

Ben Deatsman

Data Visualization Lab

In [35]:

```
from completejourney_py import get_data

cj_data = get_data()

import pandas as pd

transactions = cj_data['transactions']
products = cj_data['products']
demographics = cj_data['demographics']

from bokeh.plotting import figure, output_file, show
from bokeh.resources import INLINE
import bokeh.io
from bokeh.palettes import Category20

bokeh.io.output_notebook()
```

BokehJS 3.3.4 successfully loaded.

In [15]:

transactions

Out[15]:

	household_id	store_id	basket_id	product_id	quantity	sales_value	retail_disc	coupon_disc	coupon_match_disc	week	transaction_timestamp	
	0	900	330	31198570044	1095275	1	0.50	0.00	0.0	0.0	1	2017-01-01 11:53:26
	1	900	330	31198570047	9878513	1	0.99	0.10	0.0	0.0	1	2017-01-01 12:10:28
	2	1228	406	31198655051	1041453	1	1.43	0.15	0.0	0.0	1	2017-01-01 12:26:30
	3	906	319	31198705046	1020156	1	1.50	0.29	0.0	0.0	1	2017-01-01 12:30:27
	4	906	319	31198705046	1053875	2	2.78	0.80	0.0	0.0	1	2017-01-01 12:30:27
...
1469302	679	447	41453103606	14025548	1	0.79	0.20	0.0	0.0	53	2018-01-01 03:50:03	
1469303	2070	311	41453083334	909894	1	1.73	0.17	0.0	0.0	53	2018-01-01 04:01:20	
1469304	2070	311	41453083334	933067	2	5.00	2.98	0.0	0.0	53	2018-01-01 04:01:20	
1469305	2070	311	41453083334	1029743	1	2.60	0.29	0.0	0.0	53	2018-01-01 04:01:20	
1469306	2070	311	41453083334	1061220	1	1.19	0.13	0.0	0.0	53	2018-01-01 04:01:20	

1469307 rows x 11 columns

In [16]:

products

Out[16]:

	product_id	manufacturer_id	department	brand	product_category	product_type	package_size
0	25671	2	GROCERY	National	FRZN ICE	ICE - CRUSHED/CUBED	22 LB
1	26081	2	MISCELLANEOUS	National	None	None	None
2	26093	69	PASTRY	Private	BREAD	BREAD:ITALIAN/FRENCH	None
3	26190	69	GROCERY	Private	FRUIT - SHELF STABLE	APPLE SAUCE	50 OZ
4	26355	69	GROCERY	Private	COOKIES/CONES	SPECIALTY COOKIES	14 OZ
...
92326	18293142	6384	DRUG GM	National	BOOKSTORE	PAPERBACK BOOKS	None
92327	18293439	6393	DRUG GM	National	BOOKSTORE	CHILDRENS LOW END	None
92328	18293696	6406	DRUG GM	National	BOOKSTORE	PAPERBACK BEST SELLER	None
92329	18294080	6442	DRUG GM	National	BOOKSTORE	PAPERBACK BOOKS	None
92330	18316298	764	GROCERY	National	PAPER TOWELS	PAPER TOWELS & HOLDERS	None

92331 rows x 7 columns

In [17]:

demographics

Out[17]:

	household_id	age	income	home_ownership	marital_status	household_size	household_comp	kids_count
0	1	65+	35-49K	Homeowner	Married	2	2 Adults No Kids	0
1	1001	45-54	50-74K	Homeowner	Unmarried	1	1 Adult No Kids	0
2	1003	35-44	25-34K	None	Unmarried	1	1 Adult No Kids	0
3	1004	25-34	15-24K	None	Unmarried	1	1 Adult No Kids	0
4	101	45-54	Under 15K	Homeowner	Married	4	2 Adults Kids	2
...
796	986	25-34	35-49K	None	Unmarried	1	1 Adult No Kids	0
797	992	45-54	35-49K	Homeowner	Married	3	2 Adults Kids	1
798	993	55-64	50-74K	Homeowner	Married	5+	1 Adult Kids	3+
799	996	55-64	25-34K	Homeowner	Married	2	2 Adults No Kids	0
800	997	45-54	75-99K	Homeowner	Unmarried	1	1 Adult No Kids	0

801 rows x 8 columns

In [18]:

```
transactions_products = pd.merge(transactions, products, on='product_id')
df = pd.merge(transactions_products, demographics, on='household_id')
```

In [19]:

```
sales_cat = df.groupby('product_category')['quantity'].sum().reset_index().sort_values('quantity', ascending=False)
sales_cat = pd.DataFrame(sales_cat)
sales_cat = sales_cat.iloc[1:]
sales_cat = sales_cat.iloc[0:10]
sales_cat
```

Out[19]:

	product_category	quantity
266	SOFT DRINKS	47769
121	FLUID MILK PRODUCTS	37628
13	BAKED BREAD/BUNS/ROLLS	30700
50	CHEESE	30649
141	FUEL	28831
298	YOGURT	27183
267	SOUP	27150
136	FRZN MEAT/MEAT DINNERS	25823
291	VEGETABLES - SHELF STABLE	25643
12	BAG SNACKS	25331

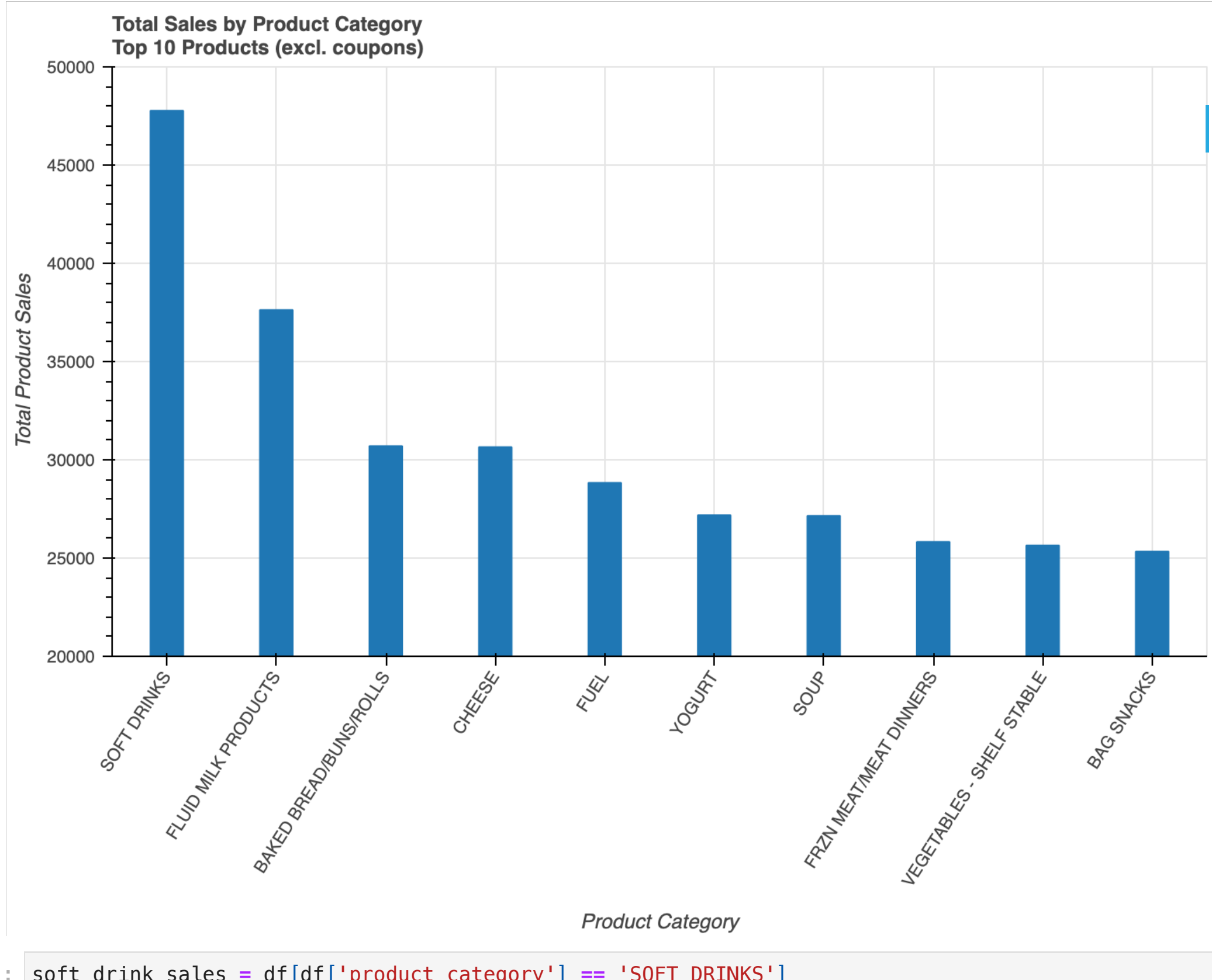
In [85]:

```
categories = sales_cat['product_category']
data = sales_cat['quantity']

p1 = figure(title= 'Total Sales by Product Category\nTop 10 Products (excl. coupons)', x_range=categories, x_axis_label='Product Category', y_axis_label='Total Product Sales')
p1.vbar(x=categories, top=data, width=.3)

p1.xaxis.major_label_orientation = 1.0

show(p1)
```



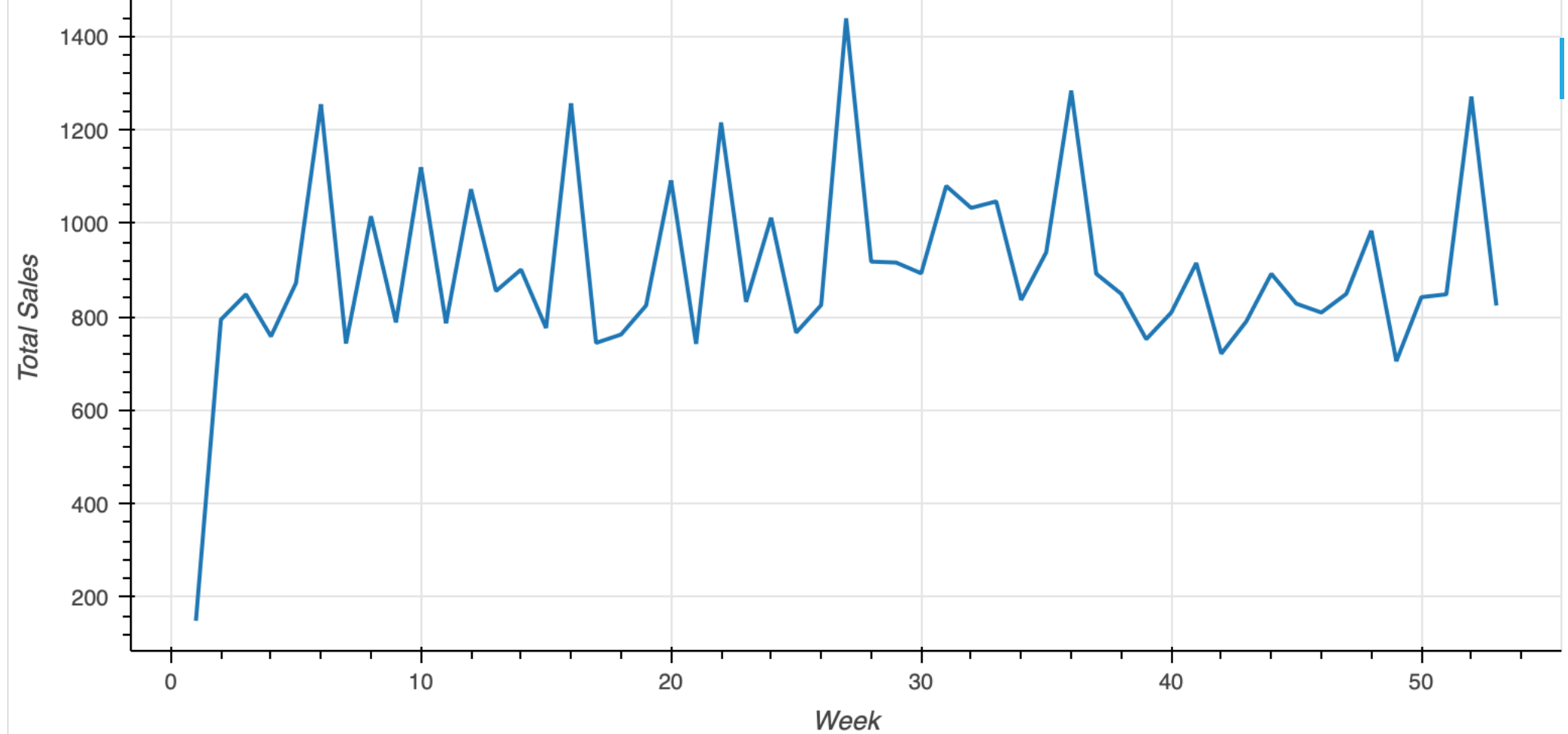
In [83]:

```
soft_drink_sales = df[df['product_category'] == 'SOFT DRINKS']

weekly_sales = soft_drink_sales.groupby('week')['quantity'].sum().reset_index()

p2 = figure(title='Total Sales of Soft Drinks by Week', x_axis_label='Week', y_axis_label='Total Sales', width=800, height=400)
p2.line(x='week', y='quantity', source=weekly_sales, line_width=2)

show(p2)
```



In [137]:

```
soda_filter = products['product_category'].str.contains('SOFT DRINKS', case=False, na=False)
soda_products = products[soda_filter]

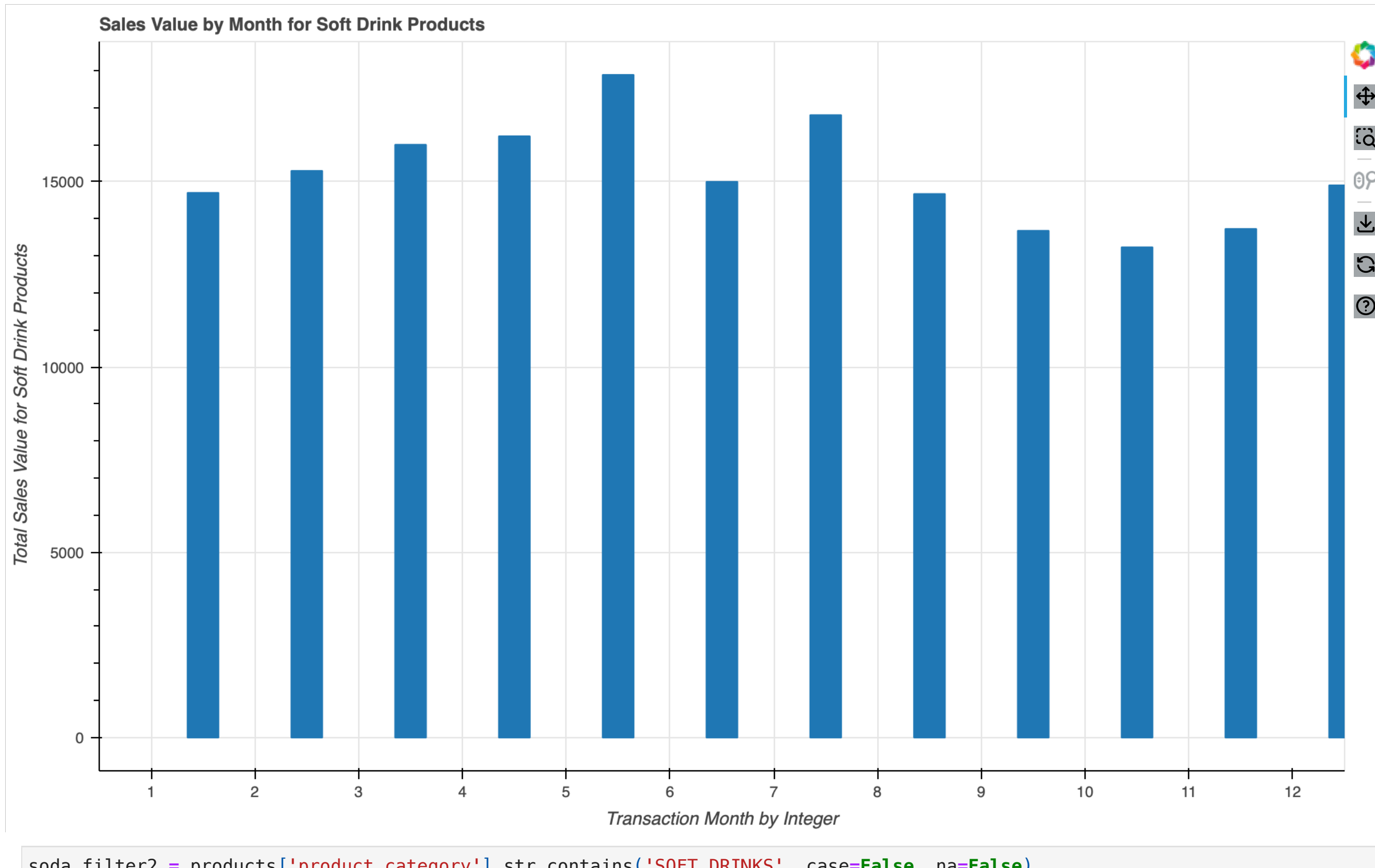
soda_merge = soda_products.merge(cj_data['transactions'], how='inner', on='product_id')
soda_merge['transaction_month'] = pd.to_datetime(soda_merge['transaction_timestamp']).dt.month

soda_month = (soda_merge.groupby('transaction_month', as_index=False)
               .agg({'sales_value': 'sum'})
               .sort_values(by='transaction_month', ascending=True))

categories = soda_month['transaction_month']
data = soda_month['sales_value']

p3 = figure(title= 'Sales Value by Month for Soft Drink Products', x_range=[str(month) for month in categories], x_axis_label='Transaction Month by Integer', y_axis_label='Total Sales Value for Soft Drink Products')
p3.vbar(x=categories, top=data, width=.3)

show(p3)
```



In [177]:

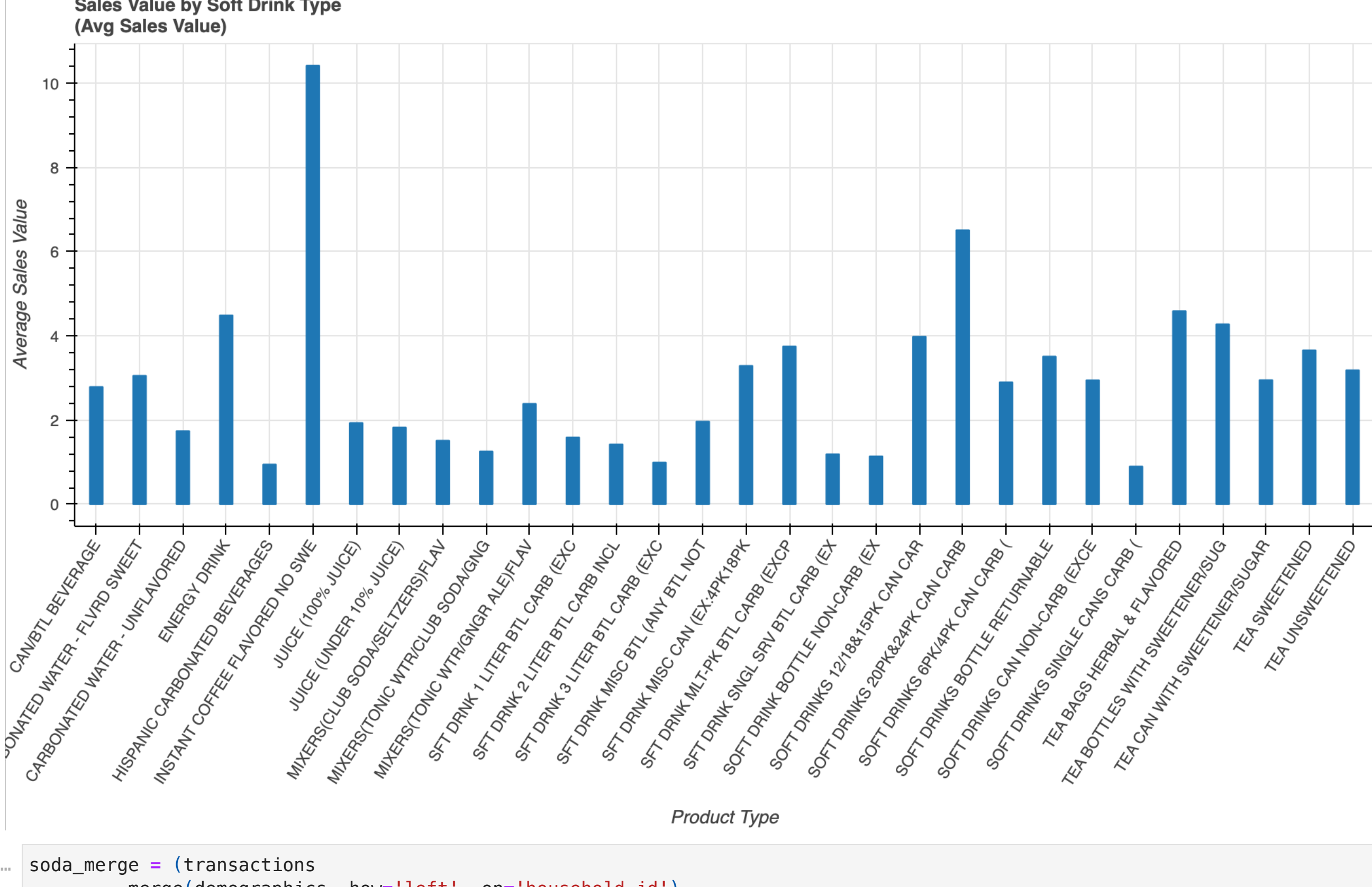
```
soda_filter2 = products['product_category'].str.contains('SOFT DRINKS', case=False, na=False)
soda_data = products[soda_filter2].merge(transactions, how='inner', on='product_id')
soda_sales_by_type = (soda_data
                      .groupby(['product_type'])
                      .agg({'sales_value': 'mean'})
                      .reset_index())

categories = soda_sales_by_type['product_type']
data = soda_sales_by_type['sales_value']

p4 = figure(title= 'Sales Value by Soft Drink Type(Avg Sales Value)', x_range=[str(month) for month in categories], x_axis_label='Product Type', y_axis_label='Average Sales Value')
p4.vbar(x=categories, top=data, width=.3)

p4.xaxis.major_label_orientation = 1.0

show(p4)
```



In [213]:

```
soda_merge = (transactions
              .merge(demographics, how='left', on='household_id')
              .merge(products[products['product_category'].str.contains('SOFT DRINKS', case=False, na=False)],
                     how='inner', on='product_id'))

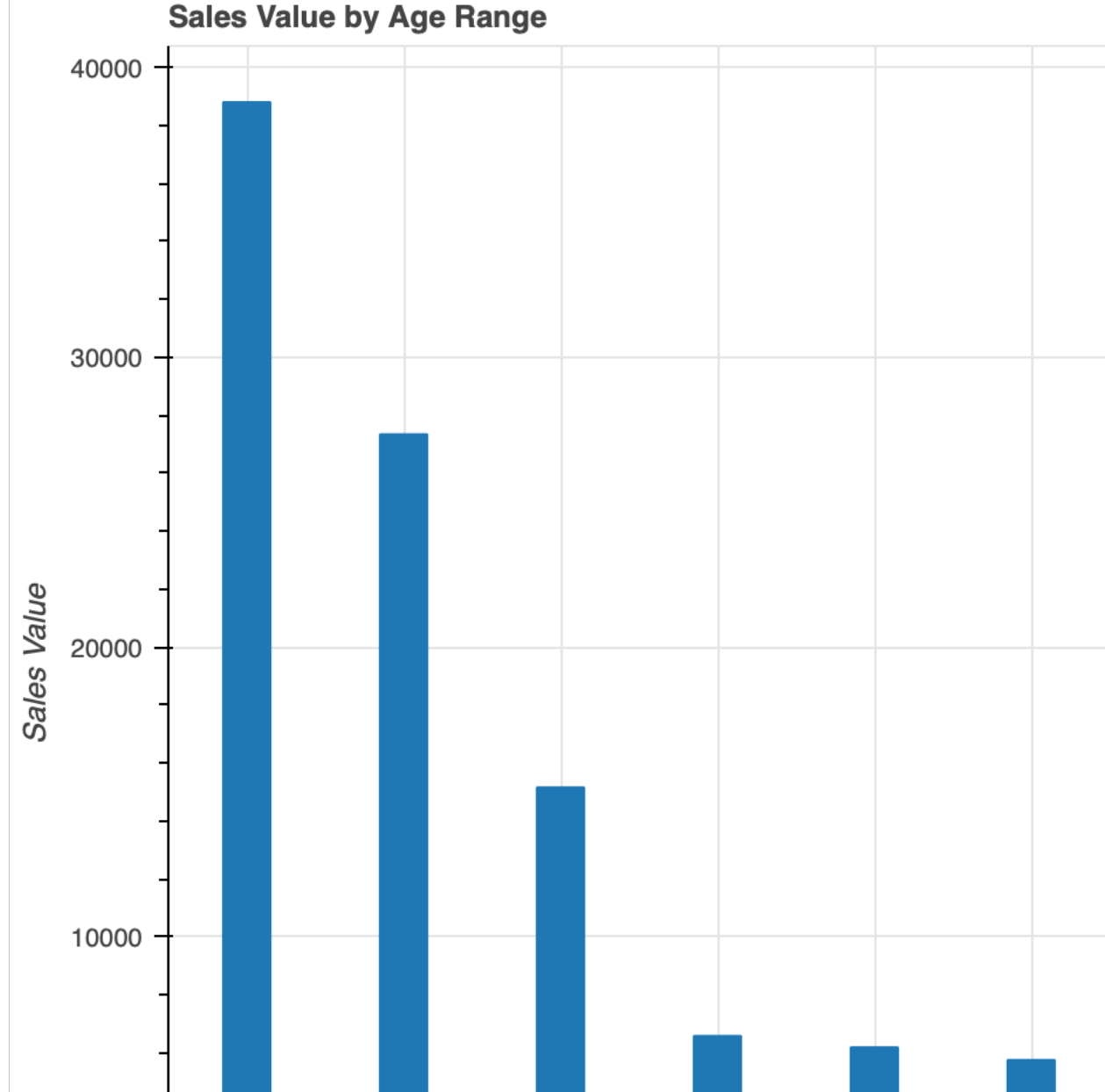
soda_age = (filtered_data
            .groupby('age', as_index=False)
            .agg({'sales_value': 'sum'})
            .sort_values(by='sales_value', ascending=False))

soda_age

categories = soda_age['age']
data = soda_age['sales_value']

p5 = figure(title= 'Sales Value by Age Range', x_range=categories, x_axis_label='Age Range', y_axis_label='Sales Value', height=600, width=500)
p5.vbar(x=categories, top=data, width=.3)

show(p5)
```



In []: