In [1]:
```python
# data is taken from the kaggle

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [65]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
for dirname, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [66]:
```python
import kagglehub

# Download latest version
path = kagglehub.dataset_download("tamsnd/adidas-webstore-shoe-data")

print("Path to dataset files:", path)
```

```
Path to dataset files: /root/.cache/kagglehub/datasets/tamsnd/adidas-webstore-shoe-data/versions/11
```

In [67]:
```python
#data is taken from the kaggle

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/shoes.csv')
print(df.info()) # get information about column and rows
print(df.describe())
print(df.head()) #view the first few rows
```

```
<ipython-input-67-bb799b6e59f8>:3: DtypeWarning: Columns (5) have mixed types. Specify dtype option on import or set low_memory
=False.
  df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/shoes.csv')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 746189 entries, 0 to 746188
Data columns (total 13 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   name           746189 non-null  object
 1   id             746189 non-null  object
 2   price          746189 non-null  float64
 3   category       746189 non-null  object
 4   color          746189 non-null  object
 5   weight         282411 non-null  object
 6   best_for_wear  277978 non-null  object
 7   size           746189 non-null  object
 8   availability   746189 non-null  int64
 9   image_url      746189 non-null  object
 10  gender         683030 non-null  object
 11  date           746189 non-null  object
 12  country_code   746189 non-null  object
dtypes: float64(1), int64(1), object(11)
memory usage: 74.0+ MB
None
              price   availability
count  746189.00000  746189.000000
mean       96.21052       6.162168
std        55.85658       6.860171
min        16.00000       0.000000
25%        58.50000       0.000000
50%        84.00000       1.000000
75%       120.00000      15.000000
max       750.00000      15.000000
             name      id  price     category  \
0  Samba OG Schuh  B75806  120.0  en/trainers
1  Samba OG Schuh  B75806  120.0  en/trainers
2  Samba OG Schuh  B75806  120.0  en/trainers
3  Samba OG Schuh  B75806  120.0  en/trainers
4  Samba OG Schuh  B75806  120.0  en/trainers

                                      color  weight best_for_wear    size  \
0  Cloud White / Core Black / Clear Granite  {None}           NaN  35 1/2
1  Cloud White / Core Black / Clear Granite  {None}           NaN      36
2  Cloud White / Core Black / Clear Granite  {None}           NaN  36 2/3
```

```
3  Cloud White / Core Black / Clear Granite  {None}          NaN  37 1/3
4  Cloud White / Core Black / Clear Granite  {None}          NaN      38

   availability                                        image_url gender  \
0            15  https://assets.adidas.com/images/w_600,f_auto,...    NaN
1            15  https://assets.adidas.com/images/w_600,f_auto,...    NaN
2            15  https://assets.adidas.com/images/w_600,f_auto,...    NaN
3            15  https://assets.adidas.com/images/w_600,f_auto,...    NaN
4            15  https://assets.adidas.com/images/w_600,f_auto,...    NaN

         date country_code
0  04/01/2025           DE
1  04/01/2025           DE
2  04/01/2025           DE
3  04/01/2025           DE
4  04/01/2025           DE
```

In [68]:
```
print(df['date'])
```

```
0         04/01/2025
1         04/01/2025
2         04/01/2025
3         04/01/2025
4         04/01/2025
             ...
746184    2025-01-15
746185    2025-01-15
746186    2025-01-15
746187    2025-01-15
746188    2025-01-15
Name: date, Length: 746189, dtype: object
```

In [69]:
```
# Print every stats for each IDs
print(df.groupby(['id']).count())
```

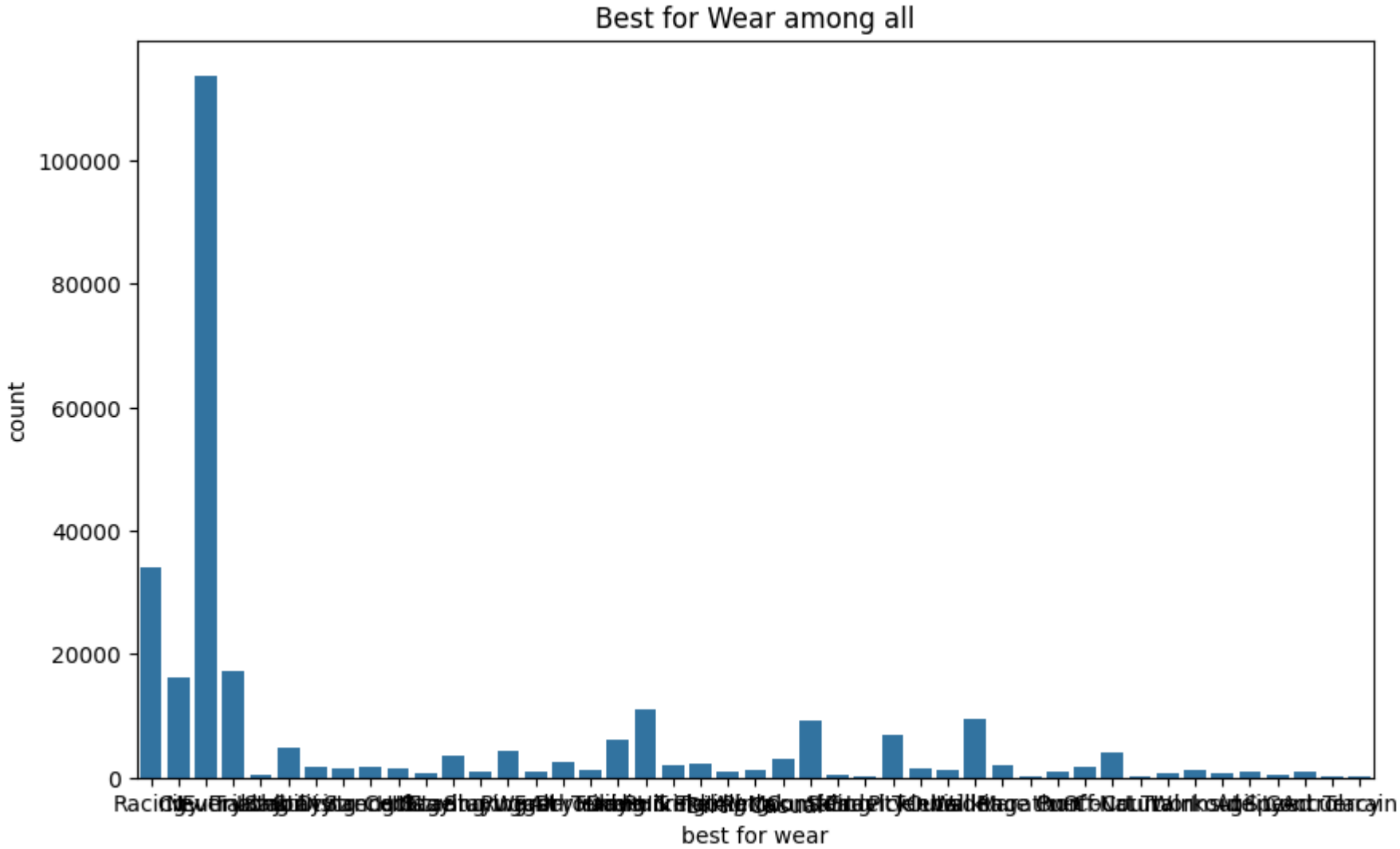
id	name	price	category	color	weight	best_for_wear	size	availability	image_url	gender	date	country_code
011040	200	200	200	200	20	200	200	200	200	180	200	200
015110	288	288	288	288	24	288	288	288	288	264	288	288
019228	200	200	200	200	20	200	200	200	200	180	200	200
019310	200	200	200	200	20	200	200	200	200	180	200	200
019351	46	46	46	46	0	0	46	46	46	46	46	46
...	...	...	...	...	...	...	...	...	...	...	...	...
Q47235	34	34	34	34	0	34	34	34	34	34	34	34
S29146	189	189	189	189	189	189	189	189	189	189	189	189
S75104	350	350	350	350	44	0	350	350	350	306	350	350
S79916	5	5	5	5	0	0	5	5	5	5	5	5
S81020	26	26	26	26	0	0	26	26	26	26	26	26

[4842 rows x 12 columns]

In [70]:
```
best=df['best_for_wear']
print(best)
```

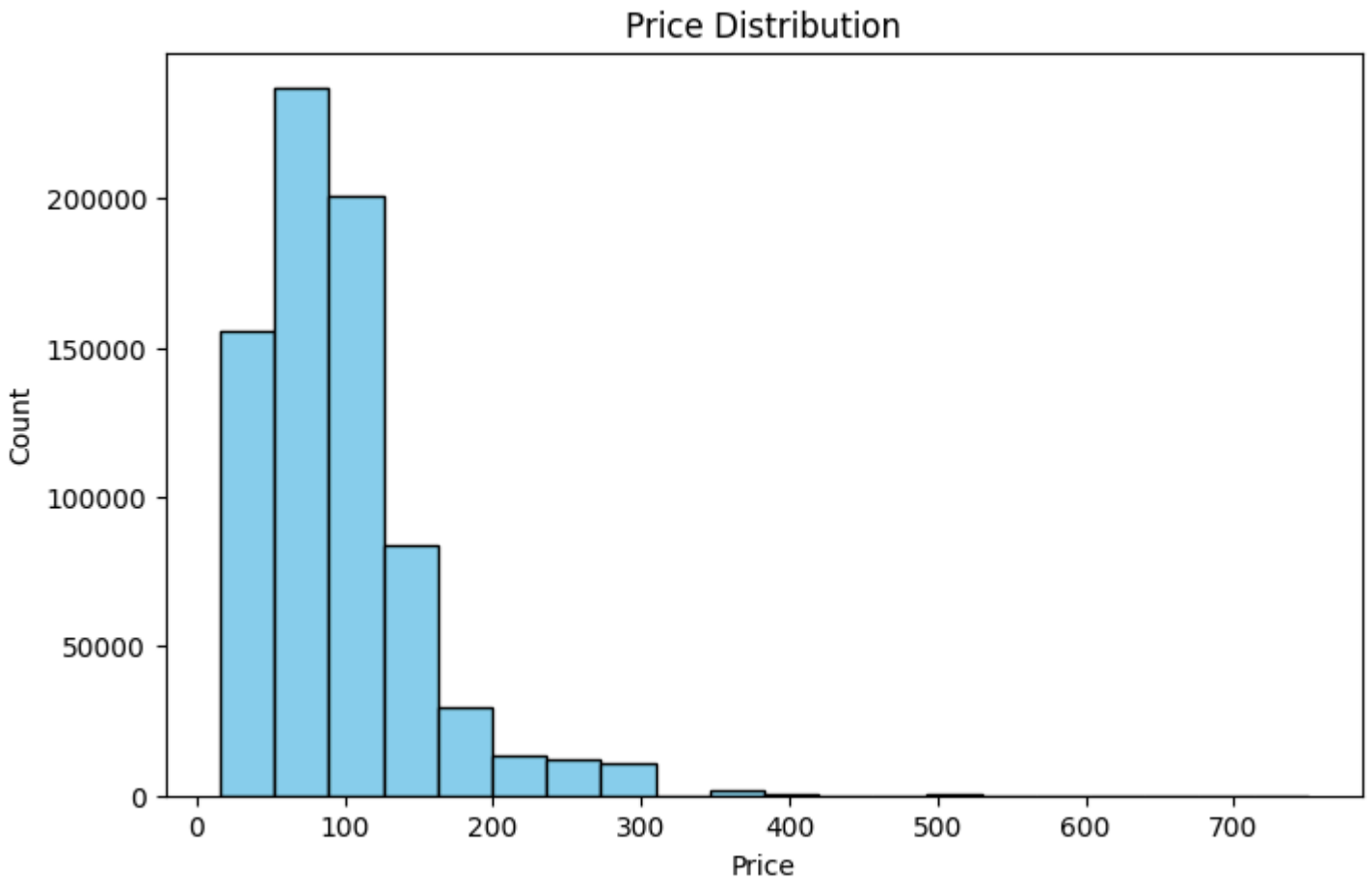
```
0                  NaN
1                  NaN
2                  NaN
3                  NaN
4                  NaN
              ...
746184    Padel Tennis
746185    Padel Tennis
746186    Padel Tennis
746187    Padel Tennis
746188    Padel Tennis
Name: best_for_wear, Length: 746189, dtype: object
```

In [71]:
```
plt.figure(figsize=(10,6))
sns.countplot(x='best_for_wear', data=df)
plt.title('Best for Wear among all')
plt.xlabel('best for wear')
plt.ylabel('count')
plt.show()
```



In [72]:
```
#histogram
plt.figure(figsize=(8,5))
plt.hist(df['price'], bins=20, color='skyblue', edgecolor='black')
plt.title('Price Distribution')
# sns.distplot(df['price'])
plt.xlabel('Price')
```

```
plt.ylabel('Count')
plt.show()
```



In [73]:
```
X = df['id']  # Features
y = df['price']  # Target
```

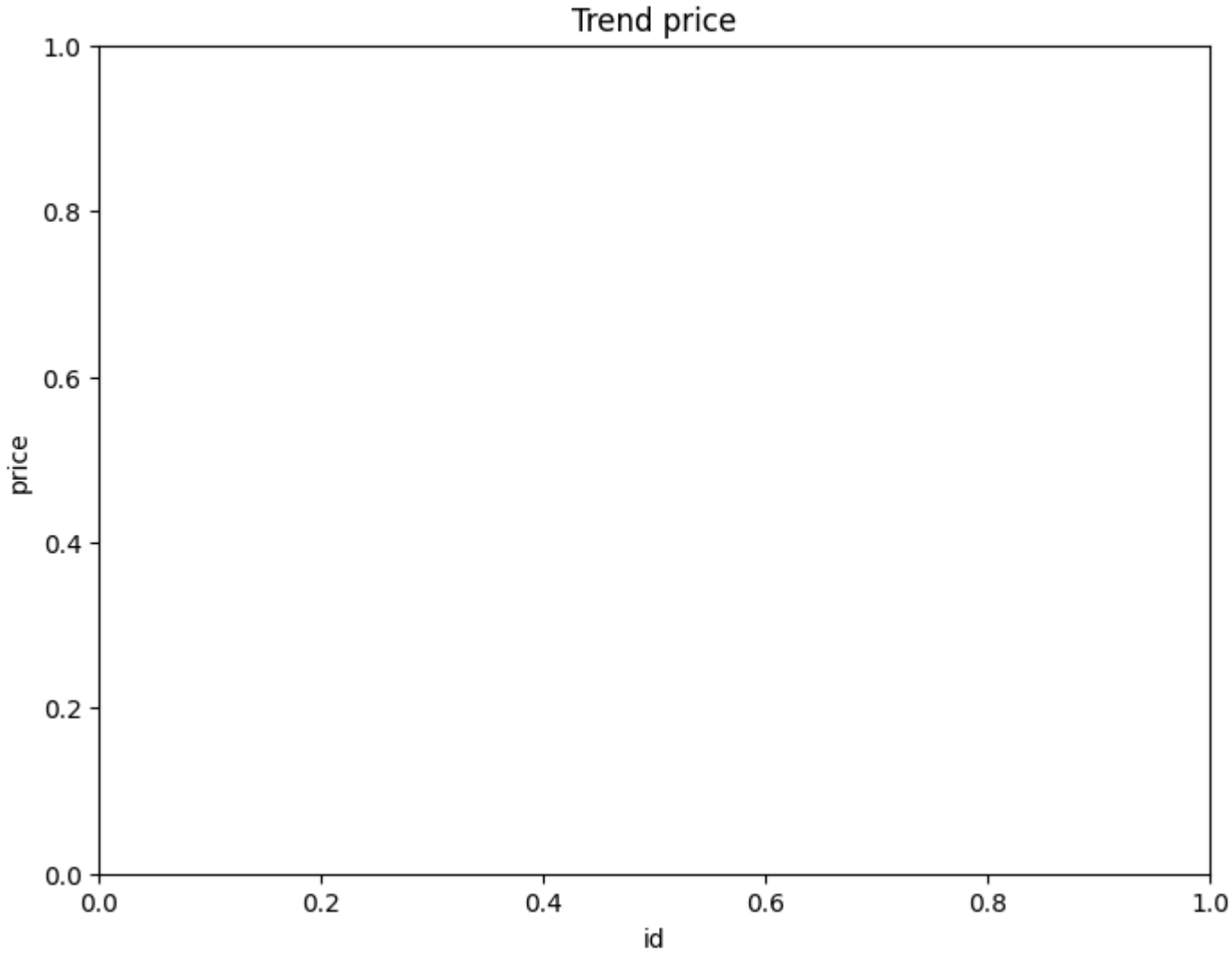
In [74]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2529)
```

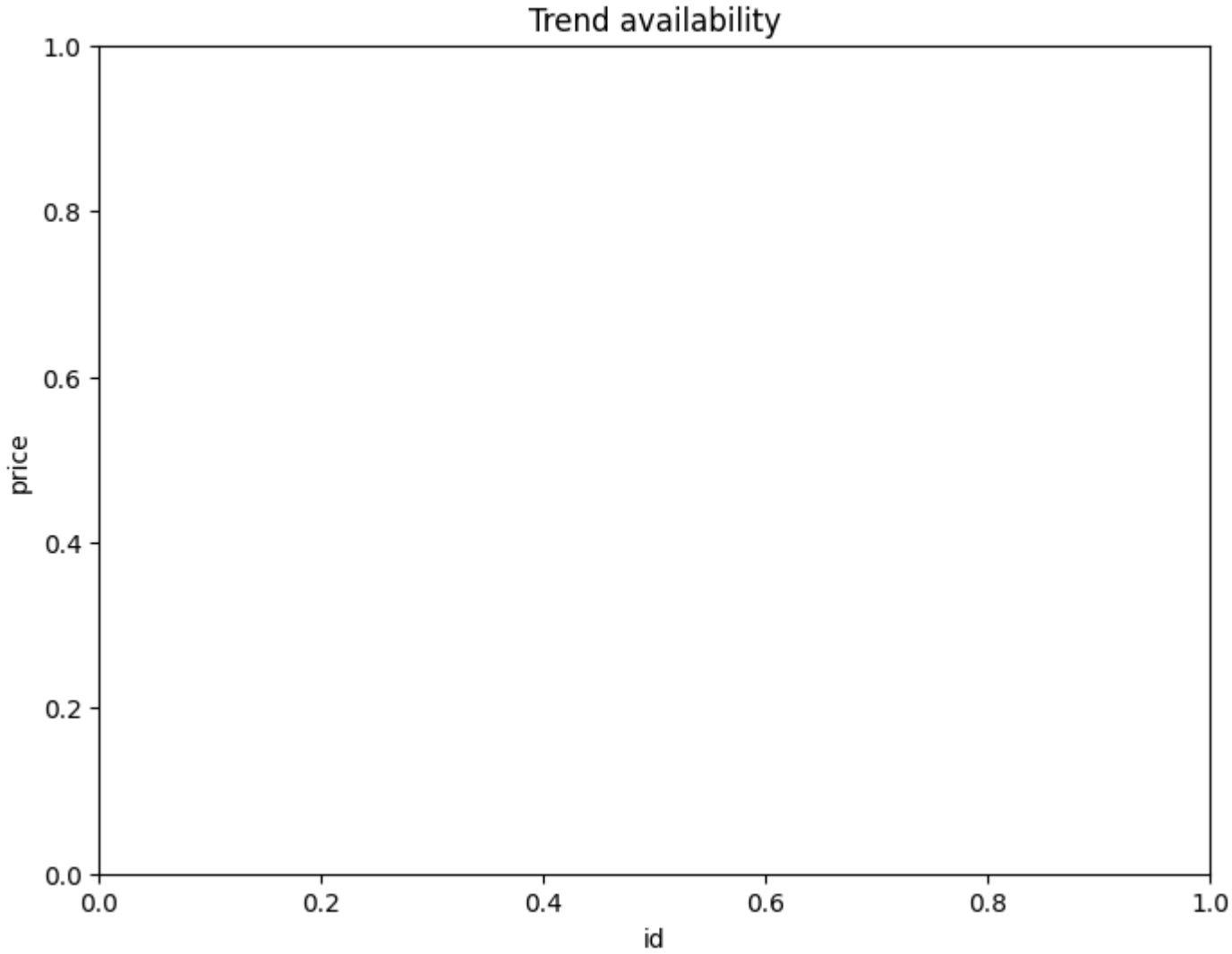
In [ ]:
```
# model=LinearRegression()
# model.fit(X_train, y_train)
```

In [77]:
```
# plt.figure(figsize=(8,6))
# plt.title('Trend price')
# plt.xlabel('id')
# plt.ylabel('price')
# plt.show()

# plt.figure(figsize=(8,6))

# plt.title('Trend availability')
# plt.xlabel('id')
# plt.ylabel('price')
# plt.show()
```


Trend price
1.0
0.8
0.6
0.4
0.2
0.0
price
0.0
0.2
0.4
0.6
0.8
1.0
id


Trend availability
1.0
0.8
0.6
0.4
0.2
0.0
price
0.0
0.2
0.4
0.6
0.8
1.0
id