In [105]:
```
import pandas as pd
import numpy as np

import plotly.graph_objects as go
import colorlover

colors = ['#F1948A', '#AED6F1','#F9E79F','#E5E8E8', '#F1948A', '#D0ECE7','#F6DDCC', '#D2B4DE',
        '#117A65','#FAE5D3','#34495E','#DC7633','#D35400','#0E6251','#FCF3CF','#E8F8F5','#D4E6F1','#FADBD8','#E59866']
```

In [106]:
```
f = "BlackFriday.csv"
data = pd.read_csv(f)
data.head()
```

Out[106]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	NaN	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	14.0	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	NaN	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	NaN	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	NaN	7969

In [107]:
```
data.shape #显示数据大小
```

Out[107]:
```
(537577, 12)
```

In [108]:
```
data.Occupation.unique()#unique将数据中职业类型全部归类显示出来
```

Out[108]:
```
array([10, 16, 15,  7, 20,  9,  1, 12, 17,  0,  3,  4, 11,  8, 19,  2, 18,
        5, 14, 13,  6], dtype=int64)
```

In [109]:
```
len(data.User_ID.unique())#统计人数
```

Out[109]:
```
5891
```

In [110]:
```
data.describe()
```

Out[110]:

	User_ID	Occupation	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
count	5.375770e+05	537577.00000	537577.000000	537577.000000	370591.000000	164278.000000	537577.000000
mean	1.002992e+06	8.08271	0.408797	5.295546	9.842144	12.669840	9333.859853
std	1.714393e+03	6.52412	0.491612	3.750701	5.087259	4.124341	4981.022133
min	1.000001e+06	0.00000	0.000000	1.000000	2.000000	3.000000	185.000000
25%	1.001495e+06	2.00000	0.000000	1.000000	5.000000	9.000000	5866.000000
50%	1.003031e+06	7.00000	0.000000	5.000000	9.000000	14.000000	8062.000000
75%	1.004417e+06	14.00000	1.000000	8.000000	15.000000	16.000000	12073.000000
max	1.006040e+06	20.00000	1.000000	18.000000	18.000000	18.000000	23961.000000

In [111]:
```
data.isna().sum()#缺失值查看
```

Out[111]:
```
User_ID                         0
Product_ID                      0
Gender                          0
Age                             0
Occupation                      0
City_Category                   0
Stay_In_Current_City_Years      0
Marital_Status                  0
Product_Category_1              0
Product_Category_2         166986
Product_Category_3         373299
Purchase                        0
dtype: int64
```

In [112]:
```
#pivot_table 是Pandas的高级应用中的透视表的功能
gender_purchase = data.pivot_table(values="Purchase",index=["User_ID"])
gender_purchase.head()
```

Out[112]:

User_ID	Purchase
1000001	9808.264706
1000002	10662.539474
1000003	11780.517241
1000004	15845.153846
1000005	7745.292453

In [113]:
```
#aggfunc指定一个函数操作，对values求和但是按照index分类
gender_purchase = data.pivot_table(values="Purchase",aggfunc="sum",index=["User_ID"])
gender_purchase.head()
```

Out[113]:

User_ID	Purchase
1000001	333481
1000002	810353
1000003	341635
1000004	205987
1000005	821001

In [114]:
```
#index为性别
gender_purchase = data.pivot_table(values="Purchase",
                                   aggfunc="sum",index=["Gender"])
gender_purchase.head()
```

Out[114]:

Gender	Purchase
F	1164624021
M	3853044357

In [115]:
```
#index添加显示性别信息
gender_purchase = data.pivot_table(values="Purchase",
                                   aggfunc="sum",
                                   index=["User_ID","Gender"])
gender_purchase.head()
```

Out[115]:

		Purchase
User_ID	**Gender**	
1000001	F	333481
1000002	M	810353
1000003	M	341635
1000004	M	205987
1000005	M	821001

上面的index不正常，需要用reset_index()更正

In [116]:
```
#index添加显示性别信息
gender_purchase = data.pivot_table(values="Purchase",
                                   aggfunc="sum",
                                   index=["User_ID","Gender"]).reset_index()
                                    #reset_index 为了变为正常的index
gender_purchase.head()
```

Out[116]:

	User_ID	Gender	Purchase
0	1000001	F	333481
1	1000002	M	810353
2	1000003	M	341635
3	1000004	M	205987
4	1000005	M	821001

In [117]:
```
gender_purchase.count() #count信息个数
```

Out[117]:
```
User_ID     5891
Gender      5891
Purchase    5891
dtype: int64
```

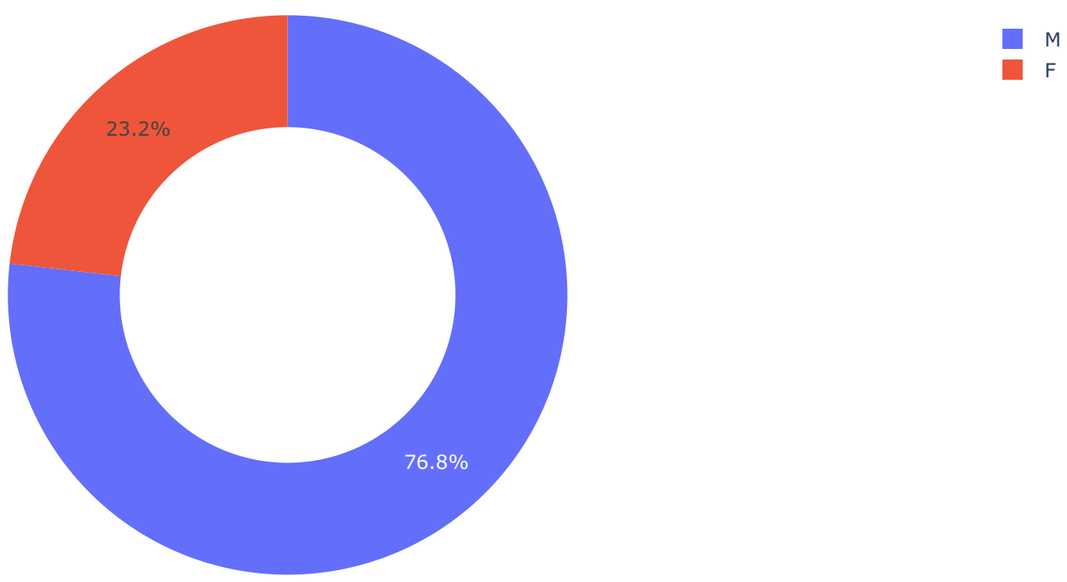
In [118]:
```
gender_count = gender_purchase.groupby(by="Gender").size().reset_index(name="人数")
print(gender_purchase.groupby(by="Gender").size())
gender_count["占比"] = gender_count["人数"] / gender_count["人数"].sum()
gender_count
```

```
Gender
F    1666
M    4225
dtype: int64
```

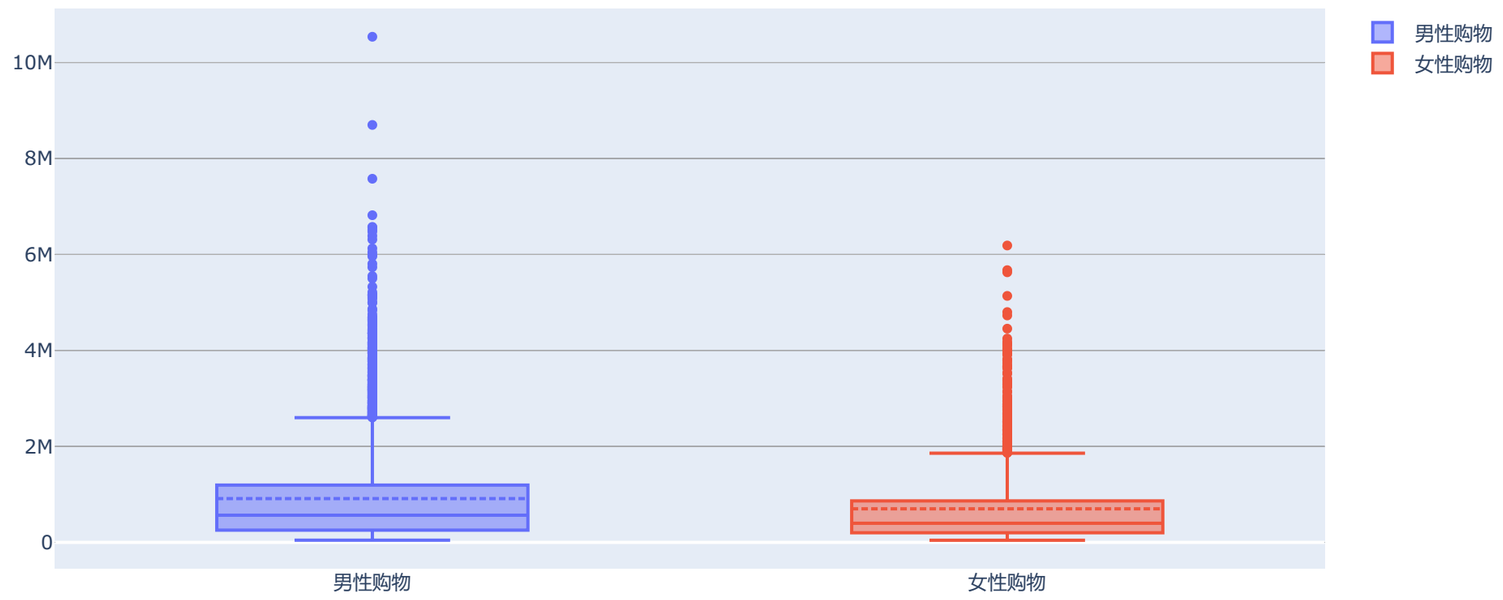
Out[118]:

	Gender	人数	占比
0	F	1666	0.282804
1	M	4225	0.717196

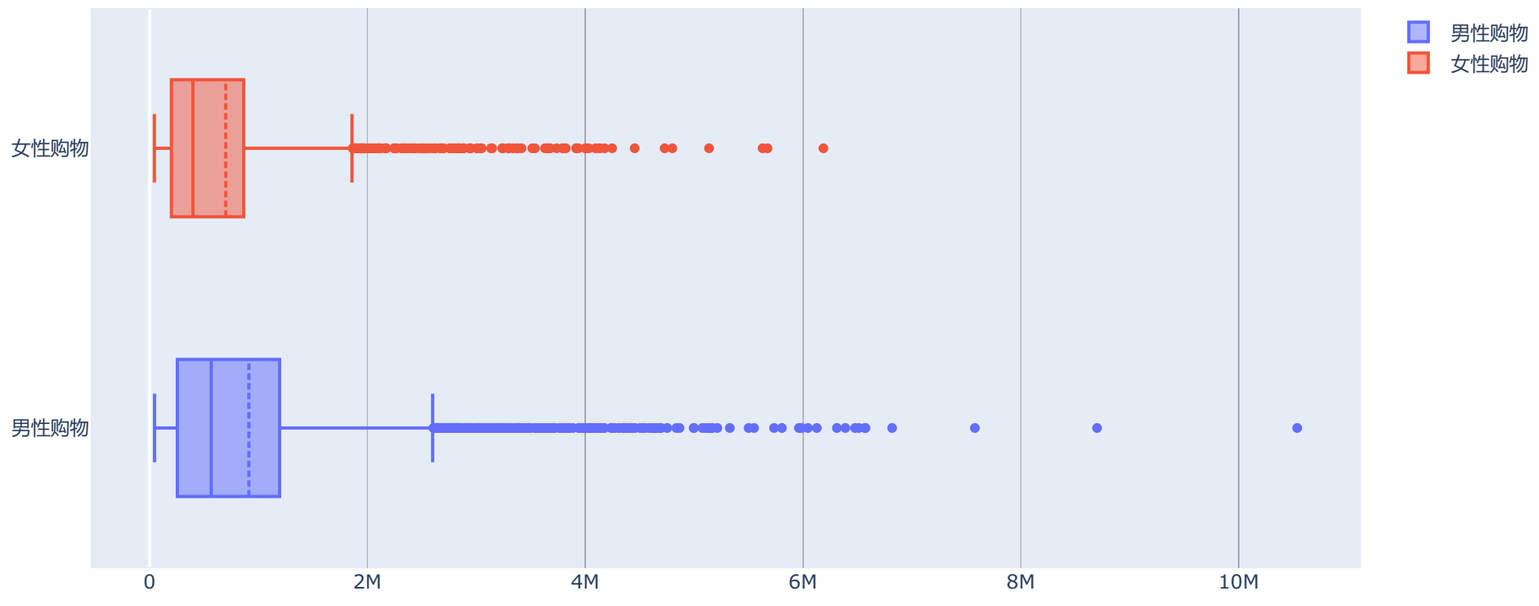
In [119]:
```
# 描绘一下数据
trace =  go.Pie(labels=gender_purchase.Gender.tolist(),
                values=gender_purchase.Purchase.tolist(),
                hole=0.6)
fig = go.Figure(data=[trace])
fig.show()
```



In [120]:
```
y_female = gender_purchase[gender_purchase.Gender == "F"].Purchase
y_male = gender_purchase[gender_purchase.Gender == "M"].Purchase
trace1 = go.Box(y=y_male,name="男性购物",boxmean=True) #显示平均值（虚线）
trace2 = go.Box(y=y_female,name="女性购物",boxmean=True)
fig = go.Figure(data=[trace1,trace2])
fig.show()
```



In [121]:
```
#横着画
y_female = gender_purchase[gender_purchase.Gender == "F"].Purchase
y_male = gender_purchase[gender_purchase.Gender == "M"].Purchase
trace1 = go.Box(x=y_male,name="男性购物",boxmean=True) #显示平均值（虚线）
trace2 = go.Box(x=y_female,name="女性购物",boxmean=True)
fig = go.Figure(data=[trace1,trace2])
fig.show()
```



In [122]:
```
top10_sellers = data.pivot_table(values=["Purchase"],
                                 index=['Product_ID'],
                                 aggfunc="sum"
                                 ).reset_index().sort_values(by="Purchase",
                                                             ascending=False).head(10)
top10_sellers
```

Out[122]:

	Product_ID	Purchase
249	P00025442	27532426
1014	P00110742	26382569
2441	P00255842	24652442
1743	P00184942	24060871
581	P00059442	23948299
1028	P00112142	23882624
1016	P00110942	23232538
2261	P00237542	23096487
565	P00057642	22493690
104	P00010742	21865042

In [123]:
```
#都哪些人购买了这些热销商品
top_seller_buyers = data[data.Product_ID.isin(top10_sellers.Product_ID.tolist())]
top_seller_buyers.head(10)
```

Out[123]:

