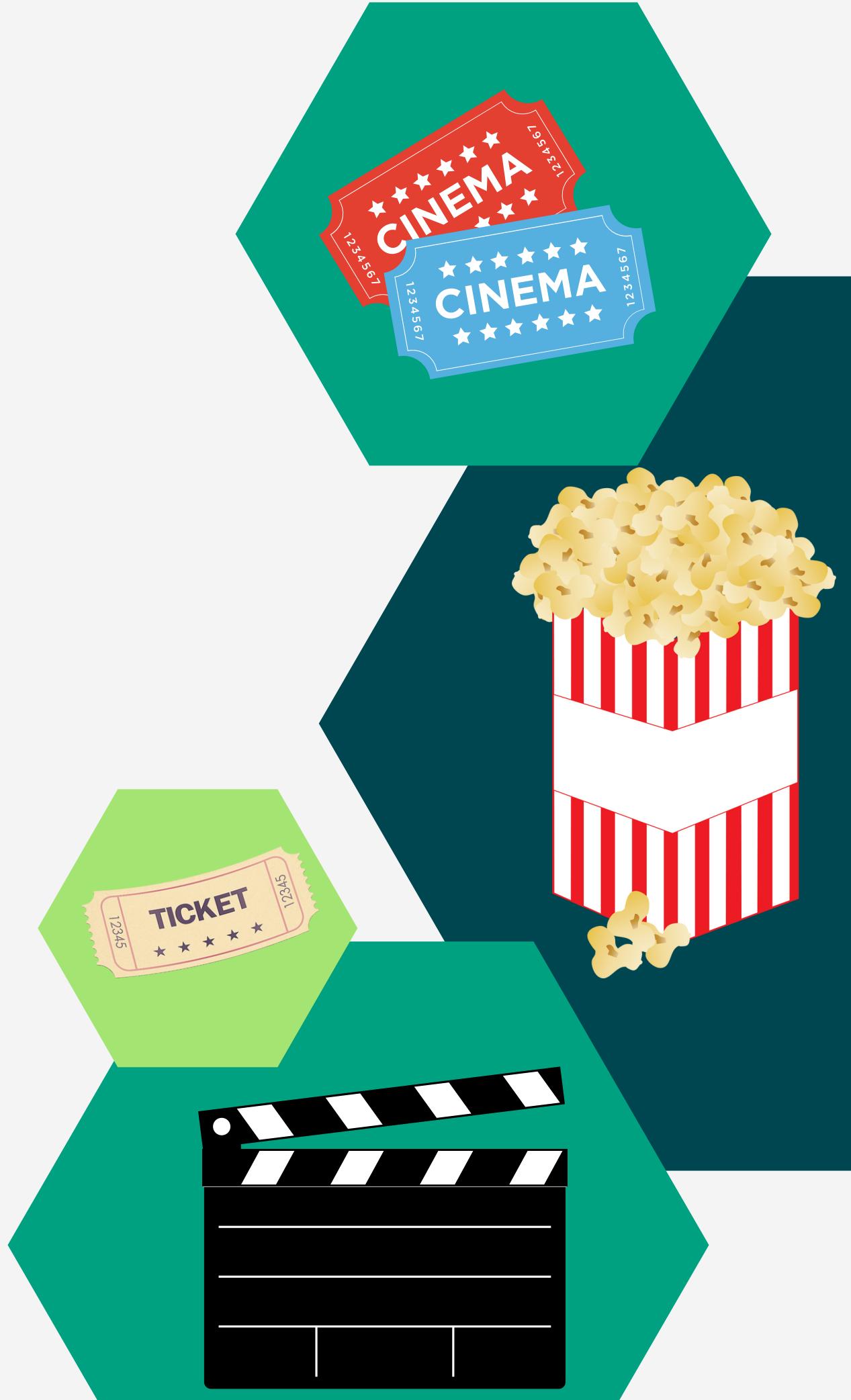


# Movie Theater Simulation

## Intro to Python Project

Insight group: Amal Almutairi, Nouf Aljohani,  
Rahaf Alzahrani, Rawan Alsudias & Salha Nasser



# Agenda

- Overview
- Movie theater simulation function
- Discount function
- Fixed Cost function
- Snacks Function

# Our Movie Theater



## Offers:

1. Snacks
2. 10% discount on Friday for senior citizens tickets.
3. 10% discount for prepurchased tickets.

# Movie Theater Simulation Function



# Function 1: Movie theater simulation function

```
● ● ●  
  
def theater_simulation():  
    # these are the variables that we need in and outside the function  
    global total_visitors ,total_children_visitors , total_seniors_visitors ,total_adults_visitors  
,week_days  
    # these are the variables we need only in the function  
    per_day_total_children_visitors= np.zeros((7,5),dtype=int)  
    per_day_total_adults_visitors= np.zeros((7,5),dtype=int)  
    per_day_total_seniors_visitors= np.zeros((7,5),dtype=int)  
    per_day_total_vistors = np.zeros((7,5),dtype=int)  
    r_mat = np.zeros((7,5))
```

# Function 1: Movie theater simulation function

```
# iterate through the week
for i in range(len(week_days)):
    # Keep track of total revenue for the day
    total_revenue =0
    # iterate through the amount of screens on a particular day
    for j in range(screens):
        #here we increase the probability of having more people in the weekends
        if i==5 or i==6:
            visitors_adults = random.randint(12,seats)

            visitors_senior = random.randint(0,seats - visitors_adults)

            visitors_children = random.randint(0,seats - (visitors_adults+visitors_senior))

        else:

            visitors_adults= random.randint(0,seats)

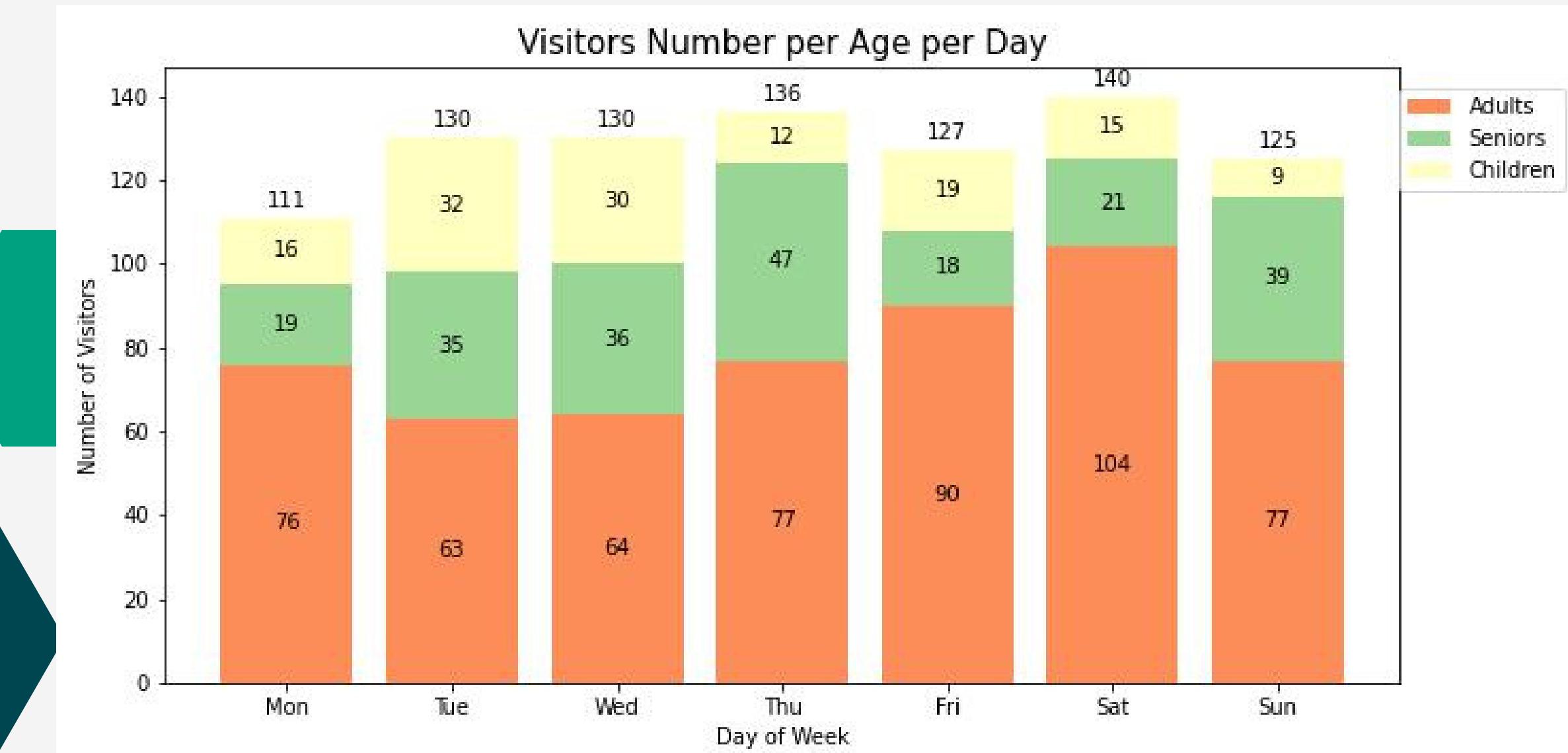
            visitors_senior=random.randint(0,seats-visitors_adults)

            visitors_children =random.randint(0,seats-(visitors_adults+visitors_senior))

    # cheack if it is Friday then there should be a 10% discount on senior citizen ticket costs
    if i==4:

        revenue_senior=vistors_senior*(senior_tickets-(senior_tickets*0.10))

    else:
        # Calculate the revenue for adults ,children, and senior
        revenue_adults = visitors_adults*adult_tickets
        revenue_children = visitors_children*children_tickets
        revenue_senior=vistors_senior*senior_tickets
```



# Discount Function

10%  
OFF

# Function 2 : Discount function

```
● ● ●  
  
def discount_ticket():  
  
    discount = 0.1  
  
    # Keep track of total discount for the day  
    global total_discount  
    total_discount = list()  
  
    # Total children visitors for each day (with presale tickets)  
    global total_visitors_children_a_day_with_presale_tickets  
    total_visitors_children_a_day_with_presale_tickets = list()  
  
    # Total seniors visitors for each day (with presale tickets)  
    global total_visitors_seniors_a_day_with_presale_tickets  
    total_visitors_seniors_a_day_with_presale_tickets = list()  
  
    # Total adults visitors for each day (with presale tickets)  
    global total_visitors_adults_a_day_with_presale_tickets  
    total_visitors_adults_a_day_with_presale_tickets = list()
```

# Function 2 : Discount function

```
for i in range(0,len(week_days)):

    # Calculate how many children, seniors and adults are tickets prebuyers

    visitors_adults_discount = total_adults_visitors[i] - (np.random.randint(0, total_adults_visitors[i]))
    total_visitors_adults_a_day_with_presale_tickets.append(visitors_adults_discount)

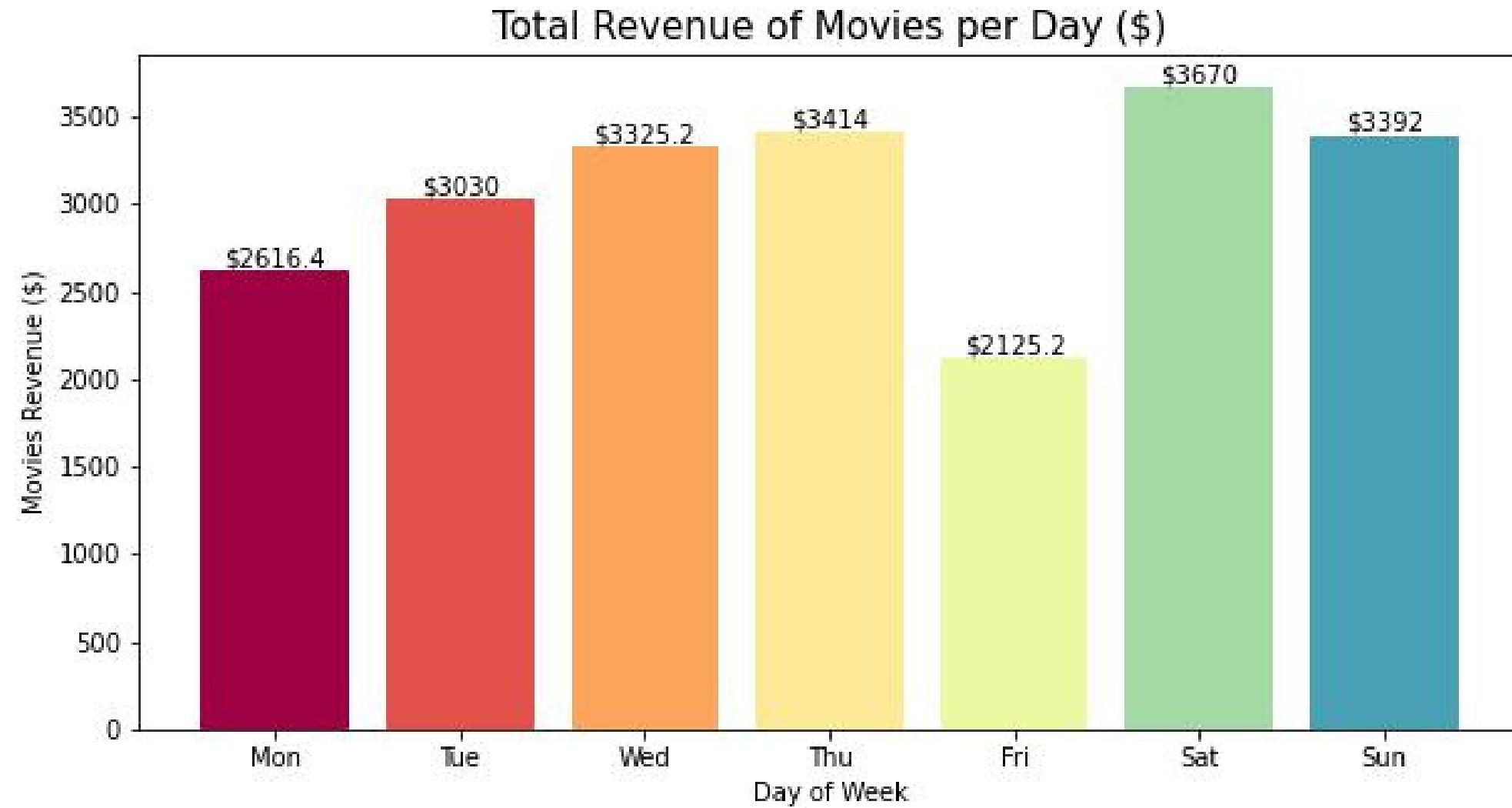
    visitors_children_discount = total_children_visitors[i] - (np.random.randint(0, total_children_visitors[i]))
    total_visitors_children_a_day_with_presale_tickets.append(visitors_children_discount)

    visitors_seniors_discount = total_seniors_visitors[i] - (np.random.randint(0, total_seniors_visitors[i]))
    total_visitors_seniors_a_day_with_presale_tickets.append(visitors_seniors_discount)

    # Calculate the discount for adults, children and seniors

    adults_discount = (adult_tickets * discount) * visitors_adults_discount
    children_discount = (children_tickets * discount) * visitors_children_discount
    seniors_discount = (senior_tickets * discount) * visitors_seniors_discount

    # Save the total discount to the corresponding day
    total_discount.append(round(adults_discount + children_discount + seniors_discount, 2))
```



# Fixed Cost Function



# Fixed Cost Function



```
#Add function to calculate property rent, insurance cost, and employee and manager costs
```

```
#fixed costs function
def fixedcost(monthly_rent,monthly_insurance_cost,monthly_employee_costs,monthly_manager_costs):
    """ calculate fixed costs monthly: property rent, insurance cost, and employee and manager costs"""
    total_monthly_fixed_cost= monthly_rent+monthly_insurance_cost+monthly_employee_costs+monthly_manager_costs
    print(f'total monthly fixed costs is {total_monthly_fixed_cost}')
```

```
#calculate total monthly fixed costs
fixedcost(
    monthly_rent=300000,
    monthly_insurance_cost=150000,
    monthly_employee_costs=90000,
    monthly_manager_costs=20000
)
```

# Snacks Function



# Function 4: Snacks

```
● ● ●

def snacks_func():
    global week_days, snack_revenue, snacks_week, s_mat, buyers_count

    # Calculate snacks revenue
    # a loop that goes through all 7 days

    for d in range(len(week_days)): # here start from 0-6 in indexing.
        i=0      # to help in indexing the snack_revenue matrix.

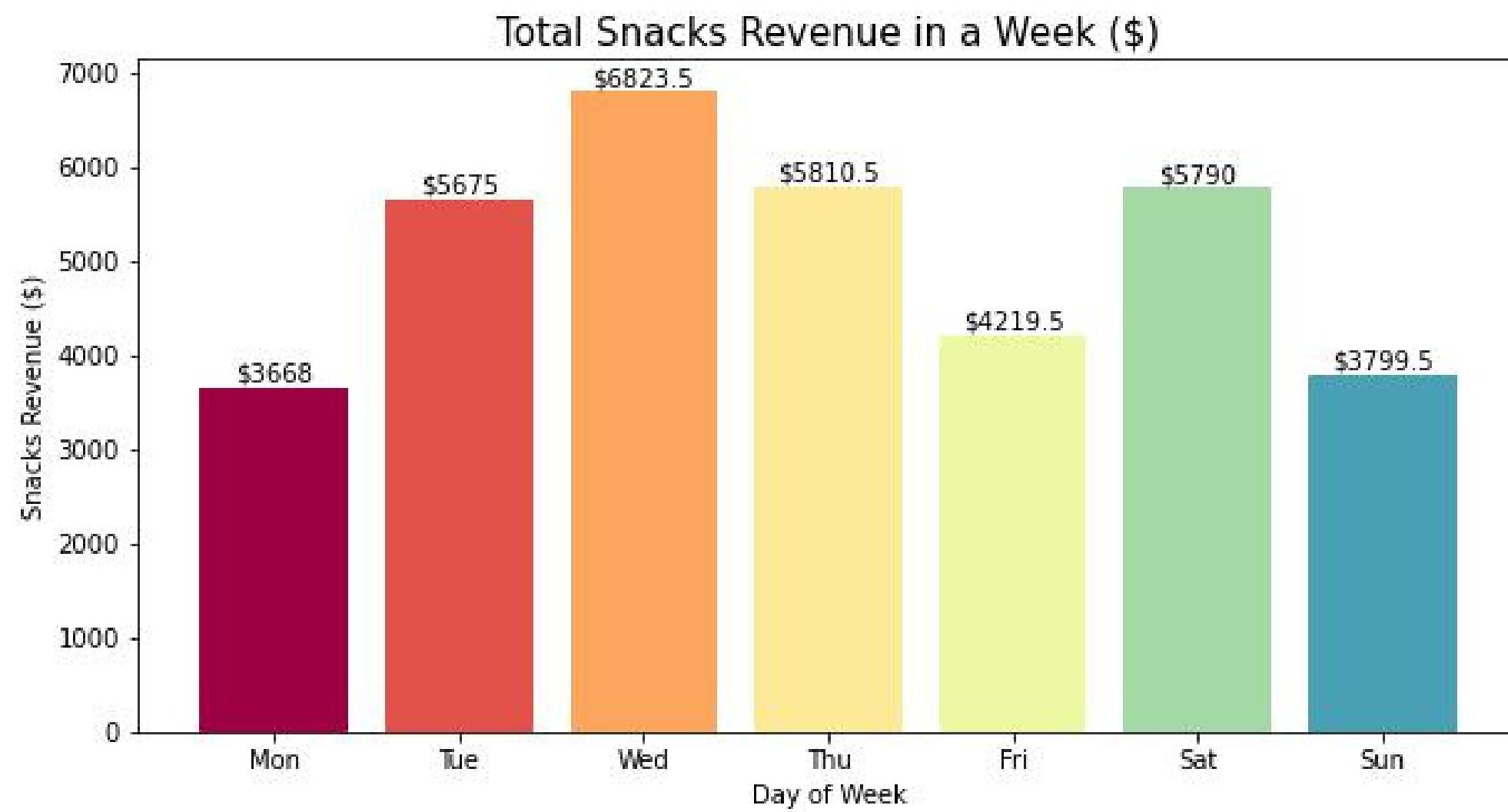
        for s in snacks: # here start from 0-3 # a loop that goes through all 4 snacks

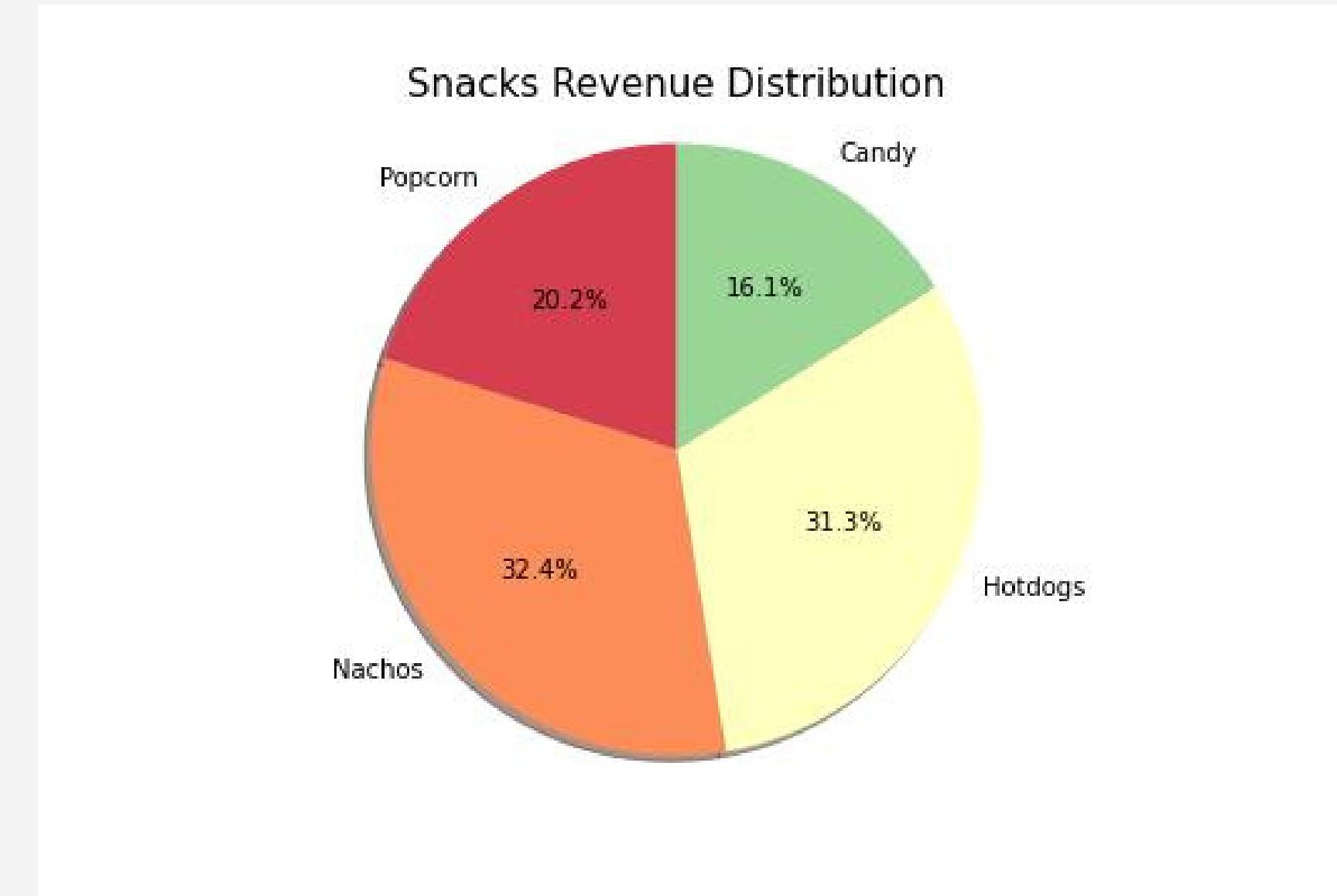
            # this variable is for storing the snack price multiplied by a random number of people
            # assuming 20% of total_visitors at least buy

            no_of_buyers = generator.integers(total_visitors[d]*0.2, total_visitors[d]*3, size=1)

            selected_snacks = snacks[s] * no_of_buyers
            # assign to the matrix
            snack_revenue[d,i] = selected_snacks
            buyers_count [d, i] = no_of_buyers
            i+=1

            # snacks revenue per day
            snacks_week[d] = np.sum(snack_revenue[d, :])
    # snacks revenue distribution
    s_mat = np.sum(snack_revenue, axis=0)
```





# Graph

