Final Assignment

In this final assignment I was challenged to to find the most popular item per zipcode and the percentage of sales per store in the period between 2016-2019. The languages I used are SQL and Python with Jupyter Notebooks.

First of all, I download the data (in sql format from the Workearly platform) and tried to make a query, which is gonna return all the liquor products from period 2016 to 2019. The query I made is:

```
277
278 • select * from finance_liquor_sales where date  between '2016-01-01' and '2019-12-31';
279
280
```

The next step was to export data as a csv file and make a jupyter notebook to begin with the second part which is to aggregate the data and get the most popular item sold based on zip code and percentage of sales per store. To start with, I made a new DataFrame called grouped. I filtered the fields *'zip_code'* and *'item_description'* with *'bottles_sold'* using the *groupby()* method and also the *methods .sum()* to calculate the summary of the ... and the *reset_index()* to reset the index of the basic DataFrame(called df). The next step I did, was to make another filter instance idx with already filtered instance called grouped, which groups the groupby instance with the *'zip_code'* and *'bottles_sold'* fields from DataFrame and in the end using the *.idxmax()* method which return the index of the first record from the DataFrame. Also, I made another DataFrame called 'most_popular_items' which equals with the indexes of the grouped DataFrame. To complete the filtered data task, I made a plot to present the results of the filtered data using the *scatterplot()* method(Seaborn library)

```
[10]: grouped = df.groupby(['zip_code', 'item_description'])['bottles_sold'].sum().reset_index()
      idx = grouped.groupby(['zip_code'])['bottles_sold'].idxmax()
      most_popular_items = grouped.loc[idx]
      most_popular_items
```

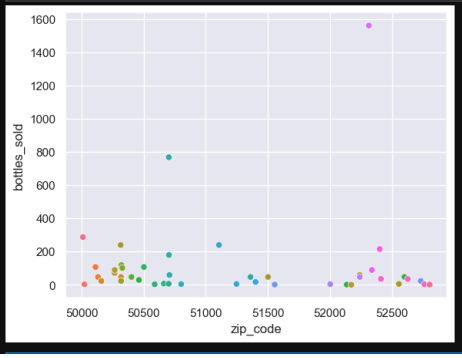
[10]:

	zip_code	item_description	bottles_sold
0	50010.0	Member's Mark Spiced Rum	288
1	50022.0	Paramount Triple Sec	4
2	50111.0	Saints N Sinners Apple Pie	108
3	50131.0	Platinum 7x Vodka	48
4	50158.0	Hennessy VS	24
7	50263.0	Jagermeister w/2 Shot Glasses	84
8	50265.0	Kahlua Coffee	72
9	50266.0	Avion Silver w/Powell & Mahoney Margarita Mix	90
11	50314.0	Juarez Triple Sec	240
14	50316.0	Hennessy VS	48
15	50317.0	Paul Masson Peach Grande Amber Brandy	24
16	50320.0	Di Amore Quattro Orange	120
19	50327.0	Bacardi Gold Rum PET	102
20	50401.0	Ole Smoky Blackberry Moonshine Mini	48
22	50461.0	Five O'clock Vodka	30
23	50501.0	Titos Vodka	108
24	50588.0	Phillips Root Beer Schnapps	4
25	50662.0	Ole Smoky Apple Pie Moonshine 70prf Mini	8
26	50701.0	Bacardi Gold	7
30	50702.0	Tortilla Gold Dss	768
33	50703.0	Pinnacle Peach w/ Punch Dispenser	180
35	50707.0	Jameson	60
36	50801.0	Bacardi Gold Rum	5

I presented the results in the above scatterplot:

```
[7]: #visualize the data

[24]: sns.set_theme()
      sns.scatterplot(data=most_popular_items,x="zip_code",y="bottles_sold", hue="item_description", legend=False)
      plt.savefig('scatterplot.png')
```



The second step was to calculate the percentage and return the sales per store. To do this, first I created a new Dataframe called ‘sales_per_store’ and group the fields *‘store_number’* and *‘store_name’* with *‘sale_dollars’* using the *.sum()* and reset_index method to get the summary of the each sale by store and reset the index as happened in the previous case with most popular items. In the new DataFrame, I created and extra field called *‘percentage’* and there I calculated the percentage from field *‘sale_dollars’* and *‘sale_dollars’.sum()* which is the sales summary of all stores and all

multiplied with 100. Also I sorted all the values from the DataFrame by percentage, with *sort_values()* and *reset_index()* method.

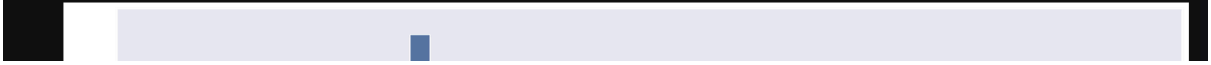
```
[29] Toggle output scrolling
sales_per_store = pd.DataFrame(df.groupby(by=['store_number','store_name']).sum()['sale_dollars']).reset_index()
# sales_per_store['percentage'] = ((df['sale_dollars'] * df['bottles_sold']) / df['sale_dollars'].sum()) * 100).map('{:.2%}'.format)
sales_per_store['percentage'] = ((df['sale_dollars'] / df['sale_dollars'].sum()) * 100).map('{:.2f}'.format)
sales_per_store.sort_values(by='percentage', ascending=False).reset_index()
# sales_per_store.head()
```

[29]:

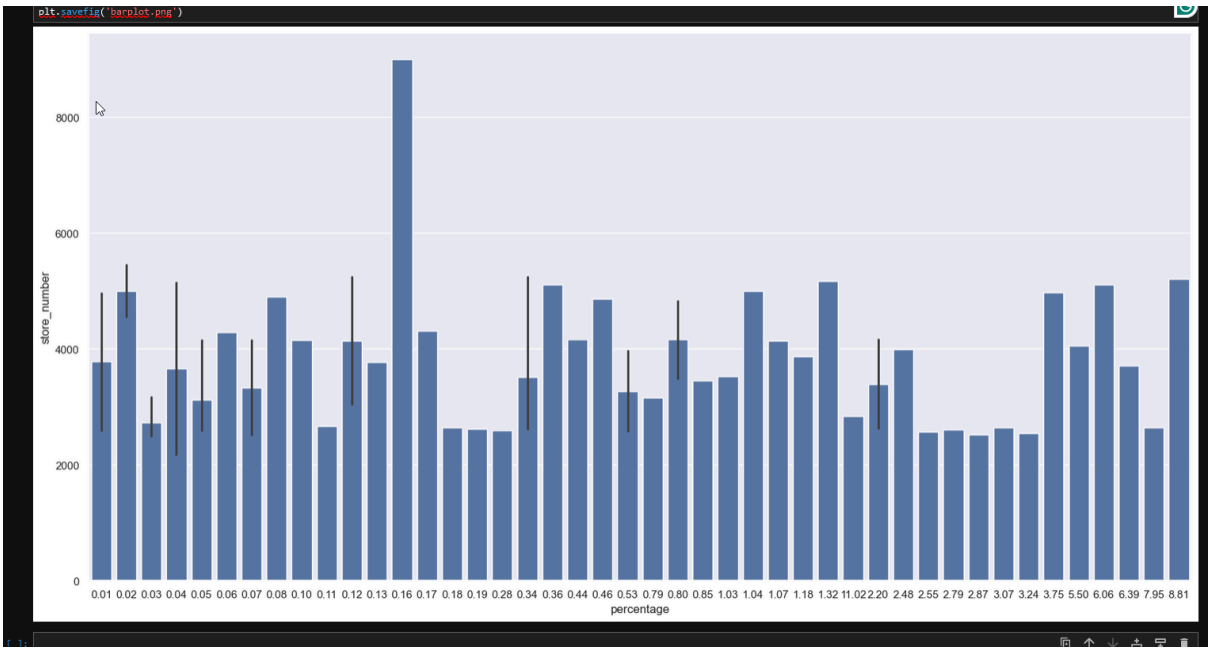
	index	store_number	store_name	sale_dollars	percentage
0	56	5204	Quik Stop / Burlington	81.60	8.81
1	19	2643	Hy-Vee Wine and Spirits / Waterloo	7.00	7.95
2	30	3705	Liquor Locker	37.50	6.39
3	52	5102	Wilkie Liquors	11620.80	6.06
4	35	4057	Tequila's Liquor Store	413.28	5.50
5	49	4971	Fareway Stores #138 / Pleasant Hill	2295.00	3.75
6	6	2544	Hy-Vee Food Store / Marshalltown	75.54	3.24
7	20	2647	Hy-Vee #7 / Cedar Rapids	486.00	3.07
8	4	2522	Hy-Vee Wine and Spirits / Spirit Lake	324.96	2.87
9	13	2601	Hy-Vee Food Store / Fairfield	21.78	2.79
10	8	2571	Hy-Vee Food Store #2 / Waterloo	1992.15	2.55
11	34	3993	New Star Liquor / W 4th S / Waterloo	80.49	2.48
12	16	2633	Hy-Vee #3 / BDI / Des Moines	4124.04	2.20
13	40	4158	Fareway Stores #409 / Carroll	32.52	2.20
14	23	2843	CVS Pharmacy #8526 / Cedar Rapids	42.00	11.02
15	55	5166	East Side Liquor and Groceries	206.64	1.32
16	32	3869	Bootleggin' Barzini's Fin	6.75	1.18
17	36	4136	Fareway Stores #983 / Grimes	1296.00	1.07
18	50	5003	Famous Liquors	17.98	1.04
19	28	3524	Sam's Club 6568 / Ames	3913.92	1.03
20	26	3447	Sam's Club 6432 / Sioux City	6641.04	0.85

Finally I visualize the data with *barplot()* method, based in sorted data from the DataFrame. Also I set the axes just to be more user friendly.

```
[27]: plt.figure(figsize=(20,10))
sns.set_theme(font_scale=1)
sns.set_theme()
sns.color_palette()
ax = sns.barplot(data=sales_per_store.sort_values(by='percentage', ascending=True), x="percentage", y="store_number")
# ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
plt.savefig('barplot.png')
```



And here is the full barplot:



Of course I faced some difficulties and the first it was the filtering. I was a little confused about the order of the fields inside the parenthesis and which field should be outside just to get the right results. Also I had problem with the calculation of percentage with the wrong way and the results was very confusing. For all of this issues, I searched a lot on the Internet