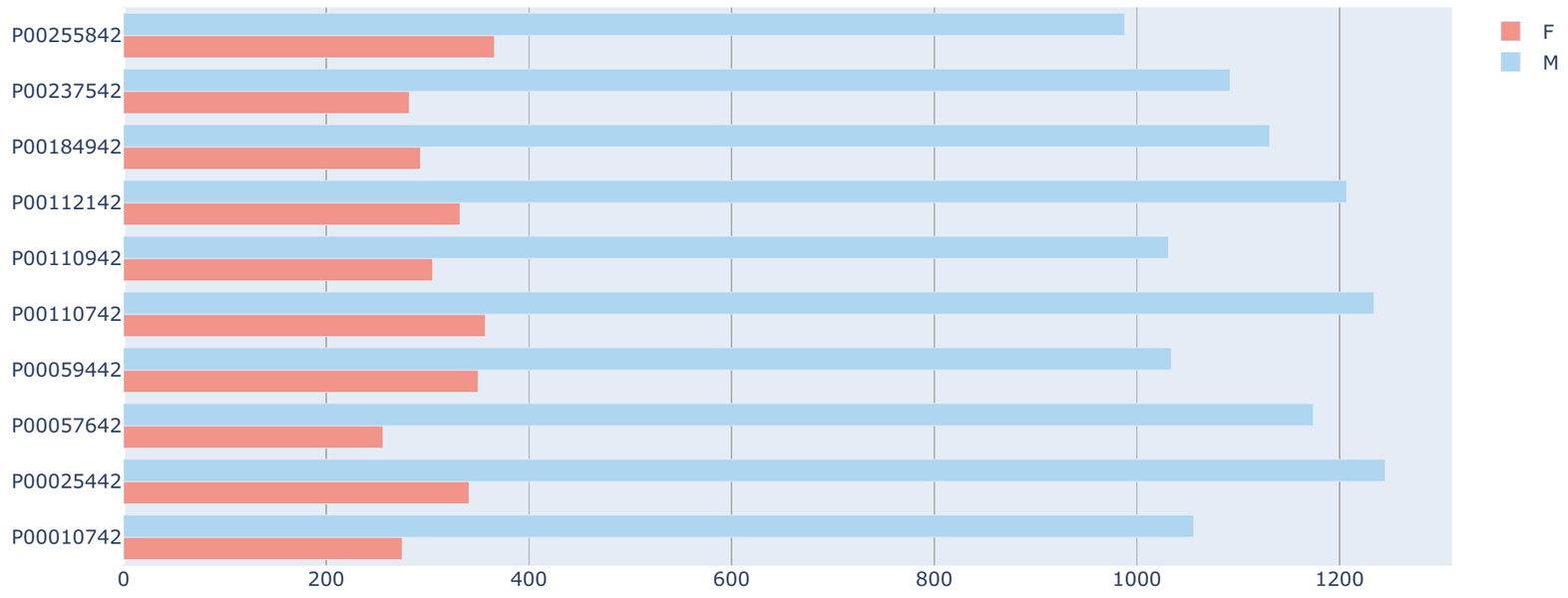
	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
6	1000004	P00184942	M	46-50	7	B	2	1	1	8.0	17.0	19215
234	1000043	P00255842	M	26-35	12	A	0	0	16	NaN	NaN	20961
235	1000044	P00112142	M	46-50	17	B	3	1	1	2.0	14.0	19072
266	1000048	P00010742	M	26-35	4	B	3	1	1	8.0	17.0	19352
336	1000056	P00237542	M	36-45	20	C	0	1	1	15.0	16.0	11370
342	1000058	P00110742	M	26-35	2	B	3	0	1	2.0	8.0	15824
369	1000059	P00010742	F	51-55	1	B	4+	1	1	8.0	17.0	7988
392	1000064	P00110942	M	18-25	1	A	1	1	1	2.0	NaN	19462
406	1000069	P00184942	F	26-35	1	A	1	0	1	8.0	17.0	11715
506	1000092	P00255842	F	18-25	4	B	1	0	16	NaN	NaN	16416

In [124]:
```
top_seller_gender = top_seller_buyers.pivot_table(values="Purchase",
                                                  index=["Product_ID","Gender"],
                                                  aggfunc="count").reset_index()
top_seller_gender.head(10)
```

Out[124]:

	Product_ID	Gender	Purchase
0	P00010742	F	275
1	P00010742	M	1056
2	P00025442	F	341
3	P00025442	M	1245
4	P00057642	F	256
5	P00057642	M	1174
6	P00059442	F	350
7	P00059442	M	1034
8	P00110742	F	357
9	P00110742	M	1234

In [125]:
```
traces = []
i = 0
for g in top_seller_gender.Gender.unique():
        trace = go.Bar(x=top_seller_gender[top_seller_gender.Gender==g].Purchase,
                       y=top_seller_gender[top_seller_gender.Gender==g].Product_ID,
                       name = g,
                       marker=dict(color=colors[i]),
                       orientation="h")
        traces.append(trace)
        i += 1
fig = go.Figure(data=traces)
fig.show()
```

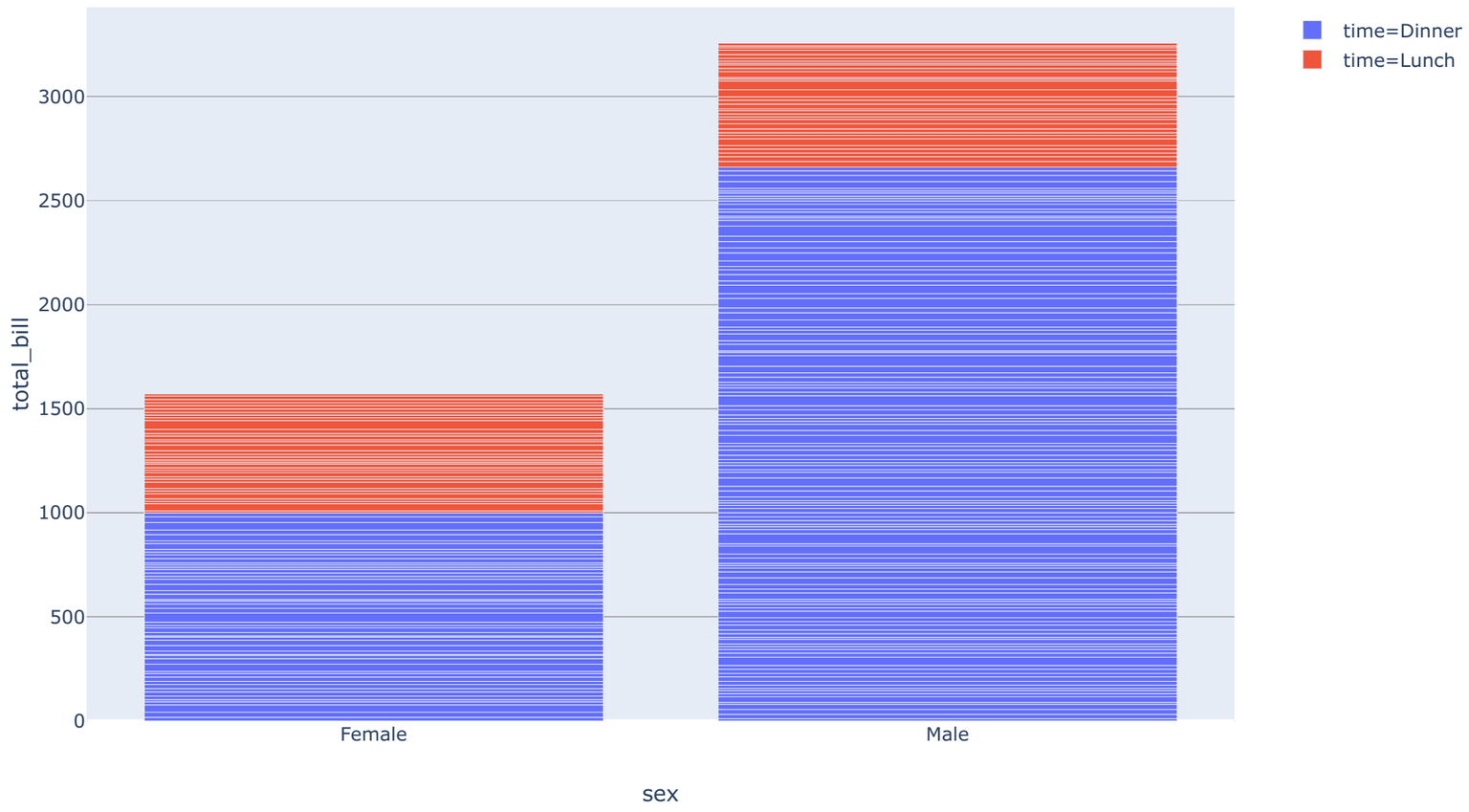


In [126]:
```
top_seller_city = top_seller_buyers.pivot_table(values="Purchase",
                                                index=["Product_ID","City_Category"],
                                                aggfunc="count").reset_index()
top_seller_city.head(10)
```

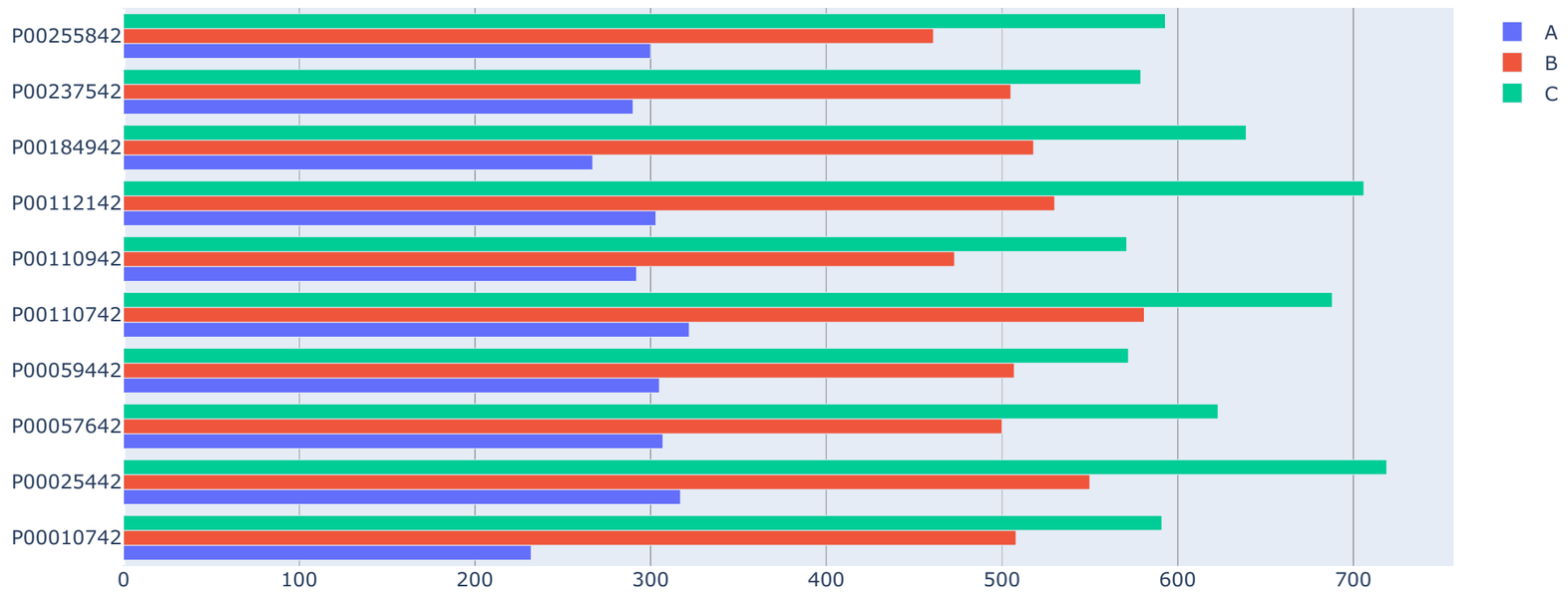
Out[126]:

	Product_ID	City_Category	Purchase
0	P00010742	A	232
1	P00010742	B	508
2	P00010742	C	591
3	P00025442	A	317
4	P00025442	B	550
5	P00025442	C	719
6	P00057642	A	307
7	P00057642	B	500
8	P00057642	C	623
9	P00059442	A	305

In [127]:
```
import plotly.express as px
tips = px.data.tips()
fig = px.bar(tips,x="sex",y="total_bill",color="time")
fig.show()
```



In [128]:
```
traces = []
i = 0
for c in top_seller_city.City_Category.unique():
    trace  = go.Bar(x=top_seller_city[top_seller_city.City_Category == c].Purchase,
                    y=top_seller_city[top_seller_city.City_Category == c].Product_ID,
                    name = c,
                    orientation = "h"
                    )
    traces.append(trace)
    i +=1
go .Figure(data=traces).show()
```



pyecharts 更简易，更多查看Github