\ 100.1] \quad X = 95$

The estimated true score of IQ 74 with reliability 0.92.

Input: `model(truescore; find=T[one]; X=74[one]; M=100[one]; SD=15[one]; r=0.92[one];`

`scale=iq; level=95)`

Result

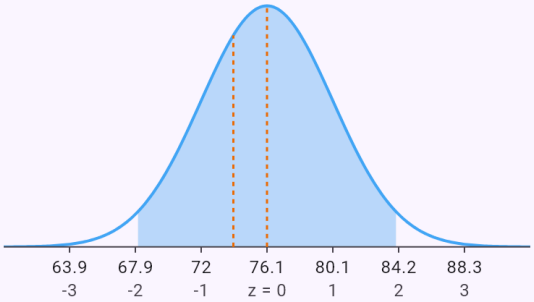
$$T' \approx 76.1$$
$$95\,\%:\ [68.1,\ 84.1]$$

Steps

- Find T'; given:
  $X = 74,\quad M = 100,\quad SD = 15,\quad r_{xx} = 0.92$
- For extreme scores the true score is estimated closer to the mean (regression to the mean): T' = M + r(X − M). The confidence interval is then built around T' with the standard error of estimation SEE = SD·√(r(1 − r)) – more accurate than around the observed score, especially with a low reliability. The model's equation
  $T' = M + r_{xx}(X - M)$
  $SEE = SD\sqrt{r_{xx}(1 - r_{xx})}$
- Substituting: $T' = 100 + 0.92 \cdot (74 - 100) = 76.08$
- The true score's estimate is closer to the mean than the observed score – measured again, an extreme result would likely come out less extreme.
- The standard error of the true score's estimate:
  $SEE = 15\sqrt{0.92 \cdot (1 - 0.92)} \approx 4.069$
- The standard normal critical value:
  $z_{1-\alpha/2} = z_{0.975} = 1.96 \quad (95\,\%)$
- Substituting:
  $T' \pm z \cdot SEE = 76.08 \pm 1.96 \cdot 4.07 \Rightarrow [68.1,\ 84.06]$

Graph

Normal density, the shaded area is the probability sought



$X = 74 \quad T'$

Two indexes 112 and 97 – is the difference significant?

Input: `model(difference; find=D[one]; X1=112[one]; X2=97[one]; SD=15[one]; r1=0.95[one];`

`r2=0.90[one]; scale=iq; level=95)`

Result

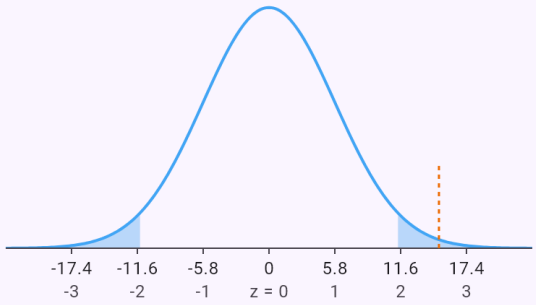
$$D = 15 \geq D_{crit} \approx 11.4$$

Steps

- Find D; given:
  $X_1 = 112, \quad X_2 = 97, \quad SD = 15, \quad r_{11} = 0.95, \quad r_{22} = 0.9$
- Do two scores (WAIS indexes, say) differ more than the measurement error would cause? The standard error of the difference SE_diff = SD·√(2 − r_{11} − r_{22}), the critical difference z·SE_diff. A difference reaching it is statistically significant – which doesn't mean it is unusual. The model's equation
  $SE_{diff} = SD\sqrt{2 - r_{11} - r_{22}}$
  $D_{crit} = z_{1-\alpha/2} \cdot SE_{diff}$
- The standard error of the difference:
  $SE_{diff} = 15\sqrt{2 - 0.95 - 0.9} \approx 5.809$
- The standard normal critical value:
  $z_{1-\alpha/2} = z_{0.975} = 1.96 \quad (95\,\%)$
- The critical difference: $D_{crit} = 1.96 \cdot 5.81 \approx 11.39$
- Substituting: $D = |X_1 - X_2| = |112 - 97| = 15 \geq 11.4$
- The difference reaches the critical value – it is statistically significant (not explained by the measurement error alone).
- A significant difference isn't necessarily unusual: 20–30 % of the norm sample often show index differences this large (the discrepancy's base rate). Its frequency in the norm matters for the interpretation too.

Graph

Normal density, the shaded area is the probability sought



$95\,\%$ $D = 15$

A screening with sensitivity 92.5 % and specificity 74.6 % at a prevalence of 2 %.

Input: `model(predictive; find=values[one]; sens=92.5[pct]; spec=74.6[pct]; p=2[pct];`

`input=rates)`

Result

$$PPV \approx 0.0692 = 6.92\,\%$$
$$NPV \approx 0.998 = 99.8\,\%$$

Steps

- Find PPV; given:
  $Se = 92.5\,\%, \quad Sp = 74.6\,\%, \quad p = 2\,\%$
- Converting the units:
  $Se = 92.5\,\% = 0.925, \quad Sp = 74.6\,\% = 0.746, \quad p = 2\,\% = 0.02$
- Sensitivity (Se) is the share of the ill the test detects, specificity (Sp) the share of the healthy it correctly rules out. What matters in practice is the probability of the disorder given a positive result – the positive predictive value PPV. By Bayes' theorem it depends on the prevalence p: PPV = Se·p / (Se·p + (1 − Sp)(1 − p)), the negative NPV = Sp(1 − p) / (Sp(1 − p) + (1 − Se)p). The model's equation
  $$PPV = \frac{Se \cdot p}{Se \cdot p + (1 - Sp)(1 - p)}$$
  $$NPV = \frac{Sp(1 - p)}{Sp(1 - p) + (1 - Se)p}$$
- Substituting
  $$PPV = \frac{0.925 \cdot 0.02}{0.925 \cdot 0.02 + 0.254 \cdot 0.98} = \frac{0.0185}{0.2674} \approx 0.0691$$
- Substituting
  $$NPV = \frac{0.746 \cdot 0.98}{0.746 \cdot 0.98 + 0.075 \cdot 0.02} = \frac{0.7311}{0.7326} \approx 0.998$$
- Likelihood ratios:
  $$LR^{+} = \frac{Se}{1 - Sp} \approx 3.642, \quad LR^{-} = \frac{1 - Se}{Sp} \approx 0.1005$$
- The predictive values depend on the prevalence: with a low prevalence most positive results are false positives even for a good test. A positive screening result needs confirming.

The predictive values from a 2×2 table.

Input: `model(predictive; find=values[one]; TP=37[one]; FP=10[one]; FN=3[one]; TN=50[one]; input=table)`

Result

$$PPV \approx 0.787 = 78.7\,\%$$
$$NPV \approx 0.943 = 94.3\,\%$$

Steps

- Find PPV; given:
  $TP = 37, \quad FP = 10, \quad FN = 3, \quad TN = 50$
- Sensitivity: $Se = \frac{TP}{TP + FN} = \frac{37}{40} = 0.925$
- Specificity: $Sp = \frac{TN}{TN + FP} = \frac{50}{60} \approx 0.8333$
- The prevalence in the sample:
  $p = \frac{TP + FN}{N} = \frac{40}{100} = 0.4$
- Sensitivity (Se) is the share of the ill the test detects, specificity (Sp) the share of the healthy it correctly rules out. What matters in practice is the probability of the disorder given a positive result – the positive predictive value PPV. By Bayes' theorem it depends on the prevalence p: PPV = Se·p / (Se·p + (1 − Sp)(1 − p)), the negative NPV = Sp(1 − p) / (Sp(1 − p) + (1 − Se)p). The model's equation
  $$PPV = \frac{Se \cdot p}{Se \cdot p + (1 - Sp)(1 - p)}$$
  $$NPV = \frac{Sp(1 - p)}{Sp(1 - p) + (1 - Se)p}$$
- Substituting
  $$PPV = \frac{0.925 \cdot 0.4}{0.925 \cdot 0.4 + 0.1667 \cdot 0.6} = \frac{0.37}{0.47} \approx 0.7872$$
- Substituting
  $$NPV = \frac{0.8333 \cdot 0.6}{0.8333 \cdot 0.6 + 0.075 \cdot 0.4} = \frac{0.5}{0.53} \approx 0.9434$$
- Likelihood ratios:
  $$LR^{+} = \frac{Se}{1 - Sp} = 5.55, \quad LR^{-} = \frac{1 - Se}{Sp} = 0.09$$
- The predictive values depend on the prevalence: with a low prevalence most positive results are false positives even for a good test. A positive screening result needs confirming.

Internal consistency of a four-item questionnaire.

Input: `data(P1 P2 P3 P4; 3 4 3 5; 2 2 3 2; 4 5 4 4; 5 4 5 5; 1 2 2 1; 3 3 4 3)`

Cronbach's alpha

Result

$\alpha \approx 0.9343$

$95\,\%:\ [0.765,\ 0.990]$

Steps

- Number of items (columns) k = 4, number of respondents (rows) 6
- The items' sample variances and their sum

  $s^2_{\mathrm{P1}} = 2,\ s^2_{\mathrm{P2}} \approx 1.467,\ s^2_{\mathrm{P3}} = 1.1,\ s^2_{\mathrm{P4}} \approx 2.667$

  $\sum_{i=1}^{4} s_i^2 = 2 + 1.467 + 1.1 + 2.667 = 7.233$

- Each respondent's total score (the sum of their answers): $15,\ 9,\ 17,\ 19,\ 6,\ 13$. The variance of the total scores: $s^2_X \approx 24.17$
- Cronbach's alpha – the internal consistency is excellent.

  $$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum s_i^2}{s_X^2}\right) = \frac{4}{3}\left(1 - \frac{7.233}{24.17}\right) \approx 0.9343$$

- Item analysis – the mean, the variance, the item's correlation with the rest of the test and alpha if the item is deleted

	$\bar{x}_i$	s_i^2	r_{it}	α_{-i}
P1	3	2	0.957	0.875
P2	3.333	1.467	0.825	0.922
P3	3.5	1.1	0.804	0.934
P4	3.333	2.667	0.869	0.918

- The 95 % confidence interval by Feldt – the ratio (1 − α)/(1 − α₀) has the F distribution with n − 1 and (n − 1)(k − 1) degrees of freedom

  $F_{0.975}(5;\ 15) = 3.576$

  $F_{0.025}(5;\ 15) = 0.1556$

  $$\alpha_L = 1 - (1-\alpha)F_{0.975} = 1 - 0.06575 \cdot 3.576$$

  $$\alpha_U = 1 - (1-\alpha)F_{0.025} = 1 - 0.06575 \cdot 0.155$$

- A rough guide to α: ≥ 0.9 excellent, ≥ 0.8 good, ≥ 0.7 acceptable internal consistency. About 0.7 is enough for research; decisions about an individual (diagnostics) call for at least 0.9.
- A high α doesn't mean the test measures a single trait, and it grows with the number of items alone. If α rises when an item is deleted (the table's last column), the item probably doesn't belong to the scale.

Agreement of four raters on six subjects (Shrout and Fleiss's example).

Input: `data(H1 H2 H3 H4; 9 2 5 8; 6 1 3 2; 8 4 6 8; 7 1 2 6; 10 5 6 9; 6 2 4 7)`

ICC (inter-rater reliability)

Result

$$ICC(1,1) \approx 0.1657 \quad [-0.133,\ 0.723]$$
$$ICC(1,4) \approx 0.4428 \quad [-0.884,\ 0.912]$$
$$ICC(2,1) \approx 0.2898 \quad [0.019,\ 0.761]$$
$$ICC(2,4) \approx 0.6201 \quad [0.071,\ 0.927]$$
$$ICC(3,1) \approx 0.7148 \quad [0.342,\ 0.946]$$
$$ICC(3,4) \approx 0.9093 \quad [0.676,\ 0.986]$$

Steps

- Number of rated subjects (rows) 6, number of raters (columns) k = 4, the grand mean $\bar{x} = 5.292$
- Analysis of variance – the sums of squares SS, degrees of freedom df and mean squares MS between subjects (R), between raters (C), residual (E) and within subjects (W)

	SS	df	MS
R	56.21	5	11.24
C	97.46	3	32.49
E	15.29	15	1.019
W	112.7	18	6.264
$\sum$	169	23	

- One-way model – each subject may be rated by different raters

$$ICC(1,1) = \frac{MS_R - MS_W}{MS_R + (k-1)MS_W} = \frac{11.24 - 6.264}{11.24 + 3 \cdot 6.264} \approx 0.1657$$
$$ICC(1,4) = \frac{MS_R - MS_W}{MS_R} = \frac{11.24 - 6.264}{11.24} \approx 0.4428$$

- Two-way model with random raters – absolute agreement

$$ICC(2,1) = \frac{MS_R - MS_E}{MS_R + (k-1)MS_E + \frac{k}{n}(MS_C - MS_E)} = \frac{11.24 - 1.019}{11.24 + 3 \cdot 1.019 + \frac{4}{6}(32.49 - 1.019)} \approx 0$$
$$ICC(2,4) = \frac{MS_R - MS_E}{MS_R + \frac{MS_C - MS_E}{n}} = \frac{11.24 - 1.019}{11.24 + \frac{32.49 - 1.019}{6}} \approx 0.6201$$

- Two-way model with fixed raters – consistency

$$ICC(3,1) = \frac{MS_R - MS_E}{MS_R + (k-1)MS_E} = \frac{11.24 - 1.019}{11.24 + 3 \cdot 1.019} \approx 0.7148$$
$$ICC(3,4) = \frac{MS_R - MS_E}{MS_R} = \frac{11.24 - 1.019}{11.24} \approx 0.9093$$

- The 95 % confidence intervals from the F distribution (Shrout and Fleiss) – the ratios of the mean squares and their bounds

$$F_W = \frac{MS_R}{MS_W} = 1.795$$
$$F_E = \frac{MS_R}{MS_E} = 11.03$$
$$F_{W,L} = \frac{F_W}{F_{0.975}(5;\ 18)} = 0.5307$$
$$F_{W,U} = F_W \cdot F_{0.975}(18;\ 5) = 11.42$$
$$F_{E,L} = \frac{F_E}{F_{0.975}(5;\ 15)} = 3.083$$
$$F_{E,U} = F_E \cdot F_{0.975}(15;\ 5) = 70.88$$
$$ICC(\cdot,1)_L = \frac{F_L - 1}{F_L + k - 1}$$
$$ICC(\cdot,k)_L = 1 - \frac{1}{F_L}$$

- All six ICCs with their 95 % confidence intervals

	ICC	95 %
$ICC(1,1)$	0.166	$[-0.133,\ 0.723]$
$ICC(1,4)$	0.443	$[-0.884,\ 0.912]$
$ICC(2,1)$	0.290	$[0.019,\ 0.761]$
$ICC(2,4)$	0.620	$[0.071,\ 0.927]$
$ICC(3,1)$	0.715	$[0.342,\ 0.946]$
$ICC(3,4)$	0.909	$[0.676,\ 0.986]$

- Which ICC to use: ICC(1,·) – each subject rated by different raters; ICC(2,·) – the raters are a random sample and absolute agreement matters; ICC(3,·) – fixed raters, consistency matters. ICC(·,1) holds for a single rater, ICC(·,k) for the mean of k raters. A rough guide (Koo and Li): below 0.5 poor, 0.5–0.75 moderate, 0.75–0.9 good, above 0.9 excellent reliability.
- ICC(3,k) equals Cronbach's alpha of the same data (the raters as items).
- An ICC's confidence interval tends to be wide with few subjects – even a high ICC may have a low lower bound. The ICC(2,·) intervals are approximate (Satterthwaite's degrees of freedom).
- Careful: there are few data, the estimates (of the variance and quartiles especially) are uncertain.

Agreement of two doctors in three diagnostic categories.

Input: `data(A B; 1 1; 1 1; 2 2; 1 2; 2 2; 3 3; 1 1; 2 3; 3 3; 2 2; 1 1; 3 2)` Cohen's kappa

Result

$\kappa \approx 0.6211$

$95\,\%:\ [0.256,\ 0.986]$

$p_o = 0.75$

$p_e \approx 0.3403$

Steps

- The pairs of ratings make the agreement table (rows rater A, columns rater B, agreement on the diagonal)

$A\backslash B$	1	2	3	$\sum$
1	**4**	1	0	5
2	0	**3**	1	4
3	0	1	**2**	3
$\sum$	4	5	3	12

- The observed agreement – the share of cases on the diagonal: $p_o = \frac{\sum n_{ii}}{N} = \frac{4+3+2}{12} = \frac{9}{12} = 0.75$
- The agreement expected by chance – from the row and column totals

$$p_e = \frac{\sum n_{i\cdot} n_{\cdot i}}{N^2} = \frac{5\cdot 4 + 4\cdot 5 + 3\cdot 3}{12^2} = \frac{49}{144} \approx 0.3403$$

- Cohen's kappa: $\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.75 - 0.3403}{1 - 0.3403} \approx 0.6211$ – the agreement is substantial (Landis and Koch).
- The 95 % confidence interval from the asymptotic standard error (Fleiss, Cohen and Everitt)

  $SE \approx 0.1861$

  $\kappa \pm z \cdot SE = 0.6211 \pm 1.96 \cdot 0.1861 \Rightarrow [0.256,\ 0.986]$

- κ corrects the agreement for the agreement expected by chance: 0 means agreement by chance alone, 1 perfect agreement. When one category prevails, κ comes out low even with a high percentage of agreement – that's why p_o is given too.
- The categories are taken as nominal – every disagreement weighs the same. Ordered categories (e.g. a degree of severity) call for a weighted κ, which penalizes a smaller disagreement less.

The same diagnoses with quadratic weights – neighbouring categories count as partial agreement.

Input: `data(A B; 1 1; 1 1; 2 2; 1 2; 2 2; 3 3; 1 1; 2 3; 3 3; 2 2; 1 1; 3 2)`

Cohen's kappa, Quadratic weights

Result

$$\kappa_w \approx 0.7955$$

$$95\,\%:\ [0.576,\ 1.000]$$

$$p_o = 0.9375$$

$$p_e \approx 0.6944$$

Steps

- The pairs of ratings make the agreement table (rows rater A, columns rater B, agreement on the diagonal)

$A\backslash B$	1	2	3	$\sum$
1	**4**	1	0	5
2	0	**3**	1	4
3	0	1	**2**	3
$\sum$	4	5	3	12

- Quadratic agreement weights – a small disagreement costs little, a large one a lot: $w_{ij} = 1 - \dfrac{(i-j)^2}{(k-1)^2}$

	1	2	3
1	1	0.75	0
2	0.75	1	0.75
3	0	0.75	1

- The weighted observed agreement – the table's counts times the weights: $p_o = \dfrac{\sum w_{ij} n_{ij}}{N} = \dfrac{11.25}{12} = 0.9375$
- The weighted agreement expected by chance – from the row and column totals with the weights: $p_e = \dfrac{\sum w_{ij} n_{i\cdot} n_{\cdot j}}{N^2} = \dfrac{100}{144} \approx 0.6944$
- The weighted Cohen's kappa: $\kappa_w = \dfrac{p_o - p_e}{1 - p_e} = \dfrac{0.9375 - 0.6944}{1 - 0.6944} \approx 0.7955$ – the agreement is substantial (Landis and Koch).
- The 95 % confidence interval from the asymptotic standard error (Fleiss, Cohen and Everitt)

  $SE \approx 0.1119$

  $\kappa_w \pm z \cdot SE = 0.7955 \pm 1.96 \cdot 0.1119 \Rightarrow [0.576,\ 1.000]$
- κ corrects the agreement for the agreement expected by chance: 0 means agreement by chance alone, 1 perfect agreement. When one category prevails, κ comes out low even with a high percentage of agreement – that's why p_o is given too.
- The weighted kappa counts partial agreement too: ratings one step apart aren't a full disagreement. With quadratic weights the weighted kappa comes close to the ICC; it is good to report which weights were used.

Cohen's kappa from a 2×2 agreement table.

Input: `data(20 5; 10 15)` Cohen's kappa

Result

$$\kappa = 0.4$$
$$95\,\%:\ [0.151,\ 0.649]$$
$$p_o = 0.7$$
$$p_e = 0.5$$

Steps

- The agreement table (rows rater A, columns rater B, agreement on the diagonal) with the row and column totals

$A\backslash B$	1	2	$\sum$
1	**20**	5	25
2	10	**15**	25
$\sum$	30	20	50

- The observed agreement – the share of cases on the diagonal: $p_o = \frac{\sum n_{ii}}{N} = \frac{20+15}{50} = \frac{35}{50} = 0.7$
- The agreement expected by chance – from the row and column totals

$$p_e = \frac{\sum n_{i\cdot} n_{\cdot i}}{N^2} = \frac{25 \cdot 30 + 25 \cdot 20}{50^2} = \frac{1250}{2500} = 0.5$$

- Cohen's kappa: $\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4$ – the agreement is fair (Landis and Koch).
- The 95 % confidence interval from the asymptotic standard error (Fleiss, Cohen and Everitt)

  $SE \approx 0.127$

  $\kappa \pm z \cdot SE = 0.4 \pm 1.96 \cdot 0.127 \Rightarrow [0.151,\ 0.649]$

- κ corrects the agreement for the agreement expected by chance: 0 means agreement by chance alone, 1 perfect agreement. When one category prevails, κ comes out low even with a high percentage of agreement – that's why p_o is given too.

How well a test score separates patients (state 1) from healthy people (state 0).

Input: `data(12 0; 15 1; 9 0; 20 1; 18 1; 11 0; 7 0; 14 1; 16 0; 10 0)` ROC curve and AUC

Result

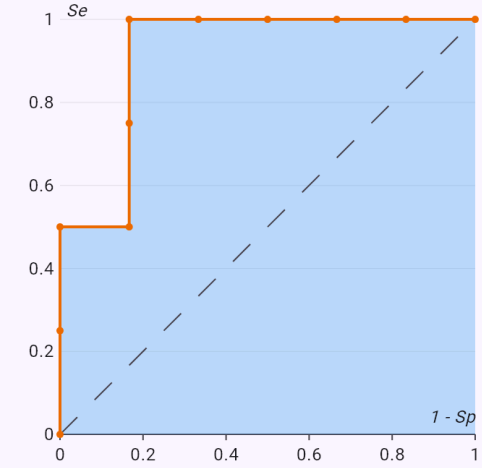
$$AUC \approx 0.9167$$
$$95\,\%:\ [0.706,\ 1.000]$$
$$z \approx 2.132,\ p \approx 0.03301 \le \alpha = 0.05$$
$$c = 14:\ Se = 1,\ Sp \approx 0.8333$$

Steps

- Positive cases (state 1) and negative ones (state 0): $n_1 = 4, \quad n_0 = 6$
- All the scores are sorted and ranked. The positive cases' rank sum: $R_1 = 6 + 7 + 9 + 10 = 32$
- The Mann–Whitney U – the number of (positive, negative) pairs where the positive case scores higher: $U = R_1 - \frac{n_1(n_1+1)}{2} = 32 - \frac{4 \cdot 5}{2} = 22$
- The area under the ROC curve: $AUC = \frac{U}{n_1 n_0} = \frac{22}{4 \cdot 6} \approx 0.9167$ – the test's discrimination is outstanding.
- The 95 % confidence interval by Hanley and McNeil
  $$Q_1 = \frac{AUC}{2 - AUC} = 0.8462$$
  $$Q_2 = \frac{2\,AUC^2}{1 + AUC} = 0.8768$$
  $$SE = \sqrt{\frac{AUC(1-AUC) + (n_1-1)(Q_1 - AUC^2)}{n_1 n_0}}$$
  $$AUC \pm z \cdot SE = 0.9167 \pm 1.96 \cdot 0.1074 \Rightarrow [0.706,\ 1.0$$
- The null hypothesis: the test doesn't discriminate (AUC = 0.5), the alternative: it does – the Mann–Whitney test: $H_0: AUC = 0.5, \quad H_1: AUC \ne 0.5$
- The test statistic – under H_0 it has the standard normal distribution: $z = \frac{U - \frac{n_1 n_0}{2}}{\sigma_U} = \frac{22 - 12}{4.69} \approx 2.132$
- The critical value at the level α: $z_{0.975} \approx 1.96$, the critical region $|z| \ge 1.96$
- The p-value: $p = 2\,P(Z \ge 2.132) \approx 0.03301$
- Since $p \approx 0.03301 \le \alpha = 0.05$, we reject the null hypothesis – the result is statistically significant.
- The p-value is the probability of a result at least as extreme as the observed one if H_0 holds – not the probability that H_0 holds. Not rejecting H_0 doesn't mean H_0 is true; statistical significance says nothing about the size or practical importance of a difference.
- The cut-off with the largest Youden's index is 14 (a score ≥ 14 is positive)
  $Se = 1$
  $Sp \approx 0.8333$
  $J = Se + Sp - 1 \approx 0.8333$
- The AUC is the probability that a randomly chosen positive case scores higher than a randomly chosen negative one (0.5 – the test doesn't discriminate, 1 – it discriminates perfectly). The cut-off by Youden's index is just one option – in practice it is also chosen by the cost of false positive and false negative results.
- Careful: there are few data, the estimates (of the variance and quartiles especially) are uncertain.

Graph

ROC curve – the shaded area is the AUC, the dashed diagonal is a test that doesn't discriminate



Scores of patients and healthy people in two groups of different sizes, with the Mann–Whitney test.

Input: `data(pacienti = 15 20 18 14 17; zdravi = 12 9 11 7 16 10 13)` ROC curve and AUC

Result

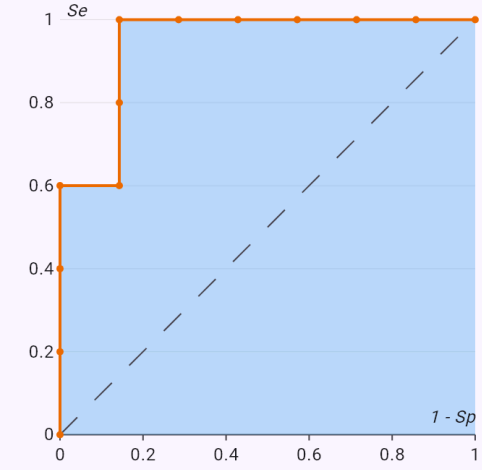
$$AUC \approx 0.9429$$
$$95\,\%:\ [0.787,\ 1.000]$$
$$z \approx 2.517,\ p \approx 0.01183 \le \alpha = 0.05$$
$$c = 14:\ Se = 1,\ Sp \approx 0.8571$$

Steps

- The positive group $\mathrm{pacienti}$ and the negative group zdravi: $n_1 = 5,\quad n_0 = 7$
- All the scores are sorted and ranked. The positive cases' rank sum: $R_1 = 7 + 8 + 10 + 11 + 12 = 48$
- The Mann–Whitney U – the number of (positive, negative) pairs where the positive case scores higher: $U = R_1 - \frac{n_1(n_1+1)}{2} = 48 - \frac{5 \cdot 6}{2} = 33$
- The area under the ROC curve: $AUC = \frac{U}{n_1 n_0} = \frac{33}{5 \cdot 7} \approx 0.9429$ – the test's discrimination is outstanding.
- The 95 % confidence interval by Hanley and McNeil
  $$Q_1 = \frac{AUC}{2 - AUC} = 0.8919$$
  $$Q_2 = \frac{2\,AUC^2}{1 + AUC} = 0.9151$$
  $$SE = \sqrt{\frac{AUC(1-AUC) + (n_1 - 1)(Q_1 - AUC^2)}{n_1 n_0}}$$
  $$AUC \pm z \cdot SE = 0.9429 \pm 1.96 \cdot 0.07971 \Rightarrow [0.787,\ 1.$$
- The null hypothesis: the test doesn't discriminate (AUC = 0.5), the alternative: it does – the Mann–Whitney test: $H_0: AUC = 0.5,\quad H_1: AUC \ne 0.5$
- The test statistic – under H_0 it has the standard normal distribution: $z = \frac{U - \frac{n_1 n_0}{2}}{\sigma_U} = \frac{33 - 17.5}{6.158} \approx 2.517$
- The critical value at the level α: $z_{0.975} \approx 1.96$, the critical region $|z| \ge 1.96$
- The p-value: $p = 2\,P(Z \ge 2.517) \approx 0.01183$
- Since $p \approx 0.01183 \le \alpha = 0.05$, we reject the null hypothesis – the result is statistically significant.
- The p-value is the probability of a result at least as extreme as the observed one if H_0 holds – not the probability that H_0 holds. Not rejecting H_0 doesn't mean H_0 is true; statistical significance says nothing about the size or practical importance of a difference.
- The cut-off with the largest Youden's index is 14 (a score ≥ 14 is positive)
  $$Se = 1$$
  $$Sp \approx 0.8571$$
  $$J = Se + Sp - 1 \approx 0.8571$$
- The AUC is the probability that a randomly chosen positive case scores higher than a randomly chosen negative one (0.5 – the test doesn't discriminate, 1 – it discriminates perfectly). The cut-off by Youden's index is just one option – in practice it is also chosen by the cost of false positive and false negative results.
- Careful: there are few data, the estimates (of the variance and quartiles especially) are uncertain.

Graph

ROC curve – the shaded area is the AUC, the dashed diagonal is a test that doesn't discriminate



Two tests on the same people – do they discriminate equally well? (DeLong's test)

Input: `data(A B stav; 12 10 0; 15 14 1; 9 11 0; 20 13 1; 18 19 1; 11 9 0; 7 12 0; 14 16 1;`

16 8 0; 10 13 0)

ROC curve and AUC

Result

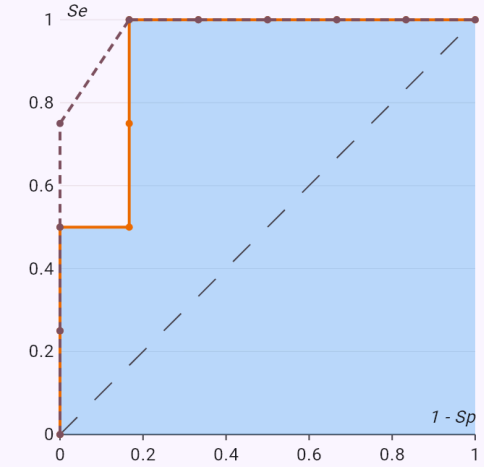
$$AUC_{\mathrm{A}} \approx 0.9167,\ AUC_{\mathrm{B}} \approx 0.9792$$
$$z \approx -0.571,\ p \approx 0.568 > \alpha = 0.05$$
$$95\,\%:\ [-0.277,\ 0.152]$$

Steps

- Two tests A and B on the same cases, the positive state 1: $n_1 = 4, \quad n_0 = 6$
- The areas under both tests' ROC curves and their standard errors (DeLong)
  $AUC_{\mathrm{A}} \approx 0.9167$
  $AUC_{\mathrm{B}} \approx 0.9792$
  $SE_{\mathrm{A}} \approx 0.09623$
  $SE_{\mathrm{B}} \approx 0.02946$
- The variance of the difference by DeLong – the tests are on the same cases, so their covariance is subtracted (the AUCs' correlation -0.327)
  $\operatorname{var}(AUC_{\mathrm{A}} - AUC_{\mathrm{B}}) = \operatorname{var} AUC_{\mathrm{A}} + \operatorname{var} AUC_{\mathrm{B}} - 2\,c$
  $= 0.009259 + 8.681 \cdot 10^{-4} - 2 \cdot$
  $SE = \sqrt{0.01198} \approx 0.1094$
- The null hypothesis: both tests discriminate equally, the alternative: they differ
  H_0: $AUC_{\mathrm{A}} = AUC_{\mathrm{B}}$
  H_1: $AUC_{\mathrm{A}} \neq AUC_{\mathrm{B}}$
- The test statistic – under H_0 it has the standard normal distribution:
  $$z = \frac{AUC_{\mathrm{A}} - AUC_{\mathrm{B}}}{SE} = \frac{-0.0625}{0.1094} \approx -0.571$$
- The critical value at the level α: $z_{0.975} \approx 1.96$, the critical region $|z| \geq 1.96$
- The p-value: $p = 2\,P(Z \geq 0.571) \approx 0.568$
- Since $p \approx 0.568 > \alpha = 0.05$, we don't reject the null hypothesis – the data don't give enough evidence against it.
- The p-value is the probability of a result at least as extreme as the observed one if H_0 holds – not the probability that H_0 holds. Not rejecting H_0 doesn't mean H_0 is true; statistical significance says nothing about the size or practical importance of a difference.
- The 95 % confidence interval of the two AUCs' difference $AUC_{\mathrm{A}} - AUC_{\mathrm{B}} \pm z \cdot SE = -0.0625 \pm 1.96 \cdot 0.1094 =$
- DeLong's test compares two tests' AUCs on the same cases – it takes into account that they are correlated. A non-significant difference doesn't mean the tests are equally good, only that the data didn't show a difference.
- Careful: there are few data, the estimates (of the variance and quartiles especially) are uncertain.

Graph

Both tests' ROC curves – the first solid with the shaded area, the second dashed



How much to lengthen a 20-item test with reliability 0.7 to reach 0.9.

Input: `model(spearmanbrown; find=k[one]; r=0.7[one]; rt=0.9[one]; items=20[one])`

Result

$$k \approx 3.86$$
$$n' = 78$$

Steps

- Find k; given: $r_{xx} = 0.7, \quad r'_{xx} = 0.9, \quad n = 20$
- A longer test measures more reliably: lengthened k times by items of the same quality (parallel ones), its reliability grows by the Spearman–Brown formula r' = k·r / (1 + (k − 1)·r). Conversely, it tells how many times to lengthen a test to reach a target reliability. The formula also corrects a split-half reliability (k = 2). The model's equation: $k = \frac{r'_{xx}(1 - r_{xx})}{r_{xx}(1 - r'_{xx})}$
- Substituting: $k = \frac{0.9 \cdot (1 - 0.7)}{0.7 \cdot (1 - 0.9)} \approx 3.857$
- Items needed: $n' = k \cdot n = 3.857 \cdot 20 \approx 77.14 \Rightarrow n' = 78$
- The formula assumes the added items are as good as the original ones (parallel). The real gain is usually smaller, and a longer test also tires the respondent more.

The corrected level for 10 comparisons.

Input: `model(bonferroni; find=alpha[one]; alpha=0.05[one]; m=10[one])`

Result

$$\alpha' = 0.005$$

Steps

- Find α'; given: $\alpha = 0.05, \quad m = 10$
- With m tests at the level α, the probability that at least one comes out significant by chance alone grows: 1 − (1 − α)^m. Bonferroni's correction therefore judges each test at the stricter level α' = α/m. Šidák's correction 1 − (1 − α)^(1/m) is slightly less strict. The model's equation: $\alpha' = \frac{\alpha}{m}$
- Substituting: $\alpha' = \frac{0.05}{10} = 0.005$
- Probability of at least one false significant result without a correction: $1 - (1 - \alpha)^m = 1 - (1 - 0.05)^{10} \approx 0.4013$
- Šidák's correction $\alpha'_S = 1 - (1 - \alpha)^{1/m} = 1 - (1 - 0.05)^{1/10} \approx 0.005116$
- Bonferroni's correction is conservative: it guards against false significant results, but with many comparisons it easily misses a real difference. Less strict are e.g. Holm's method or controlling the FDR (Benjamini–Hochberg).

Known limitations

- The percentile comes from the normal distribution – for skewed tests (TMT's times) only the norm table's percentile is right.
- The confidence intervals are asymptotic or approximate – with small samples (up to 10–20 cases) take them as a guide only.
- A task in words or from a photo (a textbook's problem) isn't supported yet – the values go into the form.

Generated from the Mathtrace app