Source code S2: Source code of the question based HALF index calculator

```
from tkinter import *
from tkinter import ttk

class QB_HALF:

    def __init__(self):

        self.root = Tk()
        self.root.title("QB HALF")
        self.root.geometry("700x500")

        beef = DoubleVar()
        pork = DoubleVar()
        poultry = DoubleVar()
        lamb_mutton = DoubleVar()
        fish = DoubleVar()
        cheese = DoubleVar()
        milk = DoubleVar()
        eggs = DoubleVar()
        quantities = DoubleVar()
        staples = DoubleVar()


        #total land surface area
        global atls #in ha
        atls = 13009000000.


        global beef_kcal
        beef_kcal = 1.6125
        global pork_kcal
        pork_kcal = 3.79
        global poultry_kcal
        poultry_kcal = 1.83
        global lamb_mutton_kcal
        lamb_mutton_kcal = 1.58
        global fish_kcal
        fish_kcal = 2.42
        global cheese_kcal
        cheese_kcal = 3.7
        global milk_kcal
        milk_kcal = 0.452
```

```
global eggs_kcal
eggs_kcal = 1.32
global extra_staple_kcal
extra_staple_kcal = 2.72

#feed conversion ratios
global beef_feed_ratio
beef_feed_ratio = 25.
global pork_feed_ratio
pork_feed_ratio = 6.4
global poultry_feed_ratio
poultry_feed_ratio = 3.3
global lamb_mutton_feed_ratio
lamb_mutton_feed_ratio = 15.
global eggs_feed_ratio
eggs_feed_ratio = 2.3
global milk_feed_ratio
milk_feed_ratio = 0.7
global cheese_feed_ratio
cheese_feed_ratio = 4.9

global beef_pasture_required #in m2/g
beef_pasture_required = 0.197
global lamb_mutton_pasture_required
lamb_mutton_pasture_required = 0.024
global milk_pasture_required
milk_pasture_required = .001
global cheese_pasture_required
cheese_pasture_required = .007


#population
global population_1850
population_1850 = 1262000000.
global population_1975
population_1975 = 4079000000.
global population_2018
population_2018 = 7631000000.


#kg/ha/yr
global plant_yield_1850
plant_yield_1850 = 780
global plant_yield_1975
```

```
        plant_yield_1975 = 1590
        global plant_yield_2018
        plant_yield_2018 = 3530

        #land use requirements for fish production
        global fish_lu
        fish_lu = .00000004 #ha/g

        FREQUENCIES = [
            ("Never", "0"),
            ("Rarely: 1-2 times a week", "1.5"),
            ("Sometimes: 3-4 times a week", "3.5"),
            ("Often: 5-7 times a week", "6.0"),
            ("More than once a day (for example: every day for dinner
and sometimes for a secondary meal)", "10")
        ]

        QUANTITIES = [
            ("Less than others", "0.7"),
            ("Equal to others", "1.0"),
            ("More than others", "1.3")
        ]

        STAPLES = [
            ("Grains (e.g. rice, maize, pasta, bread)", "1.0"),
            ("Root vegetables (e.g. potatoes, cassava)", "2.0")
        ]

        EGGS_AMOUNT = [
            ("None", "0"),
            ("A few: 1-2", "2."),
            ("Several: 3-5", "4."),
            ("Many: more than 5", "7.")

        ]

        MILK_AMOUNTS = [
            ("Never", "0"),
            ("Less than once a day", "3."),
            ("Once a day", "7."),
            ("Twice a day", "14."),
            ("More than twice a day", "18.")
        ]
```

```
        self.notebook = ttk.Notebook(self.root)

        #the first tab = welcome screen
        self.tab1 = Frame(self.notebook)
        self.notebook.add(self.tab1, text="Welcome")

        welcome_message = Label(self.tab1, text = """Welcome to the
question based HALF index calculator!\n
        This program will allow you to calculate your personal HALF
index. A HALF index
        indicates the amount of land that would be required to feed the
global population
        if everyone adhered to your diet, expressed as a percentage of
the world's surface area.\n
        After you have filled in some information about your current
diet, you will
        receive what your HALF index would have been in 1850, 1975 and
in 2018. \n""")
        welcome_message.pack()

        self.button1 = Button(self.tab1, text = "Go to the first
question >>", command = self.select1)
        self.button1.pack()

        #the second tab = first question
        self.tab2 = Frame(self.notebook)
        self.notebook.add(self.tab2, text = "Q1")

        q1 = Label(self.tab2, text="Question 1: How much do you eat
compared to others?")
        q1.pack()

        for text, quantity in QUANTITIES:
            radiobuttons_quantities = Radiobutton(self.tab2, text=text,
variable=quantities, value=quantity)
            radiobuttons_quantities.pack(pady=10)

        self.confirmQuantities = Button(self.tab2, text = "Confirm",
command = lambda: self.clickedquantities(quantities.get()))
        self.confirmQuantities.pack()

        self.button2 = Button(self.tab2, text = "Go to the next
question >>", command = self.select2)
        self.button2.pack(pady=10)
```

```
        #the third tab = second question
        self.tab3 = Frame(self.notebook)
        self.notebook.add(self.tab3, text="Q2")


        q2 = Label(self.tab3, text="Question 2: How often do you eat a
100g portion of BEEF on average? \n Assume a slice of lunch meat is
25g. ")
        q2.pack()

        for text, frequency in FREQUENCIES:
            radiobuttons_beef = Radiobutton(self.tab3, text=text,
variable=beef, value=frequency)
            radiobuttons_beef.pack(pady=10)


        self.confirmBeef = Button(self.tab3, text = "Confirm", command
= lambda: self.clickedbeef(beef.get()))
        self.confirmBeef.pack()


        self.button3 = Button(self.tab3, text = "Go to the next
question >>", command = self.select3)
        self.button3.pack(pady=10)

        #the fourth tab = third question
        self.tab4 = Frame(self.notebook)
        self.notebook.add(self.tab4, text="Q3")

        q3 = Label(self.tab4, text="Question 3: How often do you eat a
100g portion of PORK on average? \n Assume a slice of lunch meat is
25g.")
        q3.pack()

        for text, frequency in FREQUENCIES:
            radiobuttons_pork = Radiobutton(self.tab4, text=text,
variable=pork, value=frequency)
            radiobuttons_pork.pack(pady=10)

        self.confirmPork = Button(self.tab4, text = "Confirm", command
= lambda: self.clickedpork(pork.get()))
        self.confirmPork.pack()
```

```
        self.button4 = Button(self.tab4, text = "Go to the next
question >>", command = self.select4)
        self.button4.pack(pady=10)


        #the fifth tab = fourth question
        self.tab5 = Frame(self.notebook)
        self.notebook.add(self.tab5, text = "Q4")

        q4 = Label(self.tab5, text="Question 4: How often do you eat a
100g portion of CHICKEN or TURKEY on average? \n Assume a slice of
lunch meat is 25g.")
        q4.pack()

        for text, frequency in FREQUENCIES:
            radiobuttons_poultry = Radiobutton(self.tab5, text=text,
variable=poultry, value=frequency)
            radiobuttons_poultry.pack(pady=10)

        self.confirmPoultry = Button(self.tab5, text = "Confirm",
command = lambda: self.clickedpoultry(poultry.get()))
        self.confirmPoultry.pack()

        self.button5 = Button(self.tab5, text = "Go to the next
question >>", command = self.select5)
        self.button5.pack(pady=10)


        #the sixth tab = fifth question
        self.tab6 = Frame(self.notebook)
        self.notebook.add(self.tab6, text = "Q5")

        q5 = Label(self.tab6, text="Question 5: How often do you eat a
100g portion of LAMB or MUTTON on average?")
        q5.pack()

        for text, frequency in FREQUENCIES:
            radiobuttons_lamb_mutton = Radiobutton(self.tab6,
text=text, variable=lamb_mutton, value=frequency)
            radiobuttons_lamb_mutton.pack(pady=10)

        self.confirmLambMutton = Button(self.tab6, text = "Confirm",
command = lambda: self.clickedlambmutton(lamb_mutton.get()))
        self.confirmLambMutton.pack()
```

```
        self.button6 = Button(self.tab6, text = "Go to the next
question >>", command = self.select6)
        self.button6.pack(pady=10)


        #the seventh tab = sixth question
        self.tab7 = Frame(self.notebook)
        self.notebook.add(self.tab7, text = "Q6")

        q6 = Label(self.tab7, text="Question 6: How often do you eat a
100g portion of FISH on average?")
        q6.pack()

        for text, frequency in FREQUENCIES:
            radiobuttons_fish = Radiobutton(self.tab7, text=text,
variable=fish, value=frequency)
            radiobuttons_fish.pack(pady=10)

        self.confirmFish = Button(self.tab7, text = "Confirm", command
= lambda: self.clickedfish(fish.get()))
        self.confirmFish.pack()

        self.button7 = Button(self.tab7, text = "Go to the next
question >>", command = self.select7)
        self.button7.pack(pady=10)


        #the eighth tab = seventh question
        self.tab8 = Frame(self.notebook)
        self.notebook.add(self.tab8, text = "Q7")

        q7 = Label(self.tab8, text="Question 7: How often do you eat a
20g portion of CHEESE on average? \n Assume a pre-cut slice is 20g.")
        q7.pack()

        for text, frequency in FREQUENCIES:
            radiobuttons_cheese = Radiobutton(self.tab8, text=text,
variable=cheese, value=frequency)
            radiobuttons_cheese.pack(pady=10)

        self.confirmCheese = Button(self.tab8, text = "Confirm",
command = lambda: self.clickedcheese(cheese.get()))
        self.confirmCheese.pack()
```

```
        self.button8 = Button(self.tab8, text = "Go to the next
question >>", command = self.select8)
        self.button8.pack(pady=10)

        #the ninth tab = eighth question
        self.tab9 = Frame(self.notebook)
        self.notebook.add(self.tab9, text = "Q8")

        q8 = Label(self.tab9, text="Question 8: How often do you
consume a 250g portion of cow MILK \n or other LIQUID DAIRY products
(e.g. yoghurt) on average?")
        q8.pack()

        for text, milk_amount in MILK_AMOUNTS:
            radiobuttons_milk = Radiobutton(self.tab9, text=text,
variable=milk, value=milk_amount)
            radiobuttons_milk.pack(pady=10)

        self.confirmMilk = Button(self.tab9, text = "Confirm", command
= lambda: self.clickedmilk(milk.get()))
        self.confirmMilk.pack()

        self.button9 = Button(self.tab9, text = "Go to the next
question >>", command = self.select9)
        self.button9.pack(pady=10)

        #the tenth tab = ninth question
        self.tab10 = Frame(self.notebook)
        self.notebook.add(self.tab10, text = "Q9")

        q9 = Label(self.tab10, text="Question 9: How many EGGS do you
eat per week on average?")
        q9.pack()

        for text, egg_amount in EGGS_AMOUNT:
            radiobuttons_eggs = Radiobutton(self.tab10, text=text,
variable=eggs, value=egg_amount)
            radiobuttons_eggs.pack(pady=10)

        self.confirmEggs = Button(self.tab10, text = "Confirm", command
= lambda: self.clickedeggs(eggs.get()))
        self.confirmEggs.pack()
```

```
        self.button10 = Button(self.tab10, text = "Go to the next
question >>", command = self.select10)
        self.button10.pack(pady=10)


        #the eleventh tab = tenth question
        self.tab11 = Frame(self.notebook)
        self.notebook.add(self.tab11, text = "Q10")

        q10 = Label(self.tab11, text="Question 10: Which staple food do
you eat most often?")
        q10.pack()

        for text, staple in STAPLES:
            radiobuttons_staples = Radiobutton(self.tab11, text=text,
variable=staples, value=staple)
            radiobuttons_staples.pack(pady=10)

        self.confirmStaple = Button(self.tab11, text = "Confirm",
command = lambda: self.clickedstaple(staples.get()))
        self.confirmStaple.pack()

        self.button11 = Button(self.tab11, text = "Calculate the HALF
index >>", command = self.clickedhalf)
        self.button11.pack(pady=10)

        #the twelfth tab = final HALF index
        self.tab12 = Frame(self.notebook)
        self.notebook.add(self.tab12, text = "HALF")

        self.notebook.pack(fill="both", expand=1)

        self.root.mainloop()

    def select1(self):
        self.notebook.select(1)

    def select2(self):
        self.notebook.select(2)

    def select3(self):
        self.notebook.select(3)

    def select4(self):
```

```
        self.notebook.select(4)

    def select5(self):
        self.notebook.select(5)

    def select6(self):
        self.notebook.select(6)

    def select7(self):
        self.notebook.select(7)

    def select8(self):
        self.notebook.select(8)

    def select9(self):
        self.notebook.select(9)

    def select10(self):
        self.notebook.select(10)

    def clickedbeef(self, value):
        global beef_grams
        beef_grams = float(value)*100
        global beef_feed
        beef_feed = beef_grams * beef_feed_ratio
        global beef_pasture
        beef_pasture = (beef_grams * beef_pasture_required)*52 #in m2
        self.confirmBeef['state']=DISABLED

    def clickedpork(self, value):
        global pork_grams
        pork_grams = float(value)*100
        global pork_feed
        pork_feed = pork_grams * pork_feed_ratio
        self.confirmPork['state']=DISABLED

    def clickedpoultry(self,value):
        global poultry_grams
        poultry_grams = float(value)*100
        global poultry_feed
        poultry_feed = poultry_grams * poultry_feed_ratio
        self.confirmPoultry['state']=DISABLED

    def clickedlambmutton(self,value):
```

```
        global lamb_mutton_grams
        lamb_mutton_grams = float(value)*100
        global lamb_mutton_feed
        lamb_mutton_feed = lamb_mutton_grams * lamb_mutton_feed_ratio
        global lamb_mutton_pasture
        lamb_mutton_pasture = (lamb_mutton_grams *
lamb_mutton_pasture_required)*52
        self.confirmLambMutton['state']=DISABLED

    def clickedfish(self,value):
        global fish_grams_2018
        fish_grams_2018= float(value)*100*.5
        global fish_land_use
        fish_land_use = (fish_grams_2018 * fish_lu)*52 #land use in ha
per year per person

        self.confirmFish['state']=DISABLED

    def clickedcheese(self,value):
        global cheese_grams
        cheese_grams = float(value)*100
        global cheese_feed
        cheese_feed = cheese_grams * cheese_feed_ratio
        global cheese_pasture
        cheese_pasture = (cheese_grams * cheese_pasture_required)*52
        self.confirmCheese['state']=DISABLED

    def clickedmilk(self,value):
        global milk_grams
        milk_grams = float(value)*250
        global milk_feed
        milk_feed = milk_grams * milk_feed_ratio
        global milk_pasture
        milk_pasture = (milk_grams * milk_pasture_required)*52
        self.confirmMilk['state']=DISABLED

    def clickedeggs(self,value):
        global eggs_grams
        eggs_grams = float(value)*250
        global eggs_feed
        eggs_feed = eggs_grams * eggs_feed_ratio
        self.confirmEggs['state']=DISABLED

    def clickedquantities(self,value):
```

```
        global q
        q = float(value)
        self.confirmQuantities['state']=DISABLED

    def clickedstaple(self,value):
        global staple_yield_1850
        global staple_yield_1975
        global staple_yield_2018
        global extra_staple_kcal
        if float(value) == 1.0:#grains
            staple_yield_1850 = 780 #in kg/ha/yr
            staple_yield_1975 = 1700
            staple_yield_2018 = 3700
            extra_staple_kcal = 2.72 #in kcal/g
        else: #roots
            staple_yield_1850 = 10500
            staple_yield_1975 = 11300
            staple_yield_2018 = 13400
            extra_staple_kcal = 1.12
        self.confirmStaple['state']=DISABLED

    def clickedhalf(self):
        self.notebook.select(11)

        #calculations for HALF index
        animal_products_kcal = (beef_grams*beef_kcal +
pork_grams*pork_kcal + poultry_grams*poultry_kcal
+lamb_mutton_grams*lamb_mutton_kcal +fish_grams_2018*fish_kcal
+cheese_grams*cheese_kcal + milk_grams*milk_kcal +
eggs_grams*eggs_kcal)/7 #total calories from animal products per person
per day
        if
beef_grams+pork_grams+poultry_grams+lamb_mutton_grams+fish_grams_2018
>0:
            ref_ppb= 2486 #calories from plant-based products from the
reference diet
            ref_ppb_grams = 1831.51 #mass from plant-based products
from the reference diet
        else:
            ref_ppb = 2860
            ref_ppb_grams = 1570.96
        extra_staple_kg_year = ((3533-ref_ppb-
animal_products_kcal)/extra_staple_kcal)*365/1000 #staple foods needed
to scale the diet to 3533 kcal in kgs per person per year
```

```
        feed_kg_year =
(beef_feed+pork_feed+poultry_feed+lamb_mutton_feed+eggs_feed+cheese_fee
d+milk_feed)/1000*52
        ppb_kg_year = ref_ppb_grams*365/1000 + feed_kg_year #plant-
based products (without extra staples) per person in kg per year
        pasture = (beef_pasture + lamb_mutton_pasture + cheese_pasture
+ milk_pasture)/10000 #ha


        #calculate half index for the year 1850
        taap_1850 = ppb_kg_year/plant_yield_1850 +
extra_staple_kg_year/staple_yield_1850 #total land area required to
provide plant products (including extra staple) for one person per year
        aapp_1850 = q*(taap_1850 + pasture) #total agricultural area
needed for one person per year in ha, scaled for quantity; aquaculture
does not play a part yet
        taa_1850 = population_1850*(aapp_1850) #total agricultural area
needed for the world population in ha
        half_1850 = taa_1850/atls*100.

        #calculate half index for the year 1975
        taap_1975 = ppb_kg_year/plant_yield_1975 +
extra_staple_kg_year/staple_yield_1850 #total land area required to
provide plant products for one person per year
        aapp_1975 = q*(taap_1975 + pasture) #total agricultural area
needed for one person per year in ha, scaled for quantity; aquaculture
does not play a part yet
        taa_1975 = population_1975*(aapp_1975)
        half_1975 = taa_1975/atls*100.


        #calculate half index for the year 2018
        taap_2018 = ppb_kg_year/plant_yield_2018 +
extra_staple_kg_year/staple_yield_1850 #total land area required (in
ha) to provide plant products for one person per year
        aapp_2018 = q*(taap_2018 + pasture + fish_land_use) #total
agricultural area needed for one person per year in ha, scaled for
quantity
        taa_2018 = population_2018*(aapp_2018)
        half_2018 = taa_2018/atls*100.


        half_label_1850 = Label(self.tab12, text = "The HALF index in
1850 would have been %.2f%%." %half_1850)
```

```
        half_label_1850.pack()

        half_label_1975 = Label(self.tab12, text = "The HALF index in
1975 would have been %.2f%%." %half_1975)
        half_label_1975.pack()

        if half_2018 >34.0:
            half_label_2018 = Label(self.tab12, text = "The HALF index
in 2018 is %.2f%%. \n This is higher than the global average HALF value
of 34%%." %half_2018)
            half_label_2018.pack()

        else:
            half_label_2018 = Label(self.tab12, text = "The HALF index
in 2018 is %.2f%%. \n This is lower than the global average HALF value
of 34%%." %half_2018)
            half_label_2018.pack()



f=QB_HALF()
```