

ESTONIAN UNIVERSITY OF LIFE SCIENCES

Institute of Forestry and Engineering

**Tarkvararakendus „BirdfeedersForest“**

**Console Application “BirdfeedersForest”**

Software application

Curriculum in Software design, programming and Data Management

teacher Mait Lang , PhD

Tartu 2024

**TABLE OF CONTENTS:**

**Title page…………………………………………………………………….………..1**

**Table of contents…………………………………..….…………….……….………..2**

**Purpose of Application…….……………………………..………………………..... 3**

**Short guide of using the program …………………………………………….…....5**

**The Source code……………………………………………………..………….........6**

**Rules for input…………………………………………………………..….………..13**

**License information………………………………………………………….……....17**

**References……………………………………………….………………………........23**

**Application “BirdfeedersForest”**

**Purpose of the Application.**

The purpose of the console application “Birdfeeders” is to determine the impact of presence of humanmade birdfeeders in the forest subject of study on the ecosystem health which would be expressed in survival rate of birds in the forest and on the increased attractiveness of such a forest in the eyes of human visitors with children, who would be attracted by birds chirping and would enjoy bird feeding as a form of entertainment and as a way to experience outdoors.



Figure 1

As scientific studies by Millikin University in six forested sites in Illinois, US showed, “the individual health of birds improved with the presence of humanmade feeders with supplemental feeding, including increased antioxidant levels, reduced stress (heterophil-to-lymphocyte ratio) and more rapid feather growth. In some species, there was improved body condition index scores and innate immune defence” **(Wilcoxen E, et al 2015**). In general, with the presence of birdfeeders with grains, there were observed consistent patterns of birds being in greater overall health . “The simple act of putting out a bird feeder and keeping it clean and full can greatly improve the chances that birds and their offspring will survive. Survival rates for birds and their offspring were 38 percent higher in areas where bird feeders were present, and places with bird feeders have a higher percentage of young birds recruited into the breeding population than places without” (The Benefits of Bird Feeding - <https://www.perkypet.com/articles/the-benefits-of-bird-feeding> ).

I incorporated the data above in one of the functions designed to determine at the end the degree of attractiveness of a theoretical forest of an area of 5000 hectares in the visitors eyes.

Besides, bird feeders provide food during cold, barren months, give birds energy for migration, supplement natural food resources. As for humans, they offer relaxation and decrease stress. “In a world overwhelmed with technology, the joyful chirping of wild birds provides for visitors a nice escape to unwind from daily stressors.” (The Benefits of Bird Feeding”).

**Figure 2:**



**Short guide for using the program**

To run this program, you need a Python version at least 3.12.

To test my application, go to my Github (https://github.com/Makintosh24/SoftwareApplicationBirdfeeders), download the file.

Go to the repository, click the green button “CODE”, Download Zip.

Download repository as a zip file and unpack in in the folder of your choice.

I take this ZIP to my desktop. Extract it from zip, “Birdfeeders”.

We go to the source . Download Python 3.12 or higher on the computer.

Open the folder, run the command src\main.py

In the shell –

A screen shot of a computer

Description automatically generated

Open the shell- find the folder with **the source**.

I go to the source, I open the main.py and I run it in Python and test it.

I have a fully open license.

**The source.**

#

# BirdfeedersForest app

#

# to start go to folder birdfeeders and type

# python.exe .\src\main.py

#

# Copyright (c) by Shilyn Makin

#

import math

# Variable declarations

birdfeeders: int = 0 # Number of bird feeders in the forest of 5000 ha

# Calculated final survival rate based on diminishing returns formula

calculated\_survival\_rate: float = 0.0

# Forest attractiveness description

forestattractiveness: str = "The forest is considered more attractive due to the increased healthier bird population."

# Function to classify bird feeder segments and get survival rate range

def classify\_bird\_feeders(feeders: int) -> tuple:

Classify the number of feeders into segments and provide base survival rate range.

if feeders < 1:

return 'O', (0.0, 0.0) # No feeders

elif 1 <= feeders <= 3:

return 'A', (0.1, 0.3) # Segment A: Few feeders

elif 4 <= feeders <= 6:

return 'B', (0.3, 0.6) # Segment B: Moderate number of feeders

elif feeders >= 7:

return 'C', (0.6, 0.99) # Segment C: Many feeders

# Calculate the final survival rate with a diminishing returns effect based on the number of feeders.

def calculate\_survival\_rate\_with\_diminishing\_returns(feeders, initial\_rate):

k = 0.38 # Adjusted for 38% higher survival rates

# Logarithmic function to model diminishing returns

survival\_rate = initial\_rate + math.log10(1 + k \* feeders)

# Cap the survival rate between 0.1 and 0.99

return min(max(survival\_rate, 0.1), 0.99)

# Function to determine the forest attractiveness based on diminishing returns

def calculate\_attractiveness\_with\_diminishing\_returns(feeders, survival\_rate):

# Adjust attractiveness based on both the number of feeders and the survival rate

if feeders == 0 or survival\_rate < 0.2:

return "Not very attractive"

elif 1 <= feeders <= 3 and survival\_rate <= 0.4:

return "Moderately Attractive"

elif 4 <= feeders <= 6 and survival\_rate <= 0.7:

return "Quite Attractive"

elif 7 <= feeders <= 10 and survival\_rate <= 0.9:

return "Highly Attractive"

else: # Feeders > 10, where attractiveness plateaus

return "Saturated Attractiveness"

# Main function for user interaction

def main():

print("Welcome to the Bird Conservation Program!")

# Loop to allow multiple inputs

while True:

birdfeeders = int(input("Enter the number of bird feeders in the area (or -1 to exit): "))

# Exit condition

if birdfeeders == -1:

print("Exiting the program. Thank you!")

break

# Classify bird feeders and get the survival rate range

feeders\_segment, survival\_rate\_range = classify\_bird\_feeders(birdfeeders)

# Calculate survival rate with diminishing returns

final\_survival\_rate = calculate\_survival\_rate\_with\_diminishing\_returns(birdfeeders, survival\_rate\_range[0])

# Determine forest attractiveness

attractiveness = calculate\_attractiveness\_with\_diminishing\_returns(birdfeeders)

# Output the results

print("\nAnalysis Result:")

print(f"Number of Bird Feeders: {birdfeeders}")

print(f"Feeders Segment: {feeders\_segment}")

print(f"Initial Survival Rate (User Input): {initial\_survival\_rate:.2f}")

print(f"Calculated Bird Survival Rate with Diminishing Returns: {final\_survival\_rate:.2f}")

print(f"Forest Attractiveness: {attractiveness}")

print("-" \* 40)

# main function to control the app

def main():

print("\nWelcome to the Forest Attractiveness Calculator!")

# major loop to read in user input

while True:

# try to catch eventually ocurring input mistakes

try:

birdfeeders = int(input("Enter the number of bird feeders in the forest: "))

if birdfeeders < 0:

print("Number of feeders cannot be negative. Please try again.")

continue

if birdfeeders>20:

print ("Allowed number of feeders is 1-20")

continue

forestattractiveness = calculate\_attractiveness(birdfeeders)

survivalrate = float(input("Enter the local survival rate of birds: "))

if survivalrate <= 0:

print("Survival rate need to be positive. Please try again.")

continue

if survivalrate > 100:

print ("survivalrate cannot exceed 100%" )

continue

# Normalize survival rate input (percentage to decimal)

survivalrate /= 100

# User input for the initial survival rate manually

initial\_survival\_rate = float(input("Enter the initial survival rate of birds (between 0.0 and 1.0): "))

if initial\_survival\_rate < 0 or initial\_survival\_rate > 1:

print("Invalid input. Survival rate must be between 0.0 and 1.0.")

continue

# Calculate the final survival rate using diminishing returns

feeders\_segment, survival\_rate\_range = classify\_bird\_feeders(birdfeeders)

birdsurvivalrate = calculate\_survival\_rate\_with\_diminishing\_returns(birdfeeders, survival\_rate\_range[0])

# Print the results

print\_results(forestattractiveness, birdsurvivalrate)

# Check if user wants a new calculation

if input("New calculation (y or n)? ").lower() == "n":

print("Thank you for using the Forest Attractiveness Calculator. Goodbye!")

break

else:

print('\nStarting a new calculation...\n')

continue

except ValueError:

print("Invalid input. Please enter a valid number.")

# Execute the main function

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Rules for input**

The application at the final output determines the extent of attractiveness of the forest in the eyes of the human visitors as a function of the number of birdfeeders, implying that a greater amount of birdfeeders attract more birds, who have more healthy offspring, who attract more human visitors as a result by solving orderly the following two problems:

**1)**Function to classify bird feeder segments:

This function categorizes the number of bird feeders into segments such as "A" (few feeders), "B" (moderate feeders), and "C" (many feeders).

**Outputs**:

* The function returns a segment label (e.g., "A", "B", "C").
* It also returns the **range of survival rates** based on the number of feeders (for example, "A" corresponds to survival rates between 0.1 and 0.3).

For that, we input the number of the supposed birdfeeders in a theoretical forest with area of 5000 hectares. The maximum number of birdfeeders for such a forest is 20.

The function assigns feeder segments (A, B, C) directly to corresponding survival rate ranges (0.1-0.3, 0.3-0.6, 0.6-0.9). The reasoning is that as the number of feeders increases, the potential bird survival rate increases naturally.

This implementation uses a natural logic where:

* Fewer feeders lead to lower survival rates (Segment A).
* Moderate number of feeders correspond to medium survival rates (Segment B).
* Many feeders correspond to high survival rates (Segment C).

The survival rates are assigned based on the segments to reflect a realistic ecological relationship: more feeders result in higher potential survival rates thus the model reflects the idea that feeder quantity and survival are closely correlated.

2) The second function is **calculate\_survival\_rate\_with\_diminishing\_returns**: This function is designed to calculate the **bird survival rate** using a logarithmic formula to model the **diminishing returns effect** based on the number of feeders. It takes **feeders** and a user-provided **initial\_rate** as inputs and adjusts the survival rate according to a scaling factor k.

Calculate the survival rate of birds in a forest containing birdfeeders based on diminishing returns formula using a logarithmic function to model diminishing returns

survival\_rate = initial\_rate + math.log10(1 + k \* feeders)

implying that the survivalrate increases as the number of feeders, but at a diminishing rate. This approach reflects the reality that once a certain level of feeders is reached, adding more feeders has less impact on survival, as food is no longer the only limiting factor. This is a realistic model of how resources affect bird survival rates in nature.

If it has been observed that survival rates for birds and their offspring increase by **38%** in areas with bird feeders, this impacts the **scaling factor** (commonly represented as k) in the logarithmic function used to model diminishing returns.

The formula for the **survival rate with diminishing returns** is:

survival\_rate = initial\_rate + math.log10(1 + k \* feeders)

In this formula:

* k is a scaling factor that influences how much the number of feeders impacts the survival rate.
* The survival rate increment is modeled to grow at a diminishing rate as the number of feeders increases (logarithmic growth).

**Output**: It calculates the **final survival rate** using a logarithmic function to represent diminishing returns.

Inside the application, I declared the following three variables of different types:

BIRDFEEDERS: **int** = 0 – for number of birdfeeders in the forest

CALCULATED SURVIVAL RATE: **float** = 0.0 - the calculated final survival rate as a function of the number of birdfeeders based on diminishing returns formula

FOREST ATTRACTIVENESS DESCRIPTION : **str = "The forest is considered more attractive due to the increased healthier bird population.**" – a statement which describes the extent of attractiveness of the forest as a recreation site for visitors as a function of the number of the feeders. Attractiveness of the forest as a descriptive string. This specific message will automatically appear as part of the output regardless of the attractiveness levels determined by feeders.

As for the first function, To activate the Forest Attractiveness calculator, we insert the number of birdfeeders in the forest subject of study and determine the extent of its attractiveness in the eyes of the visitors, if a number under 0 is inserted, the system gives an error, the forest is not considered attractive, the threshold is 20. Then function assigns feeder segments (A, B, C) directly to corresponding survival rate ranges (0.1-0.3, 0.3-0.6, 0.6-0.9).

As for the second problem that the application is solving - This function models shows how the survival rate of birds improves with the number of bird feeders but with diminishing returns (i.e., the effect becomes less pronounced as more feeders are added).

Here, The user must insert only the input of initial\_rate manually.

The number of bird feeders (feeders) is **not directly input** by the user into this specific function. Instead, it is provided earlier when the user specifies the number of feeders in the general script workflow.

Input Structure looks like this:

**User inputs**:

  **feeders**: Entered first by the user and stored as a variable in the main script.

 **initial\_survival\_rate**: Manually entered by the user when prompted.

If upon those two procedures and subsequent solutions we desire to continue with our calculations, aca inserting new input, to the question that follows “ new calculation yes or no?” we should insert y for yes, and then the cycle continues, otherwise we answer n for “NO”, and the process halts and we get eventually to the end.

**License information :**

Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE

LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN

ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS

INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES

REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS

PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM

THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED

HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer

exclusive Copyright and Related Rights (defined below) upon the creator

and subsequent owner(s) (each and all, an "owner") of an original work of

authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for

the purpose of contributing to a commons of creative, cultural and

scientific works ("Commons") that the public can reliably and without fear

of later claims of infringement build upon, modify, incorporate in other

works, reuse and redistribute as freely as possible in any form whatsoever

and for any purposes, including without limitation commercial purposes.

These owners may contribute to the Commons to promote the ideal of a free

culture and the further production of creative, cultural and scientific

works, or to gain reputation or greater distribution for their Work in

part through the use and efforts of others.

For these and/or other purposes and motivations, and without any

expectation of additional consideration or compensation, the person

associating CC0 with a Work (the "Affirmer"), to the extent that he or she

is an owner of Copyright and Related Rights in the Work, voluntarily

elects to apply CC0 to the Work and publicly distribute the Work under its

terms, with knowledge of his or her Copyright and Related Rights in the

Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be

protected by copyright and related or neighboring rights ("Copyright and

Related Rights"). Copyright and Related Rights include, but are not

limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display,

communicate, and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or

likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work,

subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data

in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the

European Parliament and of the Council of 11 March 1996 on the legal

protection of databases, and under any national implementation

thereof, including any amended or successor version of such

directive); and

vii. other similar, equivalent or corresponding rights throughout the

world based on applicable law or treaty, and any national

implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention

of, applicable law, Affirmer hereby overtly, fully, permanently,

irrevocably and unconditionally waives, abandons, and surrenders all of

Affirmer's Copyright and Related Rights and associated claims and causes

of action, whether now known or unknown (including existing as well as

future claims and causes of action), in the Work (i) in all territories

worldwide, (ii) for the maximum duration provided by applicable law or

treaty (including future time extensions), (iii) in any current or future

medium and for any number of copies, and (iv) for any purpose whatsoever,

including without limitation commercial, advertising or promotional

purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each

member of the public at large and to the detriment of Affirmer's heirs and

successors, fully intending that such Waiver shall not be subject to

revocation, rescission, cancellation, termination, or any other legal or

equitable action to disrupt the quiet enjoyment of the Work by the public

as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason

be judged legally invalid or ineffective under applicable law, then the

Waiver shall be preserved to the maximum extent permitted taking into

account Affirmer's express Statement of Purpose. In addition, to the

extent the Waiver is so judged Affirmer hereby grants to each affected

person a royalty-free, non transferable, non sublicensable, non exclusive,

irrevocable and unconditional license to exercise Affirmer's Copyright and

Related Rights in the Work (i) in all territories worldwide, (ii) for the

maximum duration provided by applicable law or treaty (including future

time extensions), (iii) in any current or future medium and for any number

of copies, and (iv) for any purpose whatsoever, including without

limitation commercial, advertising or promotional purposes (the

"License"). The License shall be deemed effective as of the date CC0 was

applied by Affirmer to the Work. Should any part of the License for any

reason be judged legally invalid or ineffective under applicable law, such

partial invalidity or ineffectiveness shall not invalidate the remainder

of the License, and in such case Affirmer hereby affirms that he or she

will not (i) exercise any of his or her remaining Copyright and Related

Rights in the Work or (ii) assert any associated claims and causes of

action with respect to the Work, in either case contrary to Affirmer's

express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned,

surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or

warranties of any kind concerning the Work, express, implied,

statutory or otherwise, including without limitation warranties of

title, merchantability, fitness for a particular purpose, non

infringement, or the absence of latent or other defects, accuracy, or

the present or absence of errors, whether or not discoverable, all to

the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons

that may apply to the Work or any use thereof, including without

limitation any person's Copyright and Related Rights in the Work.

Further, Affirmer disclaims responsibility for obtaining any necessary

consents, permissions or other rights required for any use of the

Work.

d. Affirmer understands and acknowledges that Creative Commons is not a

party to this document and has no duty or obligation with respect to

this CC0 or use of the Work.

**References:**

**The Benefits of Bird Feeding** - https://www.perkypet.com/articles/the-benefits-of-bird-feeding

**Wilcoxen E. (Biology dep., Millikin University) Horn D. et al** (2015) Effects of Birdfeeding activities on the population of wild birds - *Conservation Psychology.* Vol 3. [www.ncbi.nlm.nih.gov/pmc/articles/PMC4778448/pdf/cov058.pdf](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4778448/pdf/cov058.pdf)

**Illustrations: Figure 1, Figure 2 -from “The Benefits of Birds Feeding*”***