In [22]:
```
import numpy as np
import pandas as pd
import ast
import plotly.express as px
from plotly import graph_objects as go
```

In [23]:
```
df = pd.read_csv("flipkart_com-ecommerce_sample.csv")
```

In [24]:
```
df.head()
```

Out[24]:

	uniq_id	crawl_timestamp	product_url	prod
0	c2d766ca982eca8304150849735ffef9	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	A Cycl
1	7f7036a6d550aaa89d34c77bd39a5e48	2016-03-25 22:59:23 +0000	http://www.flipkart.com/fabhomedecor-fabric-do...	FabH Fab
2	f449ec65dcbc041b6ae5e6a32717d01b	2016-03-25 22:59:23 +0000	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	/
3	0973b37acd0c664e3de26e97e5571454	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	A Cycl
4	bc940ea42ee6bef5ac7cea3fb5cfbee7	2016-03-25 22:59:23 +0000	http://www.flipkart.com/sicons-all-purpose-arn...	Purp Dog

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
uniq_id                       0
crawl_timestamp               0
product_url                   0
product_name                  0
product_category_tree         0
pid                           0
retail_price                 78
discounted_price             78
image                         3
is_FK_Advantage_product       0
description                   2
product_rating                0
overall_rating                0
brand                      5864
product_specifications       14
dtype: int64
```

In [26]:

```
df["retail_price"].fillna(df["retail_price"].median(),inplace=True)
df["discounted_price"].fillna(df["discounted_price"].median(),inplace=True)
```

In [27]:

```
df.head()
```

Out[27]:

	uniq_id	crawl_timestamp	product_url	prod
0	c2d766ca982eca8304150849735ffef9	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	A Cycl
1	7f7036a6d550aaa89d34c77bd39a5e48	2016-03-25 22:59:23 +0000	http://www.flipkart.com/fabhomedecor-fabric-do...	FabH Fab
2	f449ec65dcbc041b6ae5e6a32717d01b	2016-03-25 22:59:23 +0000	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	/
3	0973b37acd0c664e3de26e97e5571454	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	A Cycl
4	bc940ea42ee6bef5ac7cea3fb5cfbee7	2016-03-25 22:59:23 +0000	http://www.flipkart.com/sicons-all-purpose-arn...	Purp Dog

In [28]:

```
x=df['retail_price']-df['discounted_price']
y=(x/df['retail_price'])*100
df['discount_percentage']=y
```

In [29]:
```
df.head()
```

Out[29]:

	uniq_id	crawl_timestamp	product_url	prod
0	c2d766ca982eca8304150849735ffef9	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	A Cycl
1	7f7036a6d550aaa89d34c77bd39a5e48	2016-03-25 22:59:23 +0000	http://www.flipkart.com/fabhomedecor-fabric-do...	FabH Fab
2	f449ec65dcbc041b6ae5e6a32717d01b	2016-03-25 22:59:23 +0000	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	/
3	0973b37acd0c664e3de26e97e5571454	2016-03-25 22:59:23 +0000	http://www.flipkart.com/alisha-solid-women-s-c...	A Cycl
4	bc940ea42ee6bef5ac7cea3fb5cfbee7	2016-03-25 22:59:23 +0000	http://www.flipkart.com/sicons-all-purpose-arn...	Purp Dog

In [30]:
```
df['timestamp']=pd.to_datetime(df['crawl_timestamp'])
df['Time']=df['timestamp'].apply(lambda x : x.time)
df['date']=df['timestamp'].apply(lambda x : x.date)
df.drop(['crawl_timestamp'], axis = 1,inplace=True)
df['main_category']=df['product_category_tree'].apply(lambda x :x.split('>>')[0
```

In [31]:
```
df.head()
```

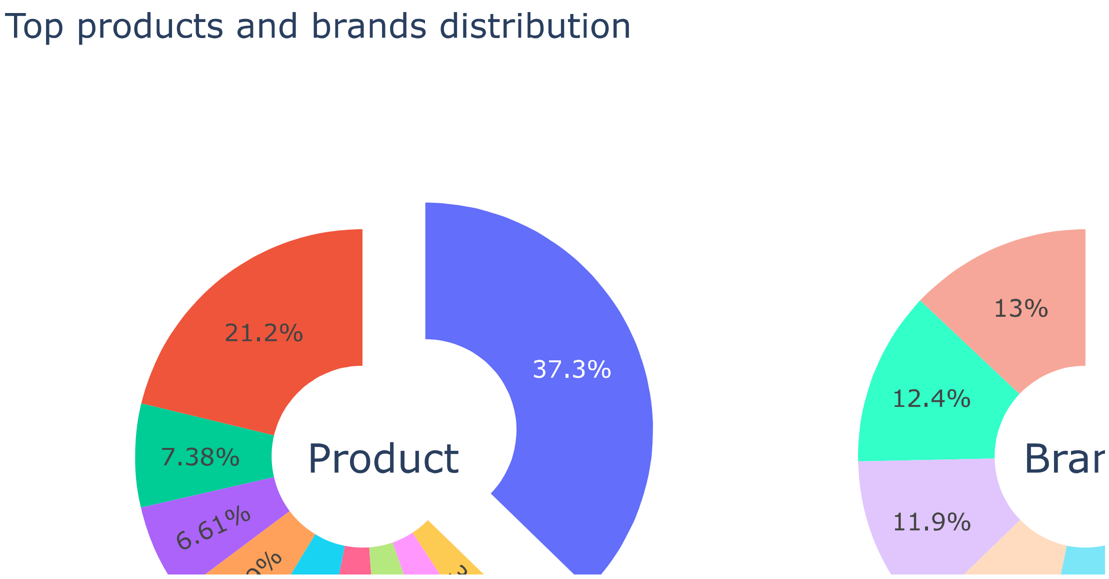
Out[31]:

	uniq_id	product_url	product_name	product
0	c2d766ca982eca8304150849735ffef9	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Cloth Clot
1	7f7036a6d550aaa89d34c77bd39a5e48	http://www.flipkart.com/fabhomedecor-fabric-do...	FabHomeDecor Fabric Double Sofa Bed	["Fu Room F
2	f449ec65dcbc041b6ae5e6a32717d01b	http://www.flipkart.com/aw-bellies/p/itmeh4grg...	AW Bellies	["Footw Footwe
3	0973b37acd0c664e3de26e97e5571454	http://www.flipkart.com/alisha-solid-women-s-c...	Alisha Solid Women's Cycling Shorts	["Cloth Clot
4	bc940ea42ee6bef5ac7cea3fb5cfbee7	http://www.flipkart.com/sicons-all-purpose-arn...	Sicons All Purpose Arnica Dog Shampoo	[' Gro

In [32]:
```
n = 10
top_products=pd.DataFrame(df['main_category'].value_counts()  [:n]).reset_index
top_products.rename(columns = {'index':'Top_Products','main_category':'Total_Co
                    inplace = True)

#Top 10 main brands being purchased

n = 10
top_brands=pd.DataFrame(df['brand'].value_counts()[:n]).reset_index()
top_brands.rename(columns = {'index':'Top_Brands','brand':'Total_Count'},  inpl
```

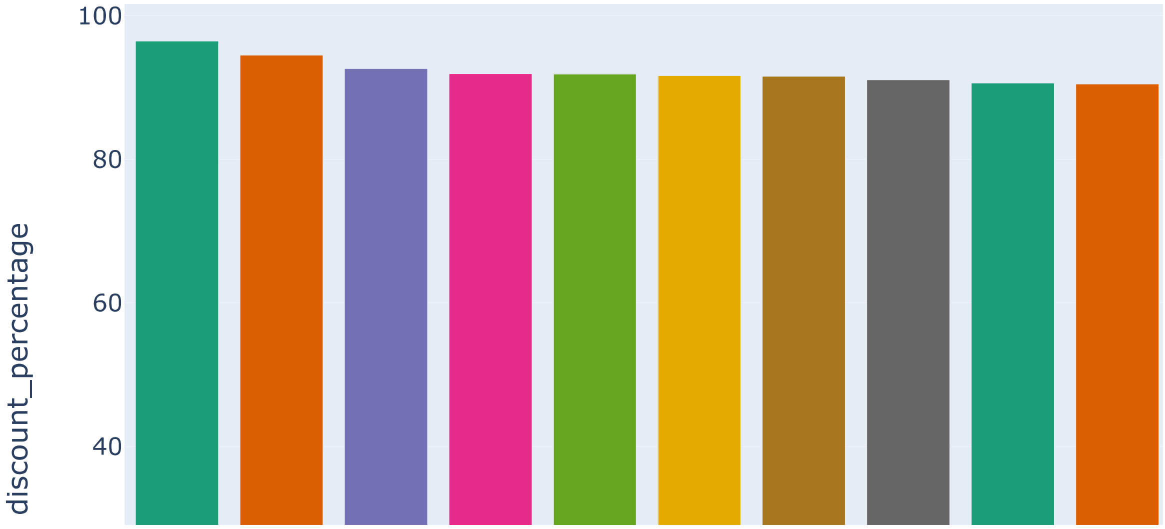
In [33]:

```
from plotly.subplots import make_subplots

label1 = top_products['Top_Products']
value1=top_products['Total_Count']
label2=top_brands['Top_Brands']
value2=top_brands['Total_Count']

# Create subplots

fig_both = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'}, {'type':'do
fig_both.add_trace(go.Pie(labels=label1, values=value1,
                          name="Top Products",pull=[0.3, 0, 0, 0]),
              1, 1)
fig_both.add_trace(go.Pie(labels=label2, values=value2,
                          name="Top Brands",pull=[0.3, 0, 0, 0]),
              1, 2)

# Use `hole` to create a donut-like pie chart

fig_both.update_traces(hole=.4, hoverinfo="label+percent+name")
#fig_both.update_traces(hoverinfo="label+percent+name")

fig_both.update_layout(
    title_text="Top products and brands distribution",
    #Add annotations in the center of the donut pies

    annotations=[dict(text='Product', x=0.18, y=0.5, font_size=20, showarrow=Fa
                 dict(text='Brand', x=0.82, y=0.5, font_size=20, showarrow=Fals
```

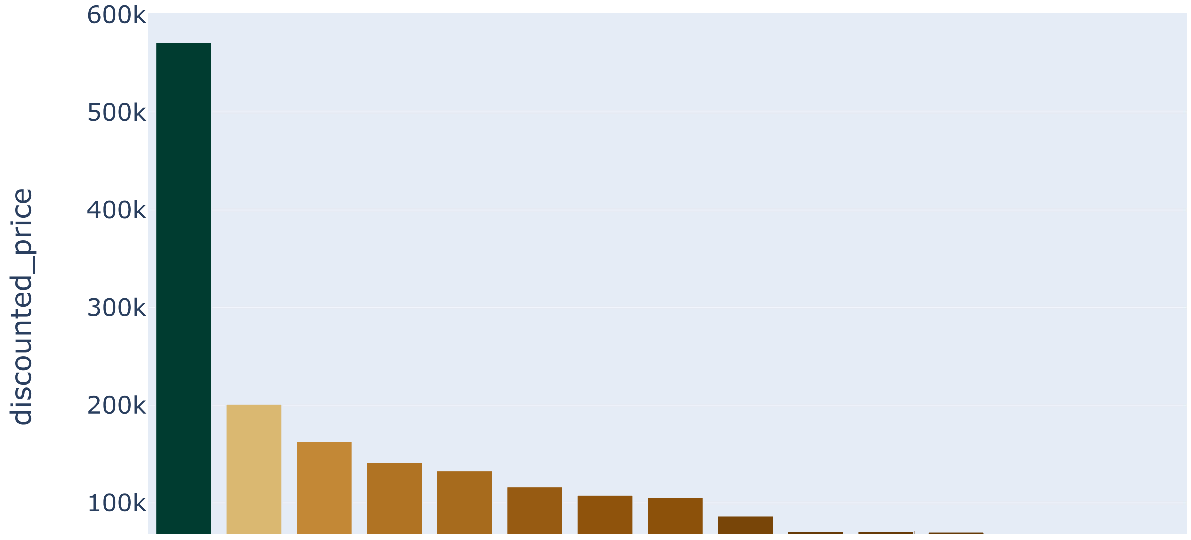


```
In [34]: df_discount=df.query('discount_percentage > 90')  #targeting brands giving high
         df_discount=df_discount.dropna() #dropping rows with NA values
         df_discount["brand"].replace('FashBlush','Fash Blush',inplace=True) #handling s
         max_discount=pd.DataFrame(df_discount.groupby('brand')[['discount_percentage']]
```

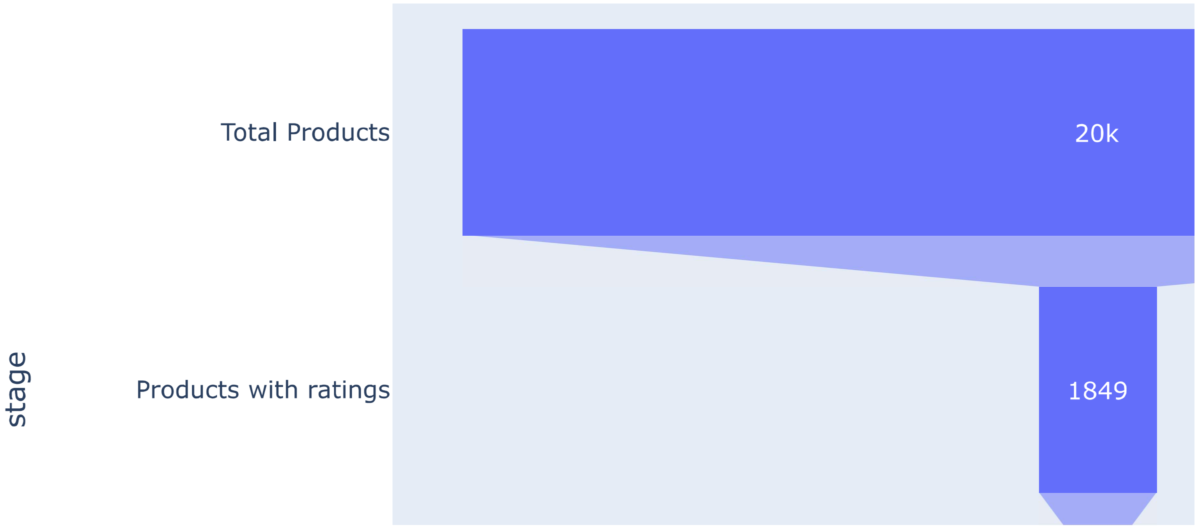
In [35]:
```
px.bar(max_discount, x= 'brand', y='discount_percentage',color='brand',color_di
```



In [36]:
```
df_customer=df.groupby("uniq_id")[["discounted_price"]].sum().sort_values(by=['

#Top 20 customers spending the most
list1=df_customer[:20]

#plotting a bar graph
px.bar(list1, x= 'uniq_id', y="discounted_price",color='discounted_price',color
```



In [37]:
```
# 5 star rating

total_prod=len(df['pid'])  #total products using pid variable
total_ratings=len(df[df['product_rating']!='No rating available']) #total ratec
top_ratings=len(df[df['product_rating']=='5']) #5 star rated products
df_funnel_1 = dict(
    number=[total_prod,total_ratings,top_ratings],
    stage=["Total Products","Products with ratings","Products with 5 star ratir
funnel_1_fig = px.funnel(df_funnel_1, x='number', y='stage')
funnel_1_fig.show()
```

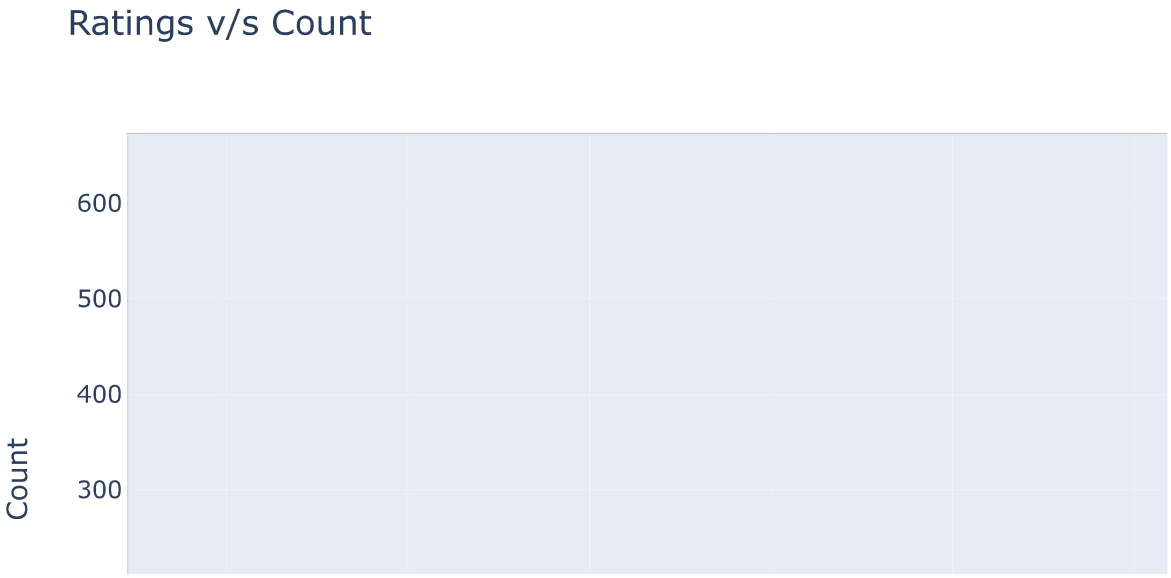


In [38]:

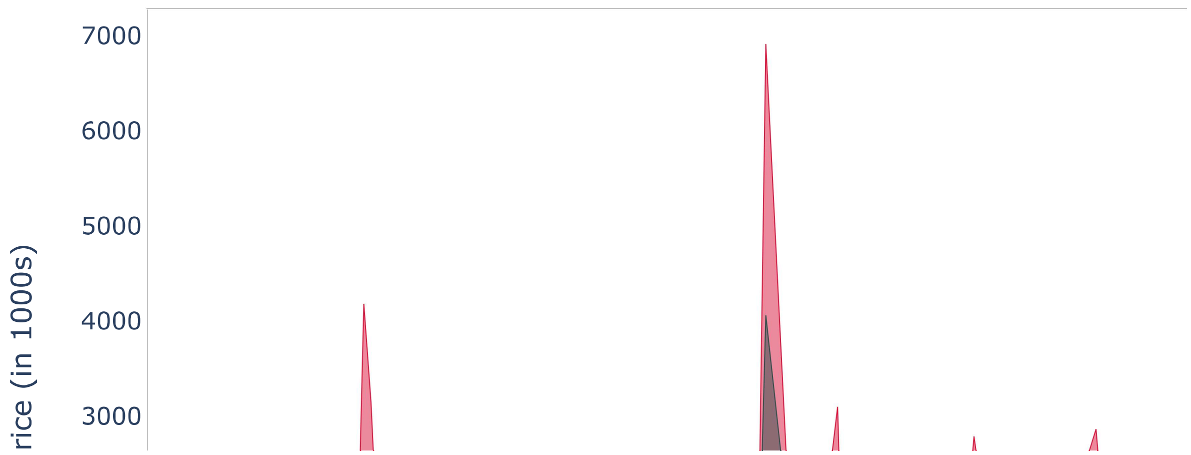
```
#5 star products/brands
rating_5=pd.DataFrame(df.loc[df['product_rating'] == '5'])
top_product_type=rating_5['main_category'].value_counts() #top products
top_brand_type=rating_5['brand'].value_counts()  #top brands

#top 5 products
df_top_product=pd.DataFrame(top_product_type[:5].reset_index()) #first 5
df_top_product.rename(columns = {'index':'top_prod'}, inplace = True)
df_top_product.drop('main_category', inplace=True, axis=1)

#top 5 brands
df_top_brand=pd.DataFrame(top_brand_type[:5].reset_index())
df_top_brand.rename(columns = {'index':'top_brands'}, inplace = True)
df_top_brand.drop('brand', inplace=True, axis=1)
df_top_brand.head()

#concatenating the 2 tables
df_product_brand_rate5=pd.concat([df_top_product,df_top_brand],axis=1)
```

In [39]:

```
df.drop(df.index[df['product_rating'] == 'No rating available'], inplace = True
ratings=pd.DataFrame(df['product_rating'].value_counts().reset_index())
ratings['index'] = ratings['index'].astype(float)
ratings.head().sort_values(by=['index'],ascending=[False])
ratings.rename(columns = {'index':'Ratings','product_rating':'Counts'}, inplace

#plotting the result
data=ratings
x=ratings['Ratings']
y=ratings['Counts']
figdot2 = go.Figure()
figdot2.add_trace(go.Scatter(
    x=x,
    y=y,
    marker=dict(color="crimson", size=12),
    mode="markers",
    name="ratings",
))

figdot2.update_layout(title="Ratings v/s Count",
                  xaxis_title="Ratings",
                  yaxis_title="Count",
                     )

figdot2.update_xaxes(showline=True, linewidth=1, linecolor='black', mirror=True
figdot2.update_yaxes(showline=True, linewidth=1, linecolor='black', mirror=True
figdot2.show()
```


Ratings v/s Count
600
500
400
300
Count

In [40]:
```
df_date_retail = pd.DataFrame(df.groupby("date")[["retail_price"]].mean().reset
df_date_discount = pd.DataFrame(df.groupby("date")[["discounted_price"]].mean()
df_date_price=pd.concat([df_date_retail,df_date_discount],axis=1)
df_date_price = df_date_price.loc[:,~df_date_price.columns.duplicated()] #remo

#Plot
x=df_date_price['date']
y1=df_date_price['retail_price']
y2=df_date_price['discounted_price']

fig_area2 = go.Figure()
fig_area2.add_trace(go.Scatter(x=x, y=y1, fill='tozeroy',name='retail price',
                               line=dict(width=0.5, color='crimson'))) # fill d
fig_area2.add_trace(go.Scatter(x=x, y=y2, fill='tozeroy',name='discount price',
                               line=dict(width=0.5, color='darkslategray')
                              )) # fill to trace0 y

fig_area2.update_layout(
    xaxis_title="Dates",
    yaxis_title="Price (in 1000s)",
    plot_bgcolor='white'
)
fig_area2.update_xaxes(showline=True, linewidth=1, linecolor='black', mirror=Tr
fig_area2.update_yaxes(showline=True, linewidth=1, linecolor='black', mirror=Tr
fig_area2.show()
```



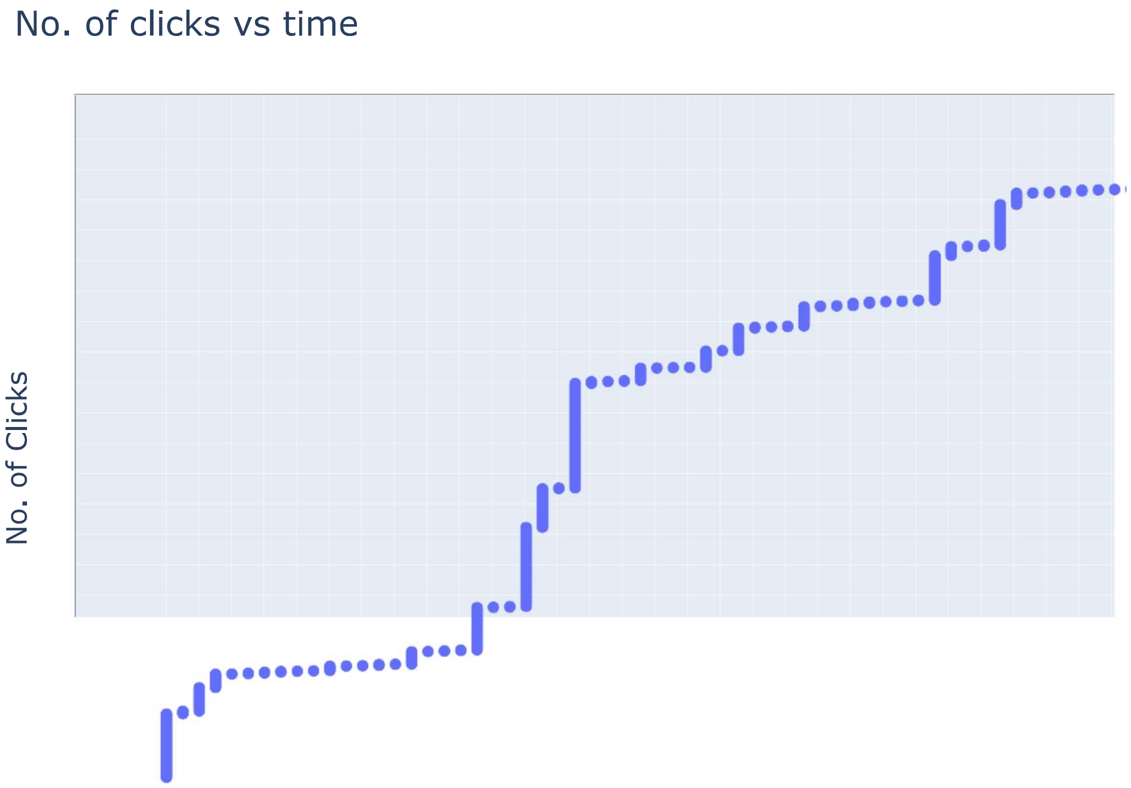
```
In [41]: df.head()
```

Out[41]:

	uniq_id	product_url	product_name	product_cate
10	e54bc0a7c3429da2ebef0b30331fe3d2	http://www.flipkart.com/ladela-bellies/p/itmeh...	Ladela Bellies	["Footwear >: Footwear >>
27	bec784ef794cf596dbe2cbbaf5427ef0	http://www.flipkart.com/bulaky-vanity-case-jew...	Bulaky vanity case Jewellery Vanity Case	["Beauty an Care >> I
59	d620fa0d35825bb3c0717e9d3446cc97	http://www.flipkart.com/roadster-men-s-zipper-...	Roadster Men's Zipper Solid Cardigan	["Clothin Clothing >
94	f355cc1ccb08bd0d283ed979b7ee7515	http://www.flipkart.com/camerii-wm64-elegance-...	Camerii WM64 Elegance Analog Watch - For Men,...	["Watche Watches :
97	c0824c9e7ee6b79006ce698a2a7a413c	http://www.flipkart.com/colat-colat-mw20-sheen...	Colat COLAT_MW20 Sheen Analog Watch - For Men...	["Watche Watches >>

In [42]:

```
scat2 = px.scatter(x=df['Time'].sort_values(ascending=True), y=df['product_url'
scat2.update_layout(
    title_text='No. of clicks vs time', # title of plot
    xaxis_title_text='Time', # xaxis label
    yaxis_title_text='No. of Clicks', # yaxis label

)
#scat.update_xaxes(showticklabels=False)
scat2.update_yaxes(showticklabels=False)
scat2.update_xaxes(showline=True, linewidth=1, linecolor='black', mirror=True)
scat2.update_yaxes(showline=True, linewidth=1, linecolor='black', mirror=True)
scat2.show()
```



In [ ]:

In [ ]:

In [ ]: