In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import plotly.express as px
```

In [2]:
```
df = pd.read_csv(r'C:\Users\Samdure\Downloads\retail_price.csv')
```

In [3]:
```
df
```

Out[3]:

	product_id	product_category_name	month_year	qty	total_price	freight_price	unit_price	proc
0	bed1	bed_bath_table	01-05-2017	1	45.95	15.100000	45.950000	
1	bed1	bed_bath_table	01-06-2017	3	137.85	12.933333	45.950000	
2	bed1	bed_bath_table	01-07-2017	6	275.70	14.840000	45.950000	
3	bed1	bed_bath_table	01-08-2017	4	183.80	14.287500	45.950000	
4	bed1	bed_bath_table	01-09-2017	2	91.90	15.100000	45.950000	
...	...	...	...	...	...	...	...	
671	bed5	bed_bath_table	01-05-2017	1	215.00	8.760000	215.000000	
672	bed5	bed_bath_table	01-06-2017	10	2090.00	21.322000	209.000000	
673	bed5	bed_bath_table	01-07-2017	59	12095.00	22.195932	205.000000	
674	bed5	bed_bath_table	01-08-2017	52	10375.00	19.412885	199.509804	
675	bed5	bed_bath_table	01-09-2017	32	5222.36	24.324687	163.398710	

676 rows × 30 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 676 entries, 0 to 675
Data columns (total 30 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   product_id                  676 non-null    object 
 1   product_category_name       676 non-null    object 
 2   month_year                  676 non-null    object 
 3   qty                         676 non-null    int64  
 4   total_price                 676 non-null    float64
 5   freight_price               676 non-null    float64
 6   unit_price                  676 non-null    float64
 7   product_name_lenght         676 non-null    int64  
 8   product_description_lenght  676 non-null    int64  
 9   product_photos_qty          676 non-null    int64  
 10  product_weight_g            676 non-null    int64  
 11  product_score               676 non-null    float64
 12  customers                   676 non-null    int64  
 13  weekday                     676 non-null    int64  
 14  weekend                     676 non-null    int64  
 15  holiday                     676 non-null    int64  
 16  month                       676 non-null    int64  
 17  year                        676 non-null    int64  
 18  s                           676 non-null    float64
 19  volume                      676 non-null    int64  
 20  comp_1                      676 non-null    float64
 21  ps1                         676 non-null    float64
 22  fp1                         676 non-null    float64
 23  comp_2                      676 non-null    float64
 24  ps2                         676 non-null    float64
 25  fp2                         676 non-null    float64
 26  comp_3                      676 non-null    float64
 27  ps3                         676 non-null    float64
 28  fp3                         676 non-null    float64
 29  lag_price                   676 non-null    float64
dtypes: float64(15), int64(12), object(3)
memory usage: 158.6+ KB
```

In [5]:
```
df.isnull()
```

Out[5]:

	product_id	product_category_name	month_year	qty	total_price	freight_price	unit_price	pro
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
671	False	False	False	False	False	False	False	
672	False	False	False	False	False	False	False	
673	False	False	False	False	False	False	False	
674	False	False	False	False	False	False	False	
675	False	False	False	False	False	False	False	

676 rows × 30 columns

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
product_id                  0
product_category_name       0
month_year                  0
qty                         0
total_price                 0
freight_price               0
unit_price                  0
product_name_lenght         0
product_description_lenght  0
product_photos_qty          0
product_weight_g            0
product_score               0
customers                   0
weekday                     0
weekend                     0
holiday                     0
month                       0
year                        0
s                           0
volume                      0
comp_1                      0
ps1                         0
fp1                         0
comp_2                      0
ps2                         0
fp2                         0
comp_3                      0
ps3                         0
fp3                         0
lag_price                   0
dtype: int64
```

In [7]:
```
df.describe()
```

Out[7]:

	qty	total_price	freight_price	unit_price	product_name_lenght	product_description_l
count	676.000000	676.000000	676.000000	676.000000	676.000000	676.0
mean	14.495562	1422.708728	20.682270	106.496800	48.720414	767.3
std	15.443421	1700.123100	10.081817	76.182972	9.420715	655.2
min	1.000000	19.900000	0.000000	19.900000	29.000000	100.0
25%	4.000000	333.700000	14.761912	53.900000	40.000000	339.0
50%	10.000000	807.890000	17.518472	89.900000	51.000000	501.0
75%	18.000000	1887.322500	22.713558	129.990000	57.000000	903.0
max	122.000000	12095.000000	79.760000	364.000000	60.000000	3006.0

8 rows × 27 columns

In [8]:
```
print(f"Columns: {df.shape[1]}\nSamples: {df.shape[0]}")
```

```
Columns: 30
Samples: 676
```

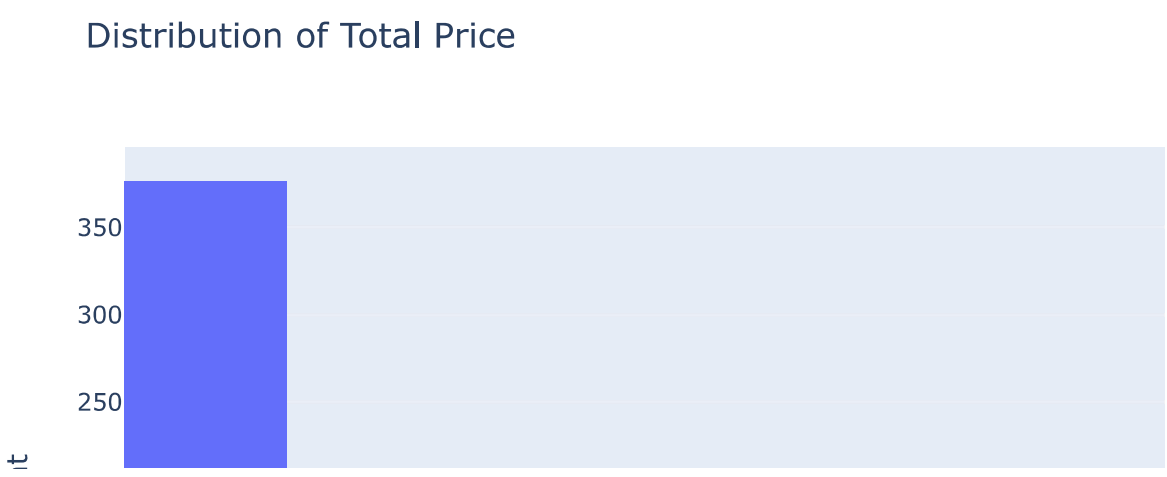
In [9]:
```
df.isna
```

Out[9]:
```
<bound method DataFrame.isna of     product_id product_category_name  month_year  qty
total_price  \
0         bed1        bed_bath_table  01-05-2017    1        45.95
1         bed1        bed_bath_table  01-06-2017    3       137.85
2         bed1        bed_bath_table  01-07-2017    6       275.70
3         bed1        bed_bath_table  01-08-2017    4       183.80
4         bed1        bed_bath_table  01-09-2017    2        91.90
..         ...                   ...         ...  ...          ...
671       bed5        bed_bath_table  01-05-2017    1       215.00
672       bed5        bed_bath_table  01-06-2017   10      2090.00
673       bed5        bed_bath_table  01-07-2017   59     12095.00
674       bed5        bed_bath_table  01-08-2017   52     10375.00
675       bed5        bed_bath_table  01-09-2017   32      5222.36

     freight_price  unit_price  product_name_lenght  \
0        15.100000   45.950000                   39
1        12.933333   45.950000                   39
2        14.840000   45.950000                   39
3        14.287500   45.950000                   39
4        15.100000   45.950000                   39
..             ...         ...                  ...
671       8.760000  215.000000                   56
672      21.322000  209.000000                   56
673      22.195932  205.000000                   56
674      19.412885  199.509804                   56
675      24.324687  163.398710                   56

     product_description_lenght  product_photos_qty  ...  comp_1  ps1  \
0                           161                   2  ...    89.9  3.9
1                           161                   2  ...    89.9  3.9
2                           161                   2  ...    89.9  3.9
3                           161                   2  ...    89.9  3.9
4                           161                   2  ...    89.9  3.9
..                          ...                 ...  ...     ...  ...
671                         162                   5  ...    89.9  3.9
672                         162                   5  ...    89.9  3.9
673                         162                   5  ...    89.9  3.9
674                         162                   5  ...    89.9  3.9
675                         162                   5  ...    89.9  3.9

           fp1      comp_2  ps2        fp2  comp_3  ps3        fp3   lag_price
0    15.011897  215.000000  4.4   8.760000   45.95  4.0  15.100000   45.900000
1    14.769216  209.000000  4.4  21.322000   45.95  4.0  12.933333   45.950000
2    13.993833  205.000000  4.4  22.195932   45.95  4.0  14.840000   45.950000
3    14.656757  199.509804  4.4  19.412885   45.95  4.0  14.287500   45.950000
4    18.776522  163.398710  4.4  24.324687   45.95  4.0  15.100000   45.950000
..         ...         ...  ...        ...     ...  ...        ...         ...
671  15.011897  215.000000  4.4   8.760000   45.95  4.0  15.100000  214.950000
672  14.769216  209.000000  4.4  21.322000   45.95  4.0  12.933333  215.000000
673  13.993833  205.000000  4.4  22.195932   45.95  4.0  14.840000  209.000000
674  14.656757  199.509804  4.4  19.412885   45.95  4.0  14.287500  205.000000
675  18.776522  163.398710  4.4  24.324687   45.95  4.0  15.100000  199.509804

[676 rows x 30 columns]>
```

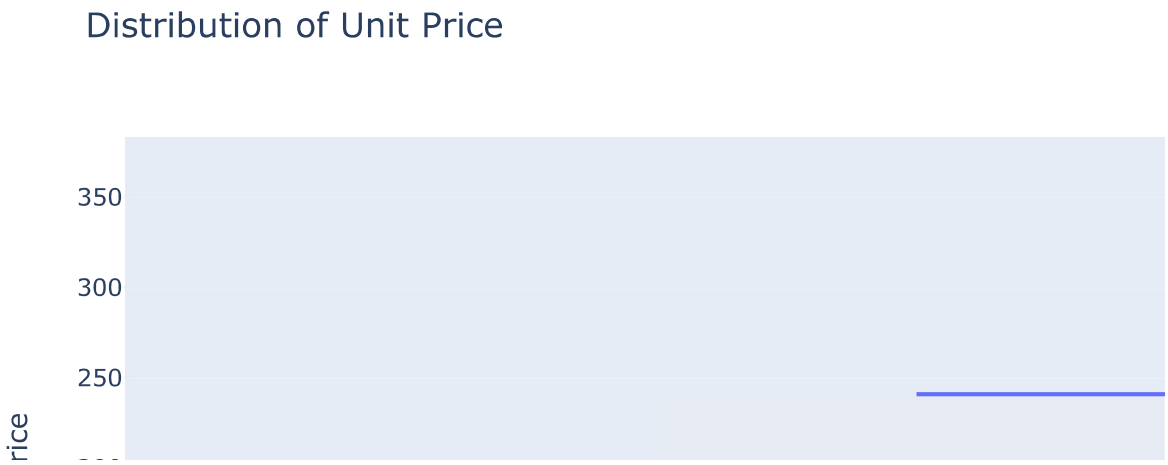
In [10]:
```
any(df.isna().sum() > 0)
```

Out[10]:
```
False
```

In [11]:
```
fig = px.histogram(df,
                   x='total_price',
                   nbins=20,
                   title='Distribution of Total Price')
fig.show()
```

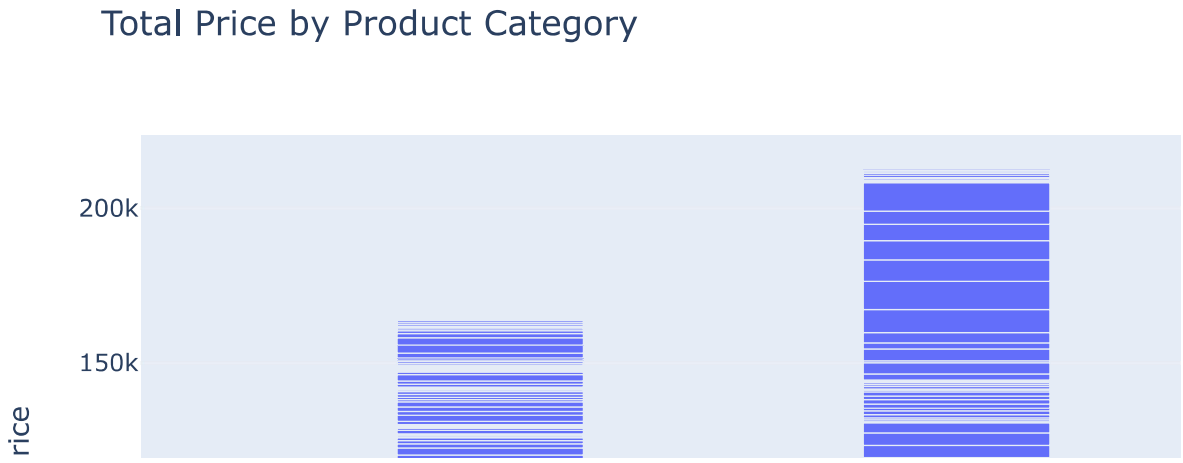


In [12]:
```
fig = px.box(df,
             y='unit_price',
             title='Distribution of Unit Price')
fig.show()
```

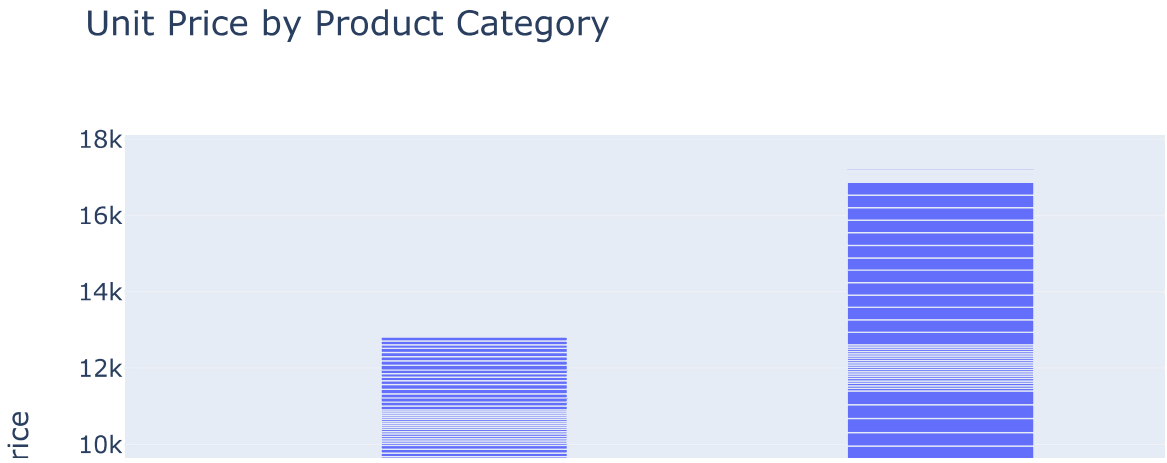


In [13]:
```
fig = px.scatter(df,
                 x='qty',
                 y='total_price', trendline='ols',
                 title='Quantity vs Total Price')
fig.show()
```

In [14]:
```
fig = px.bar(df, x='product_category_name',
             y='total_price', title='Total Price by Product Category')
fig.show()
```



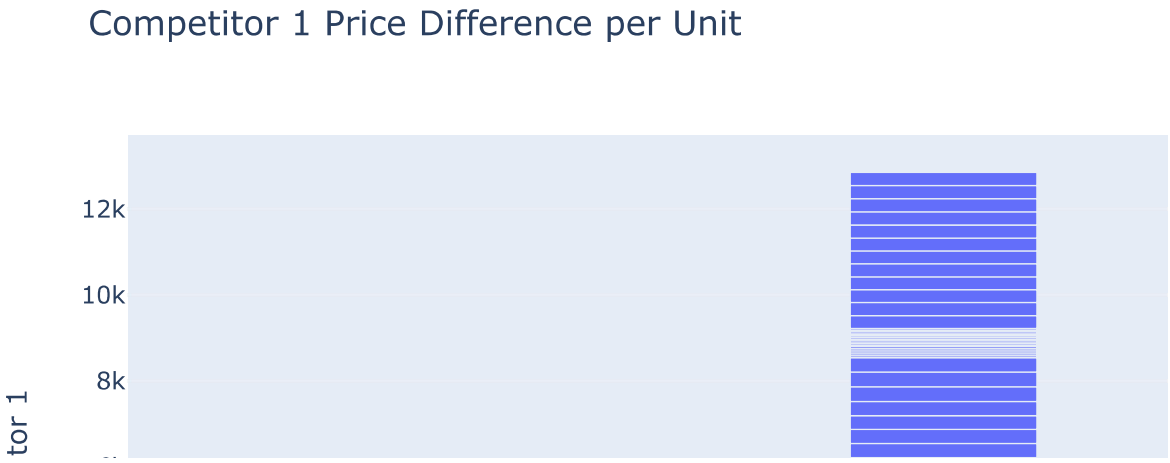
In [15]:
```
fig = px.bar(df, x='product_category_name',
             y='unit_price', title='Unit Price by Product Category')
fig.show()
```

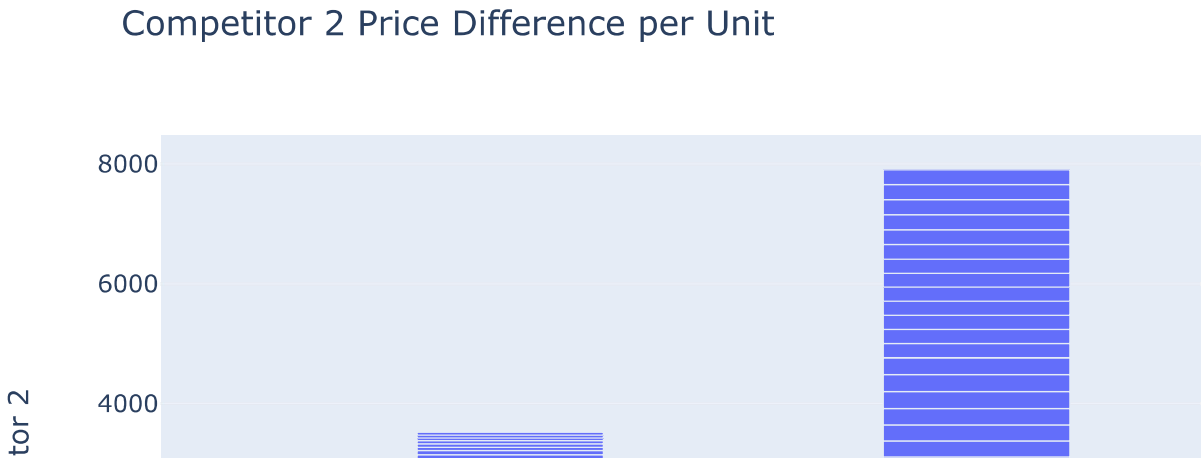


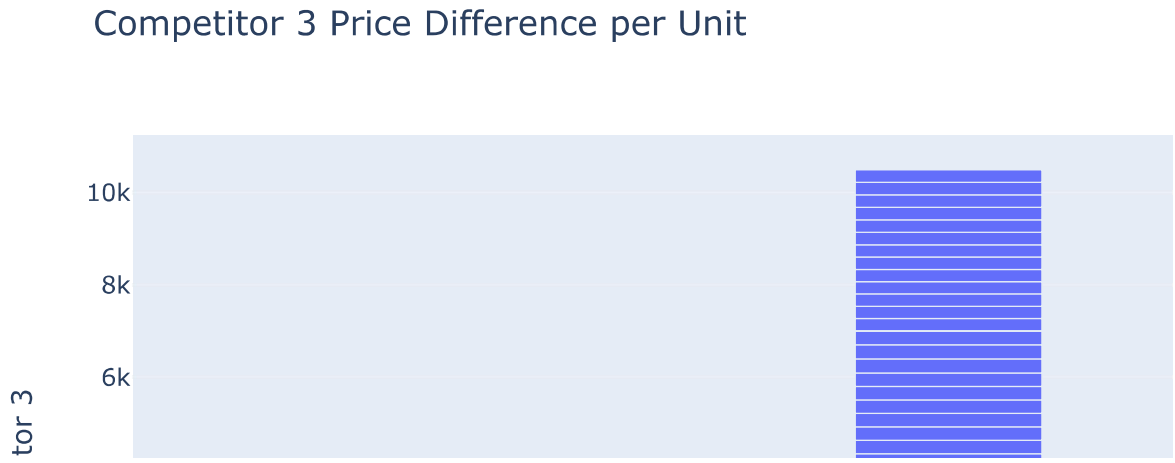
In [16]:
```
fig = px.box(df, x='weekday',
             y='total_price',
             title='Box Plot of Total Price by number of Weekdays in a Month')
fig.show()
```

In [17]:
```
fig = px.box(df, x='weekend',
             y='total_price',
             title='Box Plot of Total Price by number of Weekend days in a Month')
fig.show()
```

In [18]:
```
df['comp1_diff'] = df['unit_price'] - df['comp_1']
df['comp2_diff'] = df['unit_price'] - df['comp_2']
df['comp3_diff'] = df['unit_price'] - df['comp_3']

for i in range(1,4):
    comp = f"comp{i}_diff"
    fig = px.bar(x=df['product_category_name'],
                 y=df[comp],
                 title=f"Competitor {i} Price Difference per Unit",
                 labels={
                     'x': 'Product Category',
                     'y': f'Competitor {i}'
                 })
    fig.show()
```

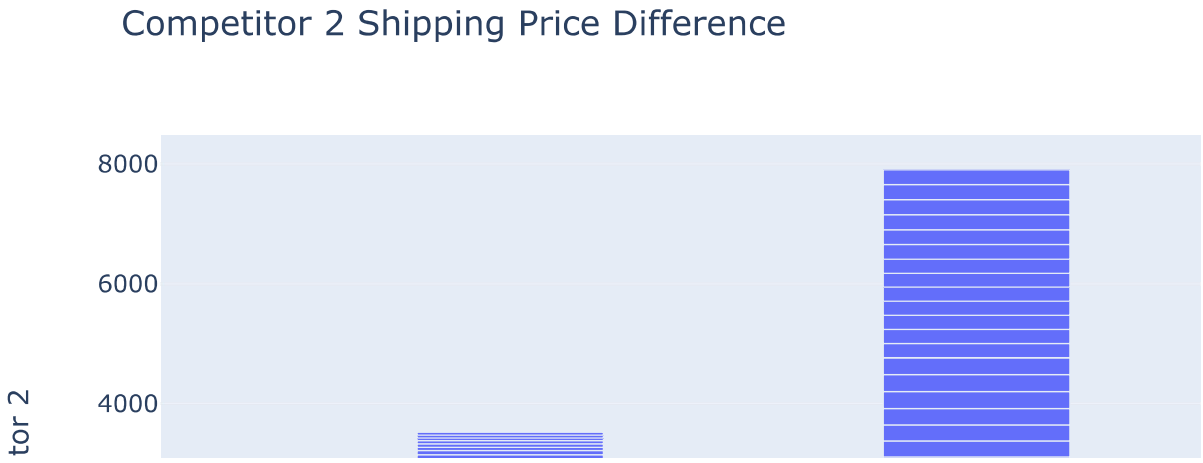

Competitor 1 Price Difference per Unit
12k
10k
8k
tor 1


Competitor 2 Price Difference per Unit
8000
6000
4000
tor 2



In [19]:
```
df['fp1_diff'] = df['freight_price'] - df['fp1']
df['fp2_diff'] = df['freight_price'] - df['fp2']
df['fp3_diff'] = df['freight_price'] - df['fp3']

for i in range(1,4):
    comp = f"comp{i}_diff"
    fig = px.bar(x=df['product_category_name'],
                 y=df[comp],
                 title=f"Competitor {i} Shipping Price Difference",
                 labels={
                     'x': 'Product Category',
                     'y': f'Competitor {i}'
                 })
    fig.show()
```


Competitor 1 Shipping Price Difference
12k
10k
8k
tor 1


Competitor 2 Shipping Price Difference
8000
6000
4000
tor 2

In [20]:
```
# Filter out non-numeric columns
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns

# Calculate correlation
corrs = df[numeric_columns].corr()['unit_price'].sort_values(ascending=False)

# Plot
fig = px.bar(x=corrs.index, y=corrs.values,
             title='Correlation of Features with Unit Price',
             labels={'x': 'Features', 'y': 'Correlation'})

fig.show()
```


Correlation of Features with Unit Price
1
0.8
0.6
0.4
rrelation

In [ ]:

In [ ]: