In [1]:
```
#importing libraries
import numpy as np
import pandas as pd

import plotly.express as px
import plotly.graph_objs as go

from datetime import date
```

In [2]:
```
data = pd.read_csv("kz.csv")
```

About this file

This file contains purchase data from April 2020 to November 2020 from a large home appliances and electronics online store, collected by Open CDP project.

In [3]:
```
data.head(10)
```

Out[3]:

	event_time	order_id	product_id	category_id	category_code	
0	2020-04-24 11:50:39 UTC	2294359932054536986	1515966223509089906	2.268105e+18	electronics.tablet	sa
1	2020-04-24 11:50:39 UTC	2294359932054536986	1515966223509089906	2.268105e+18	electronics.tablet	sa
2	2020-04-24 14:37:43 UTC	2294444024058086220	2273948319057183658	2.268105e+18	electronics.audio.headphone	h
3	2020-04-24 14:37:43 UTC	2294444024058086220	2273948319057183658	2.268105e+18	electronics.audio.headphone	h
4	2020-04-24 19:16:21 UTC	2294584263154074236	2273948316817424439	2.268105e+18	NaN	k
5	2020-04-26 08:45:57 UTC	2295716521449619559	1515966223509261697	2.268105e+18	furniture.kitchen.table	m
6	2020-04-26 09:33:47 UTC	2295740594749702229	1515966223509104892	2.268105e+18	electronics.smartphone	
7	2020-04-26 09:33:47 UTC	2295740594749702229	1515966223509104892	2.268105e+18	electronics.smartphone	
8	2020-04-26 09:33:47 UTC	2295740594749702229	1515966223509104892	2.268105e+18	electronics.smartphone	
9	2020-04-26 09:33:47 UTC	2295740594749702229	1515966223509104892	2.268105e+18	electronics.smartphone	

In [4]:
```
data.dtypes
```

Out[4]:
```
event_time        object
order_id           int64
product_id         int64
category_id      float64
category_code     object
brand             object
price            float64
user_id          float64
dtype: object
```

Cleaning the Data

In [5]:
```
data.set_index('order_id', inplace = True)
```

In [6]:
```
null_columns = data.columns[data.isnull().any()]
data[null_columns].isnull().sum()
```

Out[6]:
```
category_id       431954
category_code     612202
brand             506005
price             431954
user_id          2069352
dtype: int64
```

In [7]:
```
n_unique_products = data['product_id'].nunique()
n_unique_users = data['user_id'].nunique()
print('Number of unique users: ' + str(n_unique_users) + '. Number of unique products i
```

```
Number of unique users: 98262. Number of unique products is: 25113
```

In [8]:
```
data['event_time'] = pd.to_datetime(data['event_time'])
```

In [10]:
```
data.dropna(subset = ['category_code'], inplace = True)
```

In [11]:
```
# Getting the main category from the category_code
data['category'] = data['category_code'].str.rsplit('.', n=1, expand = True)[1]
data.drop(columns = ['category_code'], inplace = True)
```

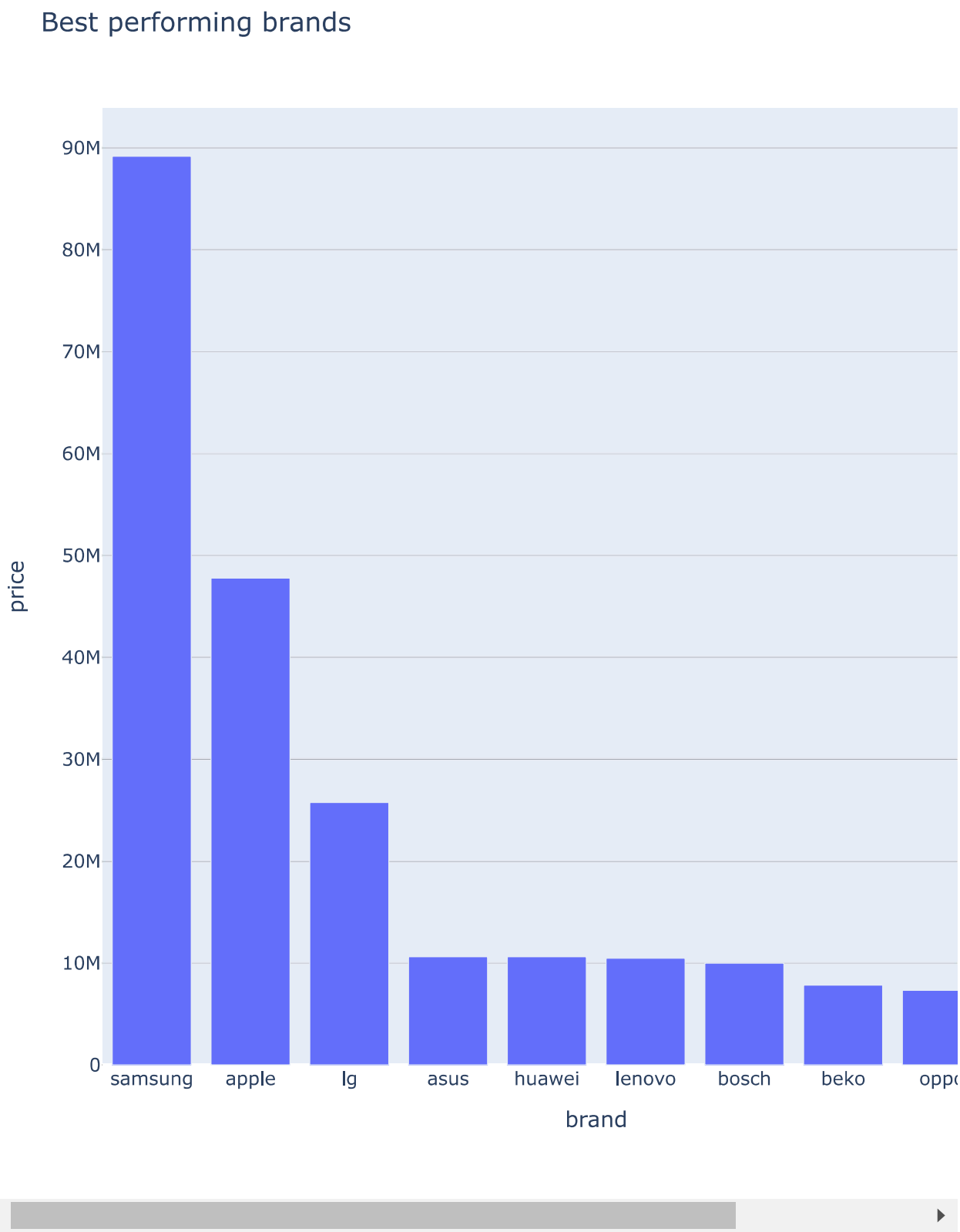
Analyzing data

Best performing brands

In [12]:
```
# best performing brands

best_performing_brands = data.groupby('brand')['price'].sum().reset_index().sort_values
```

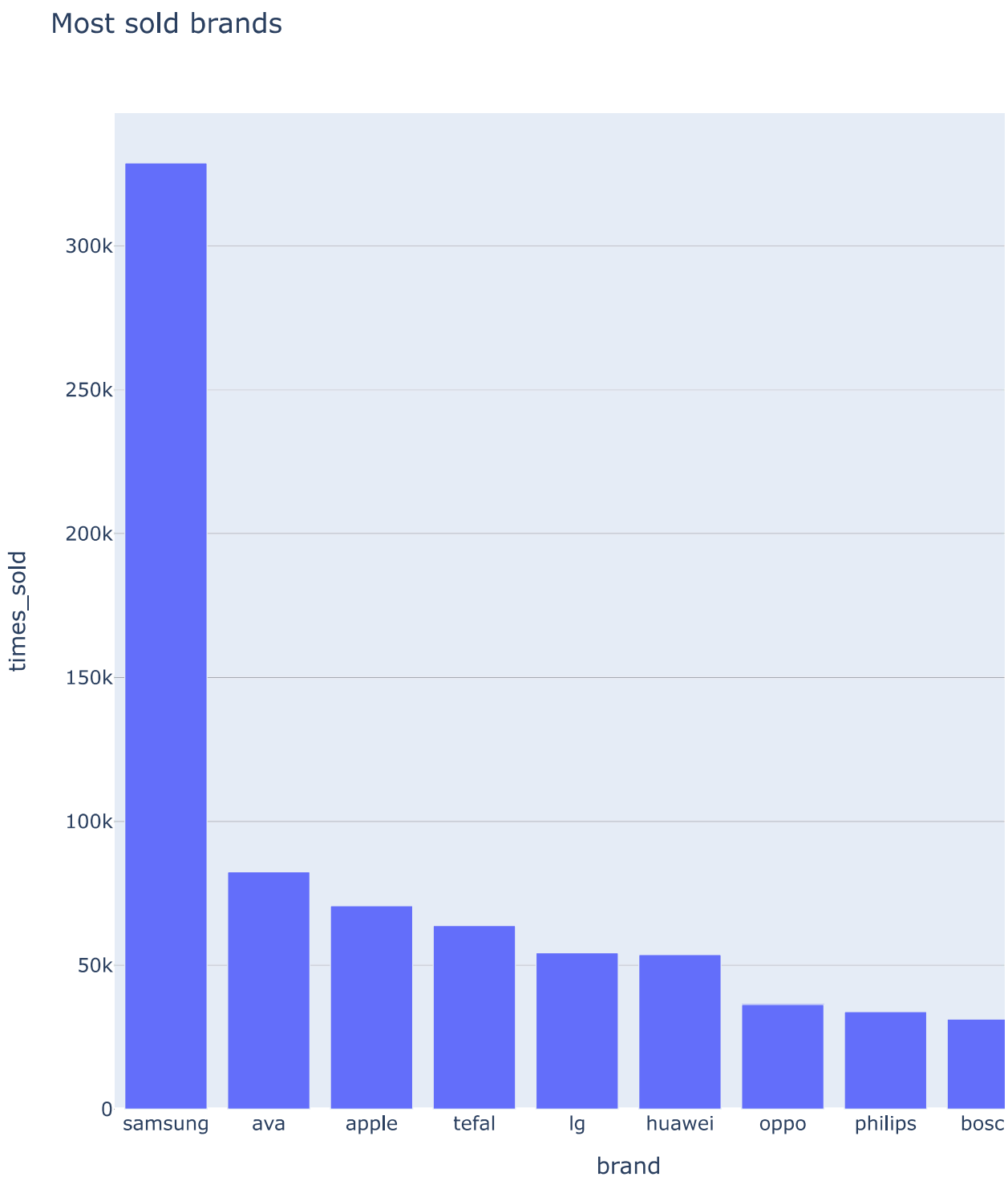
In [13]:
```
fig = px.bar(
    best_performing_brands,
    x = 'brand',
    y = 'price',
    title = 'Best performing brands',
    width = 800,
    height = 800
    )
```

```
fig.show()
```



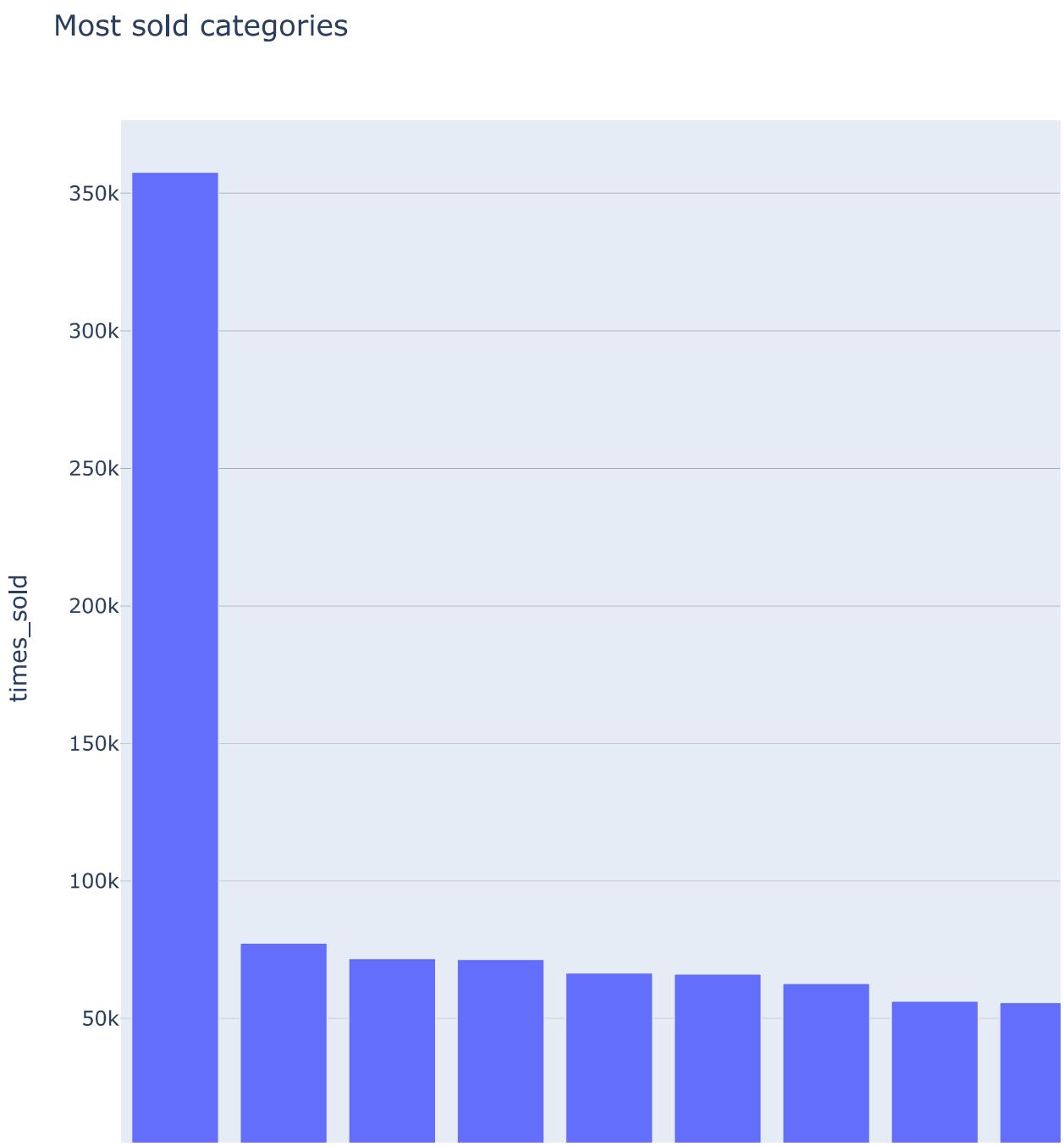
In [14]:
```
#most sold brands
most_sold_brands = data.groupby('brand')['price'].agg('count').reset_index().sort_value
most_sold_brands.rename(columns = {"brand":"brand", "price": "times_sold"}, inplace = T
```

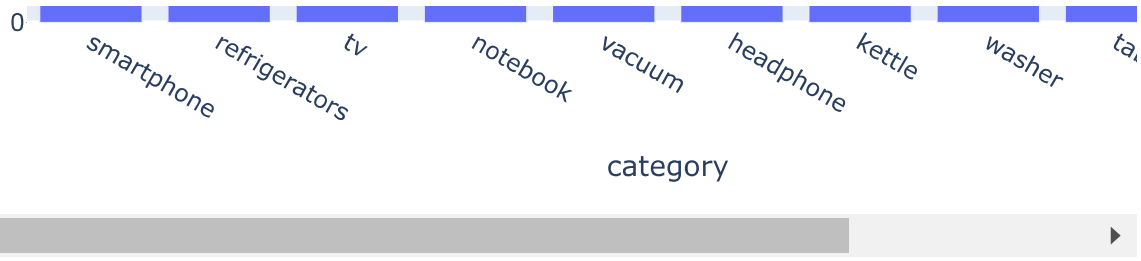
In [16]:
```
fig = px.bar(
    most_sold_brands,
    x = 'brand',
    y = 'times_sold',
    title = 'Most sold brands',
    width = 800,
    height = 800
)

fig.show()
```



In [18]:
```
#most bought categories
most_sold_categories = data.groupby('category')['price'].agg('count').reset_index().sor
most_sold_categories.rename(columns = {"category": 'category', "price": "times_sold"},
```

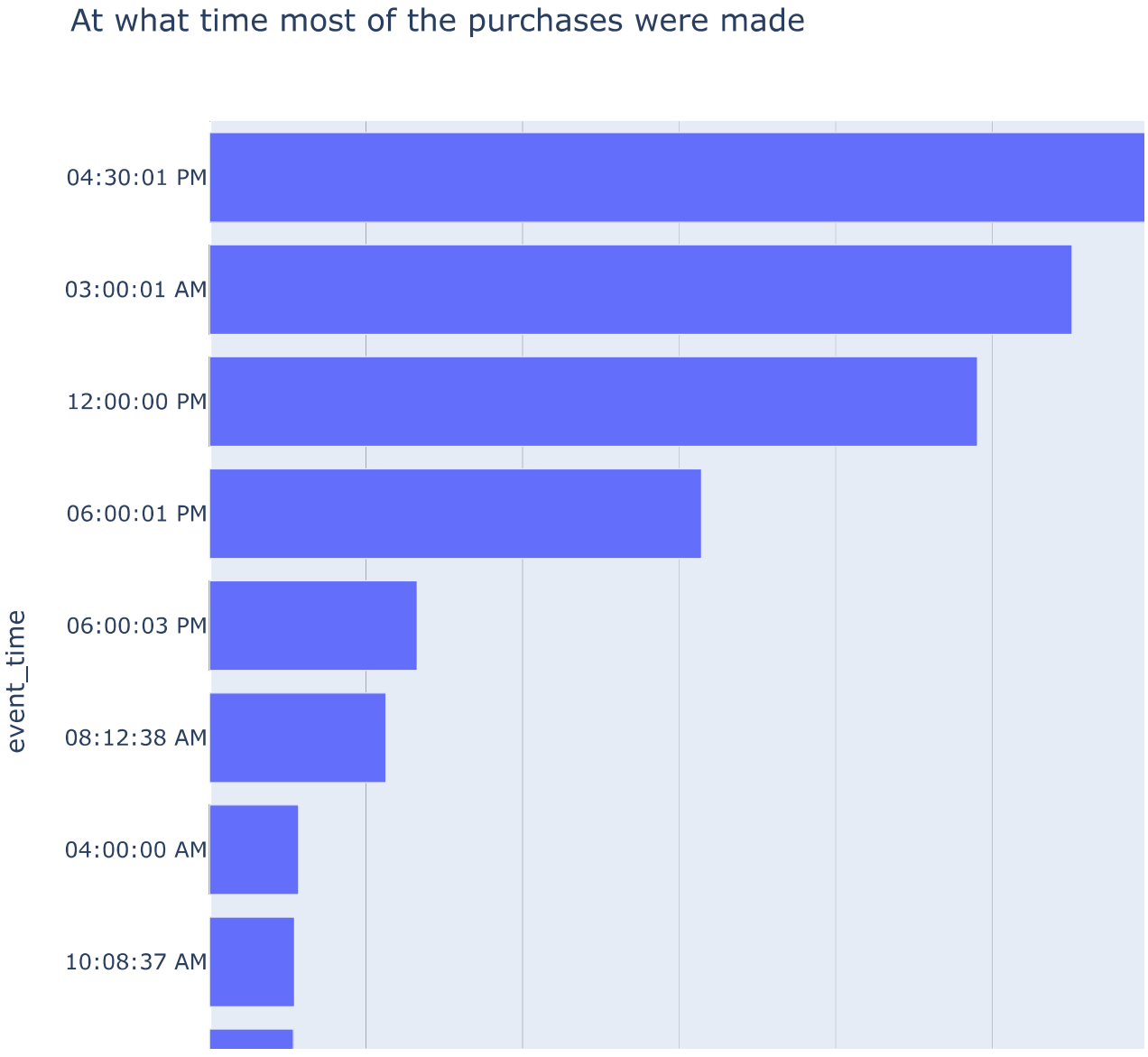
In [21]:
```
fig = px.bar(
    most_sold_categories,
    x = 'category',
    y = 'times_sold',
    title = 'Most sold categories',
    width = 800,
    height = 800
)

fig.show()
```

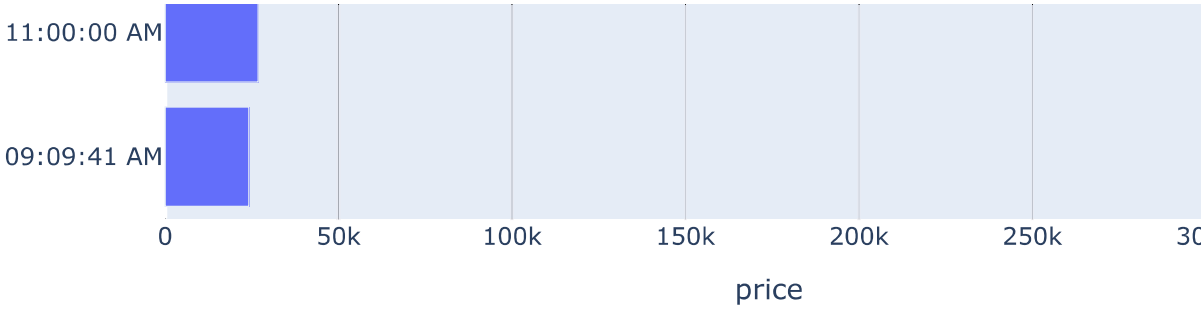




In [22]:
```
#best time for purchase. filter with lambda <900 as alot of data was on 1970-1-1 12 an
best_time = data.groupby(data['event_time'].dt.strftime('%r'))['price'].sum().sort_valu
```

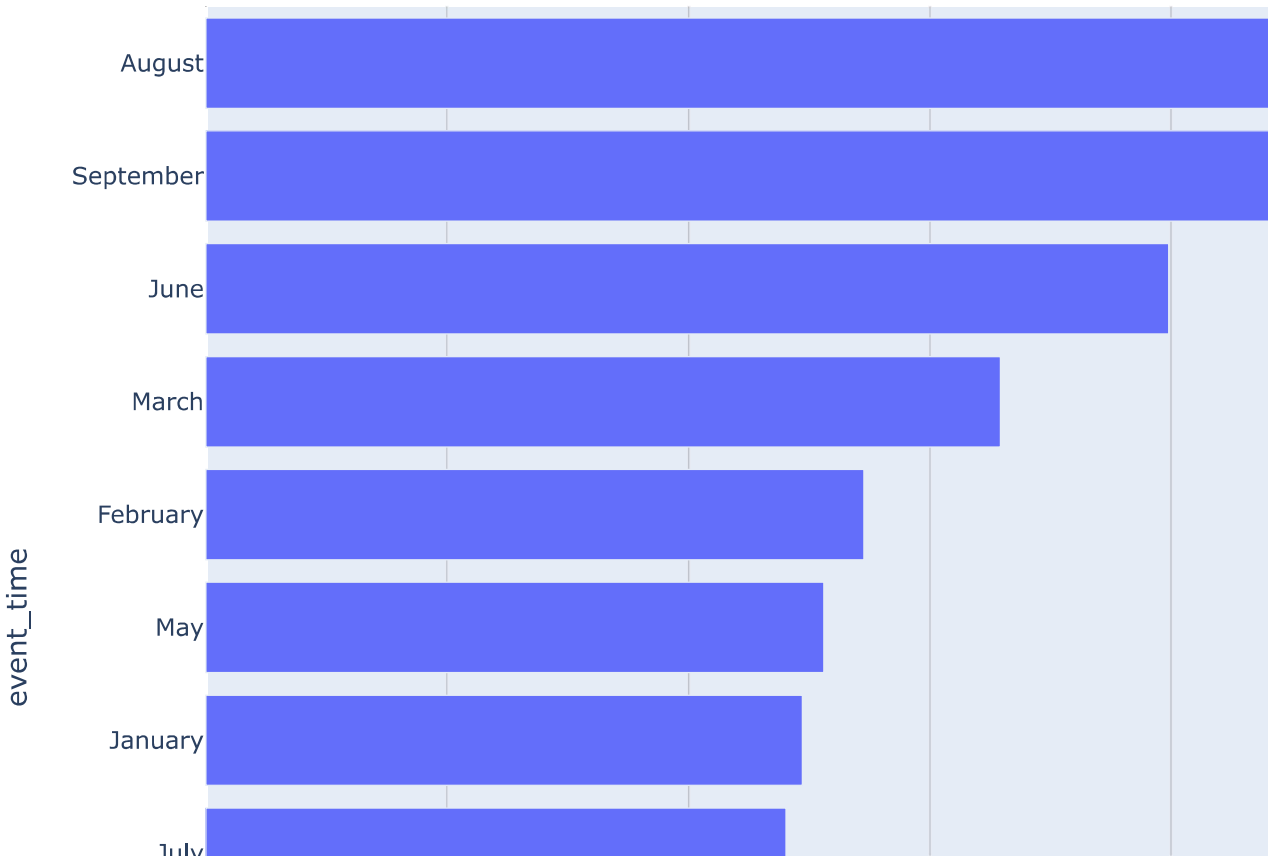
In [24]:
```
fig = px.bar(
    best_time,
    x = "price",
    orientation = 'h',
    title = "At what time most of the purchases were made",
    width = 800,
    height = 800
)

fig.show()
```

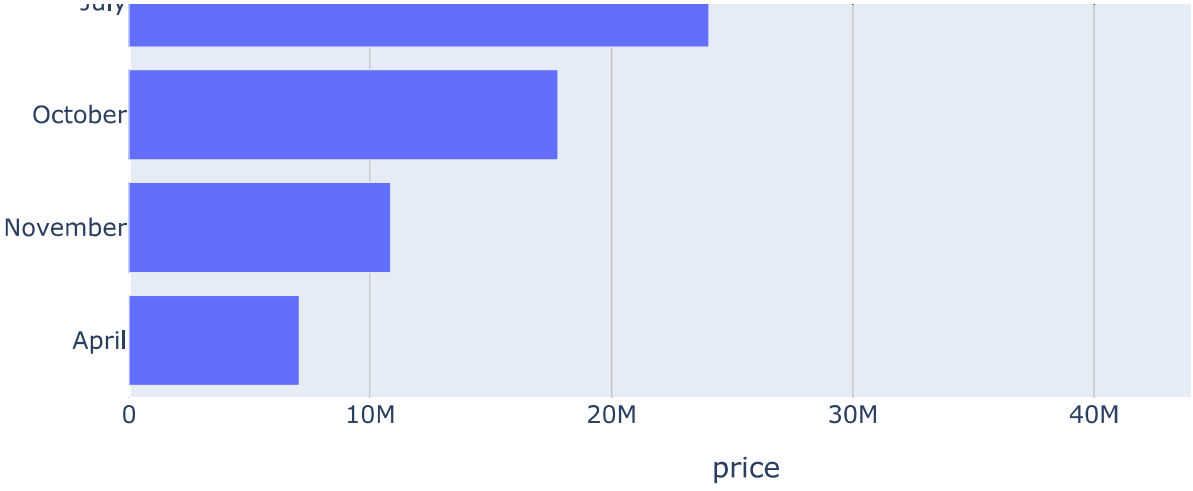




In [25]:
```
#monthly purchases
best_month = data.groupby(data['event_time'].dt.strftime('%B'))['price'].sum().sort_val
```

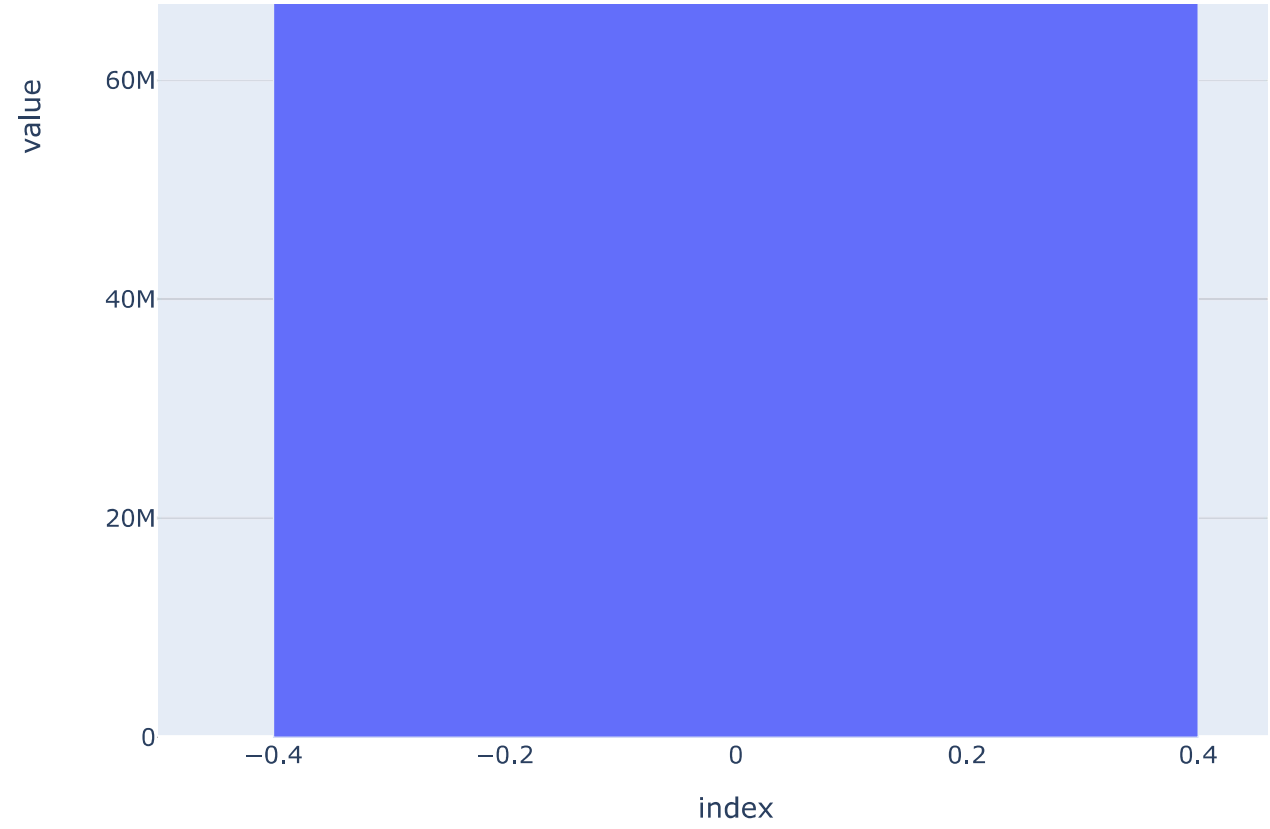
In [26]:
```
fig = px.bar(
    best_month,
    x = "price",
    orientation = 'h',
    title = "In which months most of the purchases were made",
    width = 800,
    height = 800
)

fig.show()
```





In [28]:
```
# How much money spent 20% of top buyers in comparison with other 80% clients

most_active_users = data.groupby('user_id')['price'].sum().reset_index().sort_values('p

least_active_users = data.groupby('user_id')['price'].sum().reset_index().sort_values('

top_20_percent_buyers = most_active_users['price'].sum()
bottom_80_percent_buyers = least_active_users['price'].sum()
last_data = pd.DataFrame(data = {'most_active': [most_active_users['price'].sum()], 'le
```

In [29]:
```
fig = px.bar(
    last_data,
    title = "20% of most active in comparison with other 80% of buyers",
    width = 800,
    height = 800
)

fig.show()
```



In [ ]: