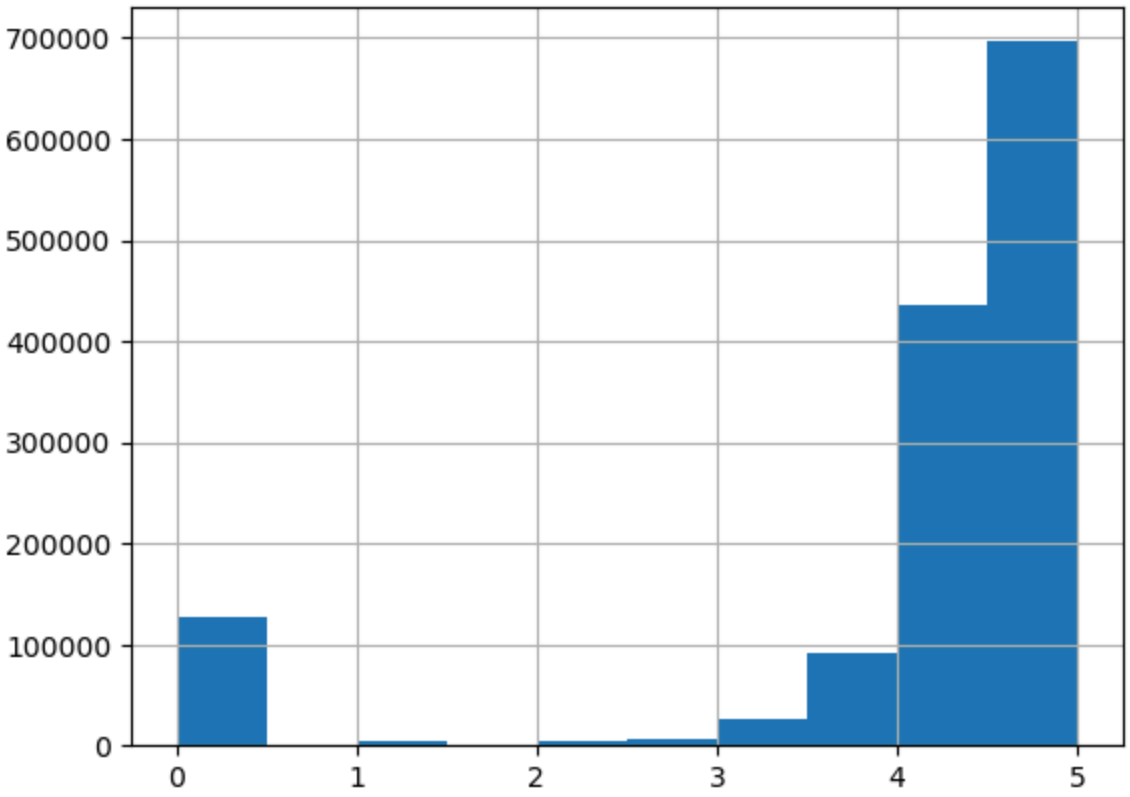
In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split

data = pd.read_csv("cleaned_data/amazon_product_cleaned.csv")
categories = pd.read_csv("cleaned_data/amazon_categories.csv")
data.head()
```

Out[1]:

	title	stars	price	listPrice	category_id	boughtInLastMonth	Discount
0	Sion Softside Expandable Roller Luggage, Black...	4.5	139.99	0.00	104	2000	1.000000
1	Luggage Sets Expandable PC+ABS Durable Suitcas...	4.5	169.99	209.99	104	1000	0.809515
2	Platinum Elite Softside Expandable Checked Lug...	4.6	365.49	429.99	104	300	0.849997
3	Freeform Hardside Expandable with Double Spinn...	4.6	291.59	354.37	104	400	0.822841
4	Winfield 2 Hardside Expandable Luggage with Sp...	4.5	174.99	309.99	104	400	0.564502

In [2]:
```
# Perform the histogram of ratings, x: rating, y: amount
data['stars'].hist(bins=10)
```

Out[2]:
```
<Axes: >
```



In [3]:
```
# We want to know which cagtegories is the most popular in Amazon website la
# for each category and sort it in descending way, print out the top 10 cate
top_categories = data.groupby('category_id')['boughtInLastMonth'].sum().sort
print(top_categories)
```

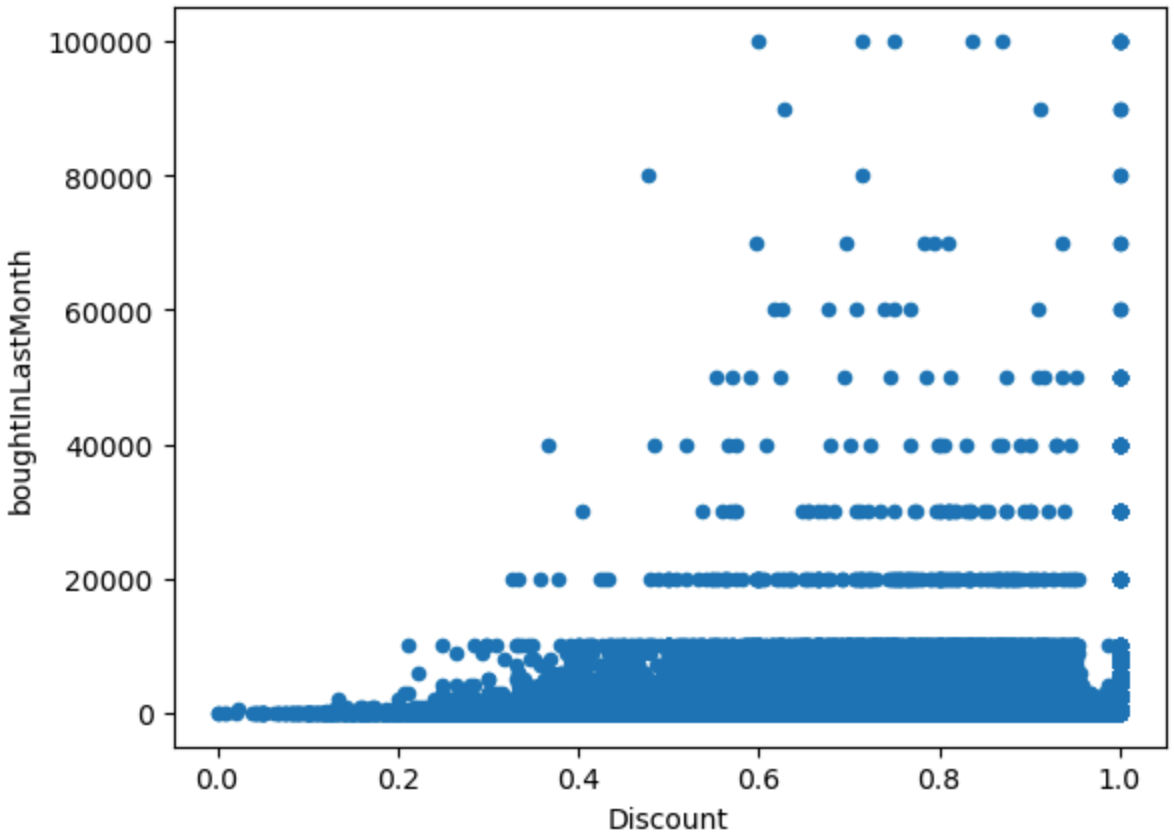
```
category_id
170    10389600
47      7922500
162     7057850
167     6783500
49      6381300
135     6039300
270     5746350
173     5338750
180     4537250
130     4248500
Name: boughtInLastMonth, dtype: int64
```

In [4]:
```
# As the amazon dataset only include the category ID, we combine the categor
# know which category it corresponding to, and print out the sales volume ar
top_categories_df = top_categories.reset_index()
top_categories_df.rename(columns={'category_id': 'id'}, inplace=True)
top_categories_merged = top_categories_df.merge(categories, on='id', how='le
print(top_categories_merged[['category_name', 'boughtInLastMonth']])
```

	category_name	boughtInLastMonth
0	Kitchen & Dining	10389600
1	Hair Care Products	7922500
2	Industrial & Scientific	7057850
3	Household Cleaning Supplies	6783500
4	Skin Care Products	6381300
5	Health & Household	6039300
6	Toys & Games	5746350
7	Home Storage & Organization	5338750
8	Dog Supplies	4537250
9	Household Supplies	4248500

In [5]:
```
# Sale volume and Discont scatter plot
data.plot.scatter(x='Discount', y='boughtInLastMonth')
```

Out[5]: <Axes: xlabel='Discount', ylabel='boughtInLastMonth'>



In [6]:
```
# Find the top 5 categories with highest and lowest rating
category_ratings = data.groupby('category_id')['stars'].mean().sort_values(a
category_ratings_df = category_ratings.reset_index()
category_ratings_df.rename(columns={'category_id': 'id'}, inplace=True)
category_ratings_merged = category_ratings_df.merge(categories, on='id', how
print("Top 5 Categories with Highest Average Rating:")
print(category_ratings_merged[['category_name', 'stars']].head())
print("\nTop 5 Categories with Lowest Average Rating:")
print(category_ratings_merged[['category_name', 'stars']].tail())
```

```
Top 5 Categories with Highest Average Rating:
             category_name     stars
0               Gift Cards  4.832374
1       Health & Household  4.568067
2  Industrial & Scientific  4.554931
3         Kitchen & Dining  4.544426
4       Household Supplies  4.543045

Top 5 Categories with Lowest Average Rating:
                                      category_name     stars
243  Xbox Series X & S Consoles, Games & Accessories  1.918826
244   Smart Home Thermostats - Compatibility Checker  1.161111
245           Virtual Reality Hardware & Accessories  0.996715
246                                 Kids' Play Boats  0.720438
247                                 Computer Servers  0.638790
```

In [7]:
```
# Create the correlation matrix between all the numerical variables
numeric_data = data.select_dtypes(include=['float64', 'int64'])
correlation_matrix = numeric_data.corr()
print(correlation_matrix)
```

```
                       stars     price  listPrice  category_id  \
stars               1.000000 -0.082226   0.025085    -0.011327
price              -0.082226  1.000000   0.198432    -0.036890
listPrice           0.025085  0.198432   1.000000    -0.009067
category_id        -0.011327 -0.036890  -0.009067     1.000000
boughtInLastMonth   0.062862 -0.027494   0.003206     0.013987
Discount           -0.077894  0.024231  -0.419558    -0.029922

                    boughtInLastMonth  Discount
stars                        0.062862 -0.077894
price                       -0.027494  0.024231
listPrice                    0.003206 -0.419558
category_id                  0.013987 -0.029922
boughtInLastMonth            1.000000 -0.088046
Discount                    -0.088046  1.000000
```

In [8]:
```
# OK let's do some sexual analysis which is interesting when researching on
data = data.merge(categories, how='left', left_on='category_id', right_on='i
data = data.drop(columns=['category_id', 'id'])
```

In [9]:
```
# I created a new categorical variable that indicate who's the target audien
# The judging condition is quite simple, just depend's on whether the catego
# I know some product categories are also designed for specific gender such
# as we have hundreds of them
men_keywords = ["Men"]
women_keywords = ["Women"]
children_keywords = ["Kids'", "Child", "Baby", "Toddler","Boys", "Girls"]

def categorize_audience(category_name):
    if any(keyword in category_name for keyword in men_keywords):
        return 'Men'
    elif any(keyword in category_name for keyword in women_keywords):
        return 'Women'
    elif any(keyword in category_name for keyword in children_keywords):
        return 'Children'
```

```
    else:
        return 'General'

