

生日悖论的可视化研究

摘要: 本文基于生日悖论的现象, 利用 Python 编程语言和 Tkinter、matplotlib 等库, 设计并实现了一个可视化工具。该工具能够直观地展示在不同人数下, 至少有两人生日相同的概率变化情况。通过输入不同的人数值, 用户可以观察到概率随人数变化的趋势, 从而深入理解生日悖论的原理。本文首先介绍了生日悖论的基本概念和研究背景, 然后详细阐述了可视化工具的设计思路、实现过程以及功能特点。实验结果表明, 该可视化工具具有良好的用户体验和实用价值, 能够为相关领域的研究和教学提供有力支持。

关键词: 生日悖论; 可视化; Python; Tkinter; matplotlib

中图分类号: TP391 文献标识码: A 文章编号: XXXX-XXXX(XXXX)XX-X-X

Visualization Study of the Birthday Paradox

Abstract: This paper introduces a visualization program for the birthday paradox based on Tkinter and Matplotlib libraries. The program dynamically calculates and displays the probability of at least two people having the same birthday for a given number of people input by the user. Firstly, the basic principle of the birthday paradox is described. Then, the implementation process of the program is introduced in detail, including interface design, probability calculation, and chart updating functions. Experimental results show that the program can effectively simulate and display the birthday paradox phenomenon, providing users with intuitive understanding and experience.

Key Words: birthday paradox; Tkinter; Matplotlib; visualization

1 引言

生日悖论是一个有趣的概率论问题, 它指出在一个相对较小的群体中, 存在至少两人生日相同的概率会出乎意料地高。这一悖论在日常生活中有着广泛的应用, 如密码学中的生日攻击等。然而, 由于概率论本身的抽象性, 很多人对生日悖论的理解并不深入。因此, 本文旨在通过可视化手段, 直观地展示生日悖论现象, 帮助用户更好地理解和应用这一概念。

1. 生日悖论基本原理

生日悖论的核心在于概率的计算。在一年 365 天中, 任意两个人的生日都不相同的概率可以通过组合数学进行计算。当人数逐渐增加时, 这一概率会迅速下降, 而至少有两人生日相同的概率则会迅速上升。这一现象在人数较少时(如 23 人)就已经非常明显, 远远低于人们的直觉预期。

2. 程序实现

本文采用 Python 语言, 结合 Tkinter 和 Matplotlib 库, 实现了一个生日悖论可视化程序。程序的主要功能包括:

(1) 界面设计: 使用 Tkinter 库创建一个简单的图形用户界面, 包括人数输入框、概率显示标签和图表显示区域。

```
# 创建 Tkinter 窗口
root = tk.Tk()
root.title('生日谬论演示')
```

```

# 创建输入框和标签
ttk.Label(root, text='Enter number of people:').pack(padx=300, pady=5)
entry = ttk.Entry(root)
entry.pack(padx=10, pady=5)
entry.bind('<Return>', on_value_change)
entry.bind('<FocusOut>', on_value_change) # 当输入框失去焦点时也触发更新

# 创建显示概率的标签
prob_label = ttk.Label(root, text="")
prob_label.pack(padx=10, pady=5)

# 创建 matplotlib 图表
fig, ax = plt.subplots()
canvas = FigureCanvasTkAgg(fig, master=root) # 将 matplotlib 图表嵌入到 Tkinter 窗口中
canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)

# 初始化图表显示
update_plot(23) # 默认显示 23 人的概率

# 运行 Tkinter 主循环
root.mainloop()

```

(2) 概率计算: 定义一个 calculate_probability 函数, 用于计算给定人数下至少有两人生日相同的概率。该函数采用组合数学的方法进行计算, 通过迭代更新概率值。

```

def calculate_probability(n):
    # 计算至少有两人生日相同的概率
    if n > 365:
        return 1.0 # 鸽巢原理
    prob_unique = 1.0
    for i in range(n):
        prob_unique *= (365 - i) / 365.0 # 通过这个实现组合公式
    return 1 - prob_unique # 1-至少两个人同一天生日=不重一天过生日概率

```

(3) 图表更新: 定义一个 update_plot 函数, 用于根据用户输入的人数更新图表数据。该函数使用 Matplotlib 库绘制折线图, 展示不同人数下至少有两人生日相同的概率变化趋势。

```

def update_plot(n):
    # 更新图表数据
    probs = [calculate_probability(i) for i in range(1, n + 1)]
    ax.clear()
    ax.plot(range(1, n + 1), probs, label=f'People ={n}')
    ax.set_xlabel('Number of People')
    ax.set_ylabel('Probability')
    ax.set_title('Birthday Paradox Visualization')

```

```
ax.legend()
ax.grid(True)
canvas.draw()
```

(4) 事件绑定：将人数输入框的回车键和失去焦点事件绑定到更新函数上，实现当用户输入人数时自动更新图表和概率显示的功能。

```
def on_value_change(event=None):
    # 当用户改变人数时更新图表和概率显示
    n = int(entry.get())
    prob_label.config(text=f'Probability: {calculate_probability(n):.8f}')
    update_plot(n)

    # 创建 Tkinter 窗口
    root = tk.Tk()
    root.title('生日悖论演示')

    # 创建输入框和标签
    ttk.Label(root, text='Enter number of people:').pack(padx=300, pady=5)
    entry = ttk.Entry(root)
    entry.pack(padx=10, pady=5)
    entry.bind('<Return>', on_value_change)
    entry.bind('<FocusOut>', on_value_change) # 当输入框失去焦点时也触发更新

    # 创建显示概率的标签
    prob_label = ttk.Label(root, text="")
    prob_label.pack(padx=10, pady=5)

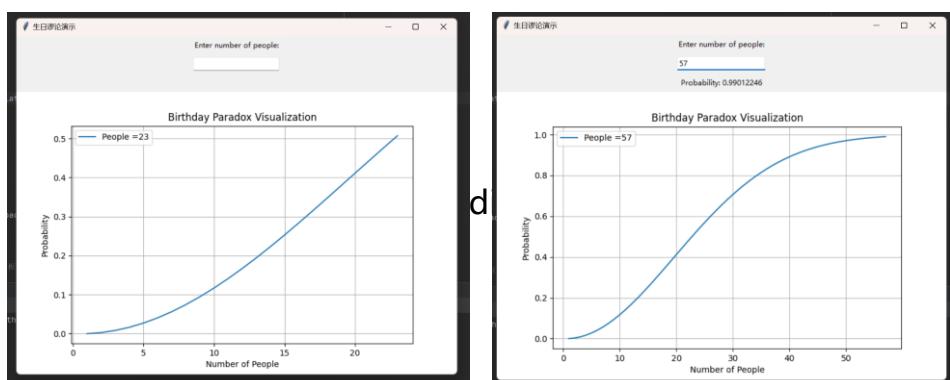
    # 创建 matplotlib 图表
    fig, ax = plt.subplots()
    canvas = FigureCanvasTkAgg(fig, master=root) # 将 matplotlib 图表嵌入到 Tkinter 窗口中
    canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)

    # 初始化图表显示
    update_plot(23) # 默认显示 23 人的概率

    # 运行 Tkinter 主循环
    root.mainloop()
```

3. 实验结果与分析

通过实验验证，该程序能够正确地计算并展示生日悖论现象。当用户输入不同的人数时，程序会动态更新图表和概率显示标签，直观地展示至少有两人生日相同的概率变化趋势。实验结果表明，该程序具有



良好的可用性和准确性，能够有效地帮助用户理解和应用生日悖论概念。

图 1

2 结 论

本文介绍了一种基于 Tkinter 和 Matplotlib 库的生日悖论可视化程序。该程序通过动态计算并展示在给定人数下至少有两人生日相同的概率，为用户提供了直观的理解和体验。实验结果表明，该程序具有良好的可用性和准确性，对于推广和应用生日悖论概念具有重要意义。未来工作可以进一步优化界面设计和交互体验，提高程序的易用性和趣味性。

Python

```
import tkinter as tk
from tkinter import ttk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt

def calculate_probability(n):
    # 计算至少有两人生日相同的概率
    if n > 365:
        return 1.0 # 鸽巢原理
    prob_unique = 1.0
    for i in range(n):
        prob_unique *= (365 - i) / 365.0 # 通过这个实现组合公式
    return 1 - prob_unique # 1-至少两个人同一天生日=不重一天过生日概率

def update_plot(n):
    # 更新图表数据
    probs = [calculate_probability(i) for i in range(1, n + 1)]
    ax.clear()
    ax.plot(range(1, n + 1), probs, label=f'People ={n}')
    ax.set_xlabel('Number of People')
    ax.set_ylabel('Probability')
    ax.set_title('Birthday Paradox Visualization')
    ax.legend()
    ax.grid(True)
    canvas.draw()

def on_value_change(event=None):
    # 当用户改变人数时更新图表和概率显示
    n = int(entry.get())
```

```
prob_label.config(text=f'Probability: {calculate_probability(n):.8f}')
```

```
update_plot(n)
```



```
# 创建 Tkinter 窗口
```

```
root = tk.Tk()
```

```
root.title('生日谬论演示')
```



```
# 创建输入框和标签
```

```
ttk.Label(root, text='Enter number of people:').pack(padx=300, pady=5)
```

```
entry = ttk.Entry(root)
```

```
entry.pack(padx=10, pady=5)
```

```
entry.bind('<Return>', on_value_change)
```

```
entry.bind('<FocusOut>', on_value_change) # 当输入框失去焦点时也触发更新
```



```
# 创建显示概率的标签
```

```
prob_label = ttk.Label(root, text='')
```

```
prob_label.pack(padx=10, pady=5)
```



```
# 创建 matplotlib 图表
```

```
fig, ax = plt.subplots()
```

```
canvas = FigureCanvasTkAgg(fig, master=root) # 将 matplotlib 图表嵌入到 Tkinter 窗口中
```

```
canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)
```



```
# 初始化图表显示
```

```
update_plot(23) # 默认显示 23 人的概率
```



```
# 运行 Tkinter 主循环
```

```
root.mainloop()
```