

## De netto koopsom van een levensverzekering bepalen.

*(Handig om deze opdracht in Spyder te maken, maar hoeft niet per se)*

Er wordt een prenummerando lijfrente gekocht op het leven van twee deelnemers, x en y. Zolang beiden leven wordt er een bedrag A per jaar uitgekeerd. De langstlevende ontvangt vanaf het overlijden van de eerst stervende een bedrag van B euro per jaar.

De afsluitdatum is per 1 januari van het rekenjaar.

- a. Bereken de netto koopsom met behulp van de prognosetafel en de yieldcurve (zie Brightspace: 'Prognosetafel en rts.xlsx').  
Hierbij moet de gebruiker om input worden gevraagd, namelijk:
  - Leeftijd deelnemer x
  - Geslacht deelnemer x
  - Leeftijd deelnemer y
  - Geslacht deelnemer y
  - Rekenjaar
  - Uitkeringshoogte A
  - Uitkeringshoogte B
- b. Teken vier grafieken (in één assenstelsel): voor elke deelnemer de sterftekans en overlevingskans vanaf huidige leeftijd deelnemer.  
Geef ook as-namen en een legenda erbij. De grafieken van deelnemer x moeten twee verschillende blauwtinten hebben en die van deelnemer y twee verschillende roze tinten.

### Stel je input is:

Deelnemer x: 60 jaar, Man.

Deelnemer y: 64 jaar, Man.

Rekenjaar: 2015

Bedrag A (beide leven): 800 euro.

Bedrag B (één leeft, één dood): 600 euro.

Dan is de koopsom 12.665,15 euro.

### Tips:

1. Bepaal eerst wiskundig hoe je deze koopsom moet uitrekenen. Zie financiële wiskunde 2.1. Deze opdracht komt sterk overeen met opdracht 4.4 uit het dictaat van dat vak. Helemaal mooi als je voor die opdracht een Excel hebt gemaakt. Veel van de formules die je in die Excelfile hebt verwerkt kun je nu mooi gebruiken. Je kunt ook de (tussentijdse) berekeningen van je Python programma controleren met deze Excel.
2. Stap voor stap: splits stukken op.
  - a. Diverse modules inlezen (pas tijdens het programmeren van je code zul je bedenken welke modules je moet inlezen)
  - b. Inlezen van de inputgegevens, zoals leeftijd, geslacht, rekenjaar, etc.

- c. Maak die input gegevens tijdelijk constant. Anders moet je tijdens het testen van je code steeds weer die variabelen invoeren als gebruiker.  
Schrijf bijv. eerst op `lftX = int(input('Wat tekst hier'))`, maak er dan een comment van door er `#` voor te zetten en zet dan op een nieuwe regel `lftX = 64` (de waarden gelijk aan het rekenvoorbeeld, want dan kun je makkelijker je programma de uitkomst 12.665,15 euro laten uitrekenen. Als dat is gelukt maak je daarna je variabelen weer echt variabel met de input functies).
  - d. Inlezen van de drie Excel-sheets in (drie) dataframes, dat omzetten naar geneste lists. Daarna handig om die lists uit te printen en te kijken welke gegevens erin staan en of die overeenkomen met de Excelfile en juist zijn.
  - e. M.b.v. de ingelezen gegevens geneste lijsten maken met erin de basiskansen, zoals Qx, Px en nPx en dat voor beide deelnemers.  
Daarna de lijsten uitprinten en controleren of de getallen kloppen.
  - f. Geneste lijsten aanmaken met benodigde kansen, zoals nPxy, etc.  
Daarna de lijsten uitprinten en controleren of de getallen kloppen.
  - g. De juiste kansen met elkaar vermenigvuldigen en/of optellen en vermenigvuldigen met de uitkeringen en die waarden constant maken met de rente (yieldcurve).
  - h. Grafieken maken.
    - Maak lijsten met de x-as waarden (leeftijden) en lijsten met de y-as waarden (kansen) en plot die met `.plot()`
    - Zoek op internet hoe je legenda toevoegt en kleuren verandert.
3. Het is handig om functies te schrijven, die maken je code overzichtelijker en waarschijnlijk korter.
  4. Vergeet comments niet.
  5. Verwijder hulpprints (prints die je gebruikte om gegevens te controleren) als je klaar bent.
  6. Misschien handig/nodig (zie bijv. w3schools):
    - a. De `index()` functie.
    - b. Voor dataframes de `.columns`