# Apply the function to create a new column
data['target_audience'] = data['category_name'].apply(categorize_audience)
data.head()
```

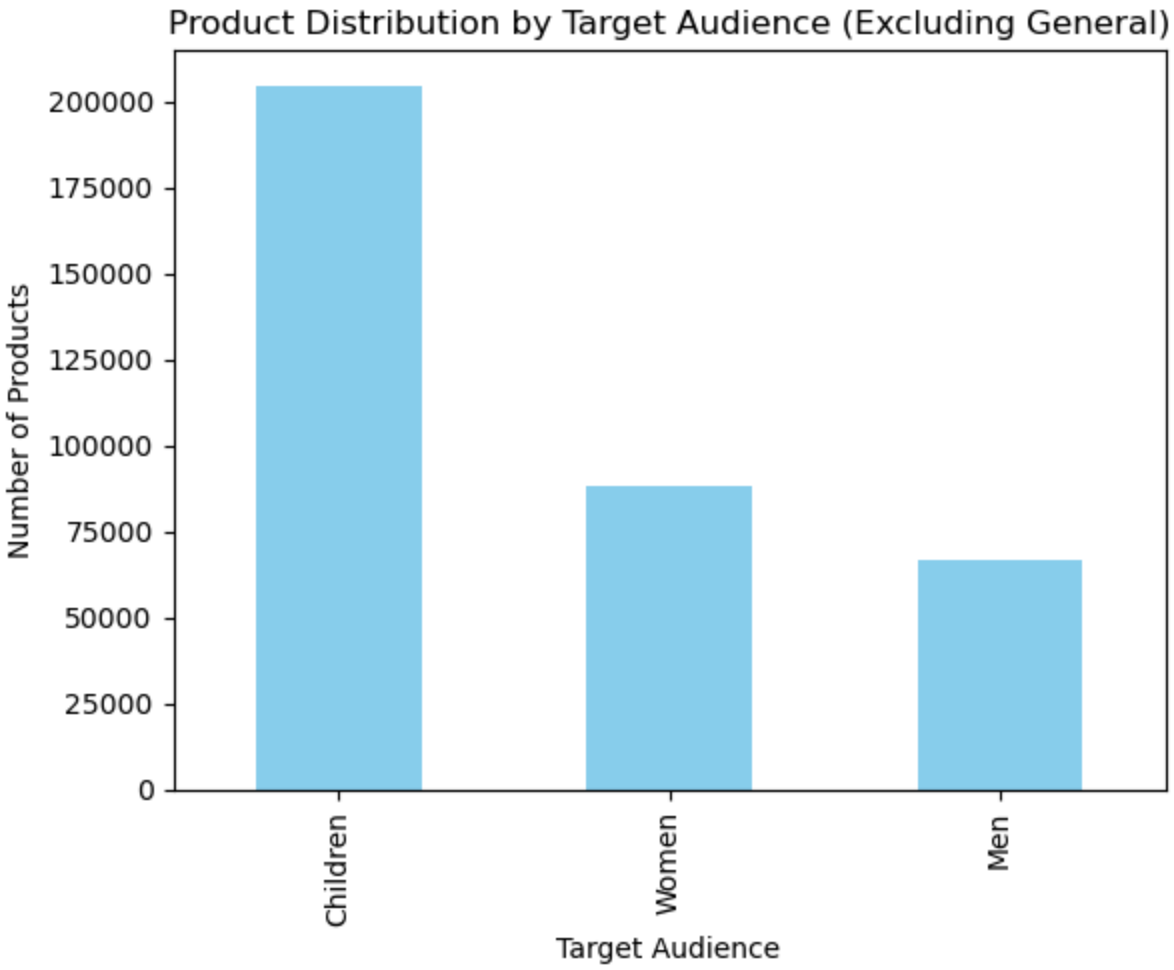
Out[9]:

	title	stars	price	listPrice	boughtInLastMonth	Discount	category_name
0	Sion Softside Expandable Roller Luggage, Black...	4.5	139.99	0.00	2000	1.000000	Suitcases
1	Luggage Sets Expandable PC+ABS Durable Suitcas...	4.5	169.99	209.99	1000	0.809515	Suitcases
2	Platinum Elite Softside Expandable Checked Lug...	4.6	365.49	429.99	300	0.849997	Suitcases
3	Freeform Hardside Expandable with Double Spinn...	4.6	291.59	354.37	400	0.822841	Suitcases
4	Winfield 2 Hardside Expandable Luggage with Sp...	4.5	174.99	309.99	400	0.564502	Suitcases

In [10]:

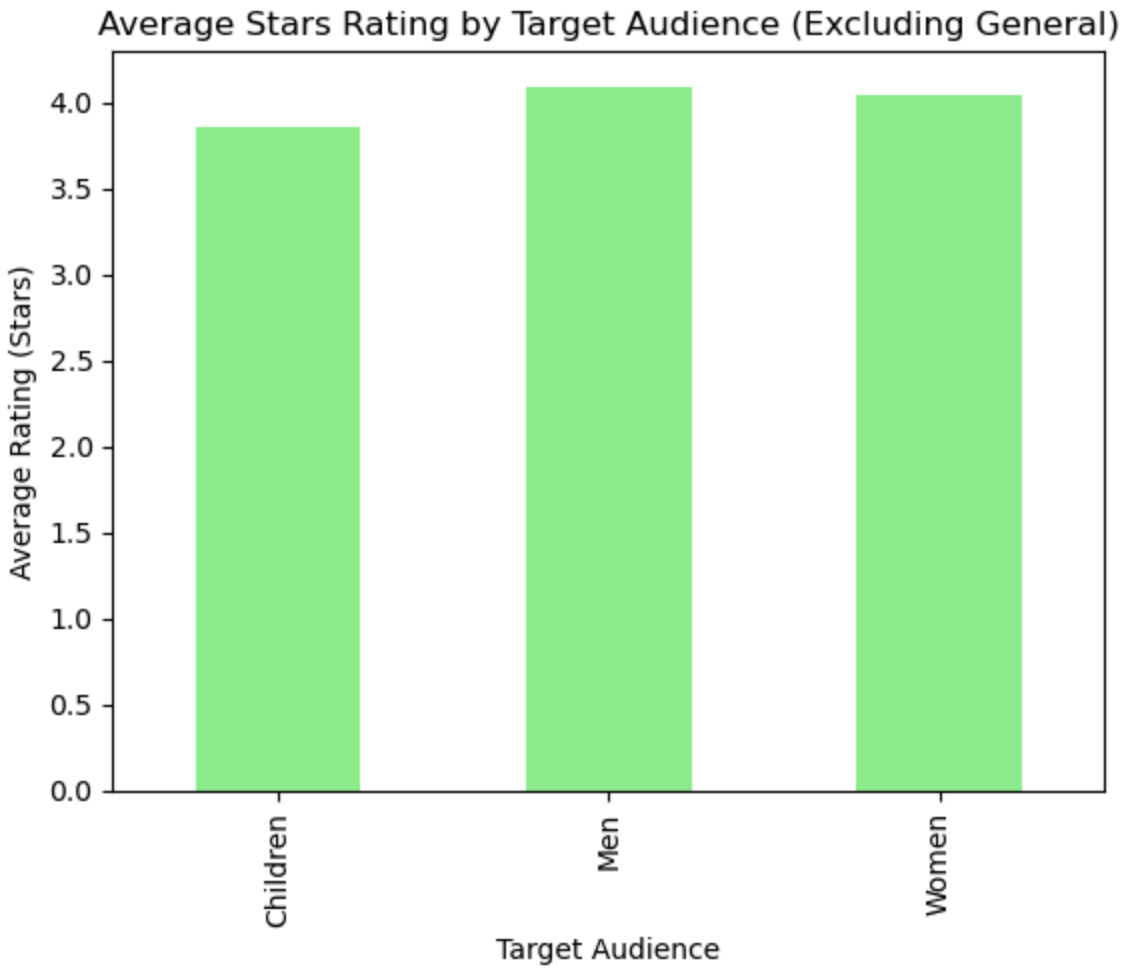
```
# Filter the data to exclude "General" target audience
filtered_data = data[data['target_audience'] != 'General']
# Count the number of products in each target audience, excluding "General"
audience_distribution = filtered_data['target_audience'].value_counts()
print(audience_distribution)

# Plot the distribution
import matplotlib.pyplot as plt

audience_distribution.plot(kind='bar', color='skyblue', title='Product Distr
plt.xlabel('Target Audience')
plt.ylabel('Number of Products')
plt.show()
```

```
target_audience
Children    204574
Women        88207
Men          66610
Name: count, dtype: int64
```



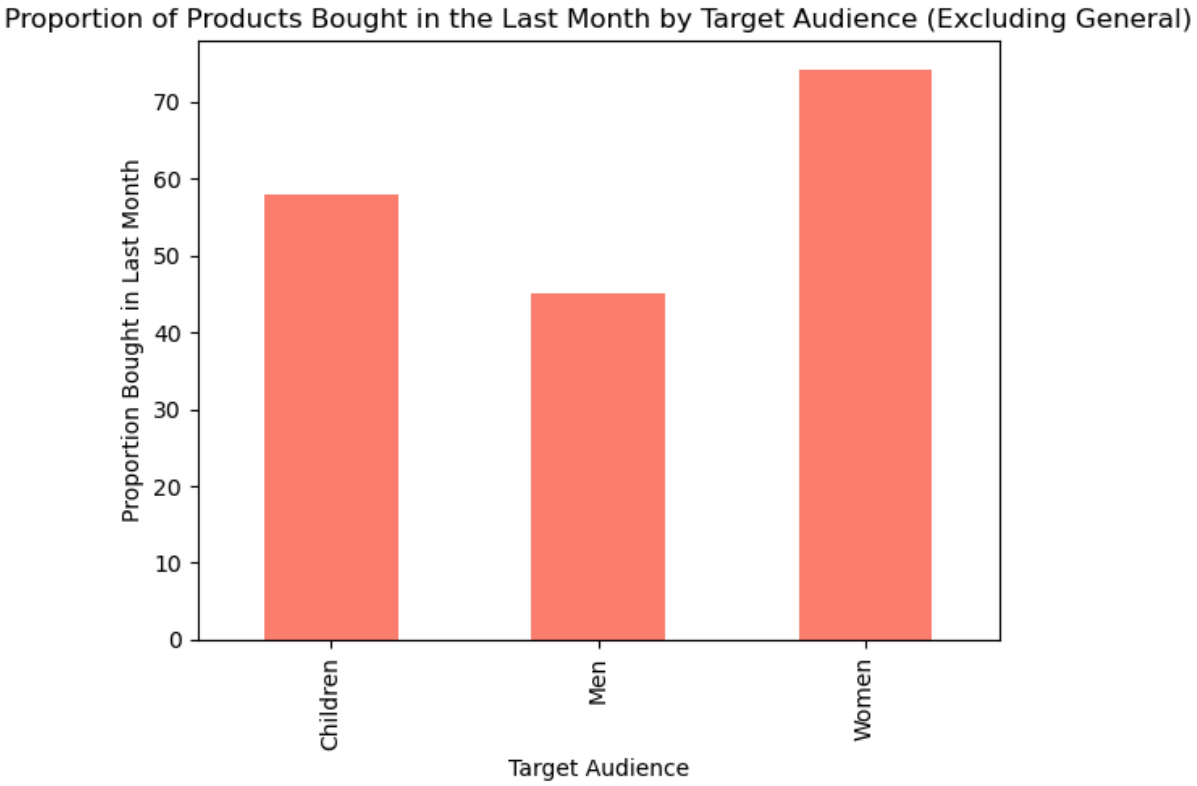
In [11]:
```
# Average stars rating by target audience, excluding "General"
avg_rating_by_audience = filtered_data.groupby('target_audience')['stars'].m
print(avg_rating_by_audience)

# Plot the average rating
avg_rating_by_audience.plot(kind='bar', color='lightgreen', title='Average S
plt.xlabel('Target Audience')
plt.ylabel('Average Rating (Stars)')
plt.show()
```

```
target_audience
Children    3.861998
Men         4.091323
Women       4.040940
Name: stars, dtype: float64
```



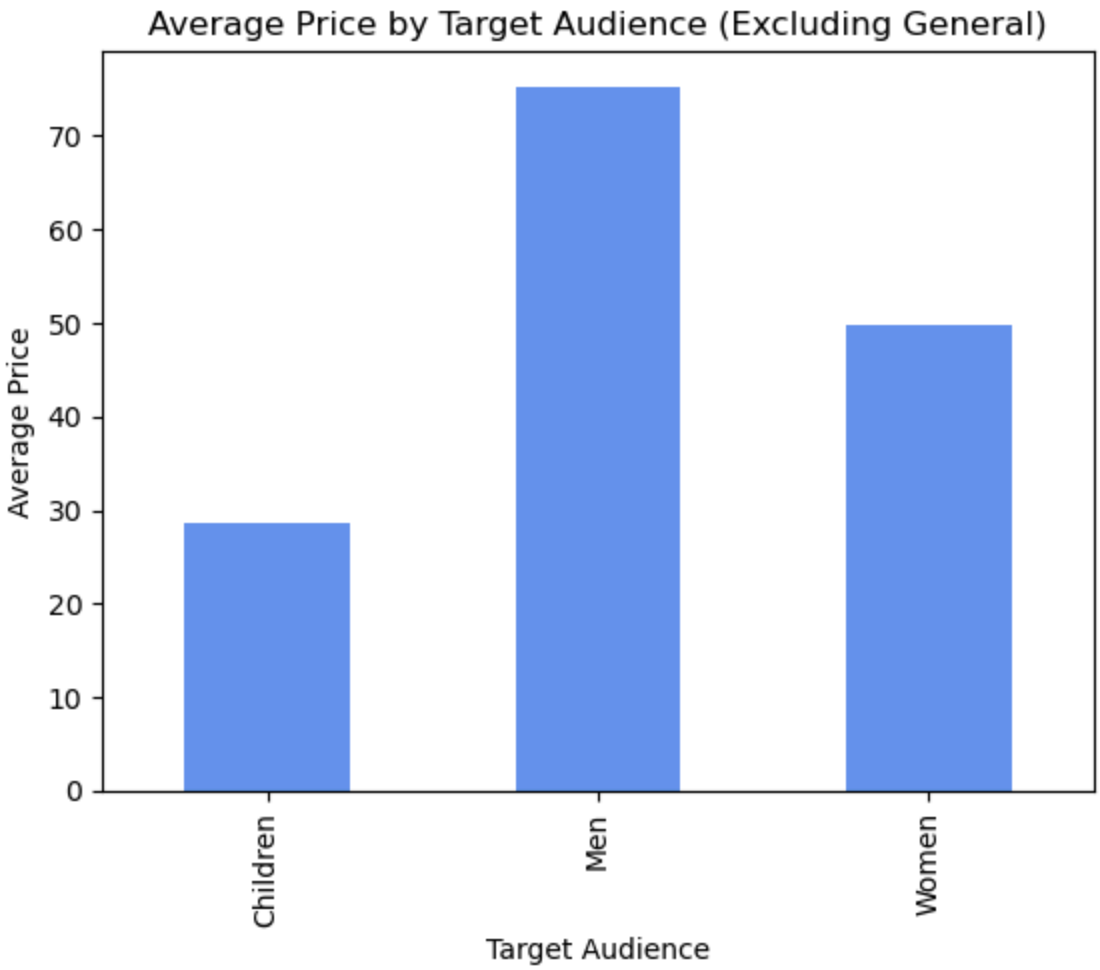
In [12]:
```
recent_sales = filtered_data.groupby('target_audience')['boughtInLastMonth']
print(recent_sales)

recent_sales.plot(kind='bar', color='salmon', title='Proportion of Products
plt.xlabel('Target Audience')
plt.ylabel('Proportion Bought in Last Month')
plt.show()
```

```
target_audience
Children    57.998328
Men         45.104339
Women       74.224268
Name: boughtInLastMonth, dtype: float64
```



In [13]:
```
avg_price_by_audience = filtered_data.groupby('target_audience')['price'].me
print(avg_price_by_audience)

avg_price_by_audience.plot(kind='bar', color='cornflowerblue', title='Averag
plt.xlabel('Target Audience')
plt.ylabel('Average Price')
plt.show()
```

```
target_audience
Children    28.527046
Men         75.267237
Women       49.709230
Name: price, dtype: float64
```



In [15]:
```
data = data.drop(columns=['title', 'category_name'])
data = pd.get_dummies(data, columns=['target_audience'], drop_first=True)
# Calculate the correlation matrix
correlation_matrix = data.corr()

# Get the correlation of all variables with the 'stars' column
correlation_with_stars = correlation_matrix['stars']

# Display the correlations
print(correlation_with_stars)
```

```
stars                       1.000000
price                      -0.082226
listPrice                   0.025085
boughtInLastMonth           0.062862
Discount                   -0.077894
target_audience_General     0.025027
target_audience_Men         0.014431
target_audience_Women       0.006950
Name: stars, dtype: float64
```

In [ ]: