Minor in AI

20 March 2025

Revision - Mastering Input and Decision-Making in Python: From Basics to Logic

# Variables

# age (int)

# height (int, in cm)

# is\_accompanied (bool)

# is\_vip (bool)

# has\_heart\_condition (bool)

# is\_banned (bool)

def can\_enter\_ride():

    if has\_heart\_condition:

        print("Entry Denied: Health risk due to heart condition.")

    elif is\_vip and not is\_banned:

        print("Entry Allowed: VIP access granted.")

    elif is\_vip and is\_banned:

        print("Entry Denied: VIP banned from rides.")

    elif age >= 12 and height >= 140:

        if 12 <= age <= 15 and not is\_accompanied:

            print("Entry Denied: Ages 12-15 must be accompanied.")

        else:

            print("Entry Allowed: Meet age and height requirements.")

    else:

        print("Entry Denied: Does not meet age or height requirements.")

age = 13

height = 145

is\_accompanied = True

is\_vip = False

has\_heart\_condition = True

is\_banned = False

can\_enter\_ride()

import random

import matplotlib.pyplot as plt

# Initialize counters and totals

low\_count = 0

low\_total = 0

average\_count = 0

average\_total = 0

high\_count = 0

high\_total = 0

scores = []

# Generate and process 100 random numbers

for i in range(100):

    num = random.randint(1, 100)

    score = num // 10

    scores.append(score)

    if score <= 4:

        low\_count += 1

        low\_total += score

    elif score <= 7:

        average\_count += 1

        average\_total += score

    else:

        high\_count += 1

        high\_total += score

if low\_count > 0:

    low\_mean = low\_total / low\_count

else:

    low\_mean = 0

if average\_count > 0:

    average\_mean = average\_total / average\_count

else:

    average\_mean = 0

if high\_count > 0:

    high\_mean = high\_total / high\_count

else:

    high\_mean = 0

# Simple prints

print("Low Bucket (0-4):", low\_count, "scores, Mean =", round(low\_mean, 2))

print("Average Bucket (5-7):", average\_count, "scores, Mean =", round(average\_mean, 2))

print("High Bucket (8-10):", high\_count, "scores, Mean =", round(high\_mean, 2))

plt.scatter(range(100), scores, color='blue')

plt.title("Scatter Plot of Scores")

plt.show()