Paul Kim

Library as third space

This project built on the first assignment of this course which was to analyze how the library was being used as a space itself rather than just as a repository. My proxy for this analysis was by looking at which items were checked out and back in on the same day. Below is the SQL query that I used to pull these items:

```
SELECT
  *
FROM
  spl_2016.inraw
WHERE
  DAY(cout) = DAY(cin)
    AND TIMESTAMPDIFF(DAY, cout, cin) < 1
```

In the last assignment, I did an analysis that revealed that non-books/DVD material, such as laptops, headphones, and other equipment, made up a significant portion of these items. My original idea for this project was to do a mapping of which branches saw how many of these same-day equipment checkouts, but some investigating into the inventory data uploaded on the Seattle Open Data Portal revealed that there are no unique identifiers for these types of equipment, meaning that they cannot actually be traced to branches. I then changed my project to analyzing the number of checkouts of these equipment compared to their total inventory by month over time to see how frequently these items were being used in comparison to how much of them the library had. This required a lot of data pre-processing which will be discussed in the next step.

Data Pre-processing

A lot of work was done in Python to merge the inventory and checkout data. The tasks were as follows:

1. Pull checkout data
2. Pull inventory data
3. Merge checkout data and inventory data
    a. This was the most challenging part as it required keeping a data column that would keep track of the inventory over time as it updates. So basically, every time the inventory changed, I'd have to have the new inventory reflect in the merged dataset. Additionally, since the inventory was missing some data, I had to impute the missing data. Below are some screenshots of my Python code:

```
for i in range(len(df_cins_eq_only_2017)):
    tot=0
    cout_mt = df_cins_eq_only_2017['cout'].iloc[i].month
    cout_yr = df_cins_eq_only_2017['cout'].iloc[i].year
    cout_mtyr = str(cout_mt) + str(cout_yr)
    cout_bib = df_cins_eq_only_2017['bibNumber'].iloc[i]
    temp_df = eqcounts[eqcounts['BibNum'] == cout_bib]
    temp_df = temp_df[temp_df['ReportDate'].dt.month.astype(str) + temp_df['ReportDate'].dt.year.astype(str) ==
    if len(temp_df) == 0:
        tot = 0
    else:
        tot = temp_df['ItemCount'].iloc[0]
    df_cins_eq_only_2017['Running Total'].iloc[i] = tot
```

```
/Users/paulkim/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:20: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
for i in range(1,len(df_cins_eq_only_2017)):
    if df_cins_eq_only_2017['Running Total'].iloc[i] == 0:
        bibnum = df_cins_eq_only_2017['bibNumber'].iloc[i]
        curr_date = df_cins_eq_only_2017['cout'].iloc[i]
        bibdf = df_cins_eq_only_2017[df_cins_eq_only_2017['bibNumber'] == bibnum]
        zero_index = np.where(bibdf[bibdf['cout']==curr_date])[0][0]
        zero_indices = np.where(bibdf[bibdf['cout']==curr_date])
        impute_tot = bibdf['Running Total'].iloc[zero_index-1]
        df_cins_eq_only_2017['Running Total'].iloc[i] = impute_tot
```

```
/Users/paulkim/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
```

In the first part of the code you will see the column “Running Total” created that keeps track of the inventory by month. In the second part of the code you will see where I imputed some missing data of the inventory. The imputation was done by just inserting the most recent inventory numbers for any missing data.

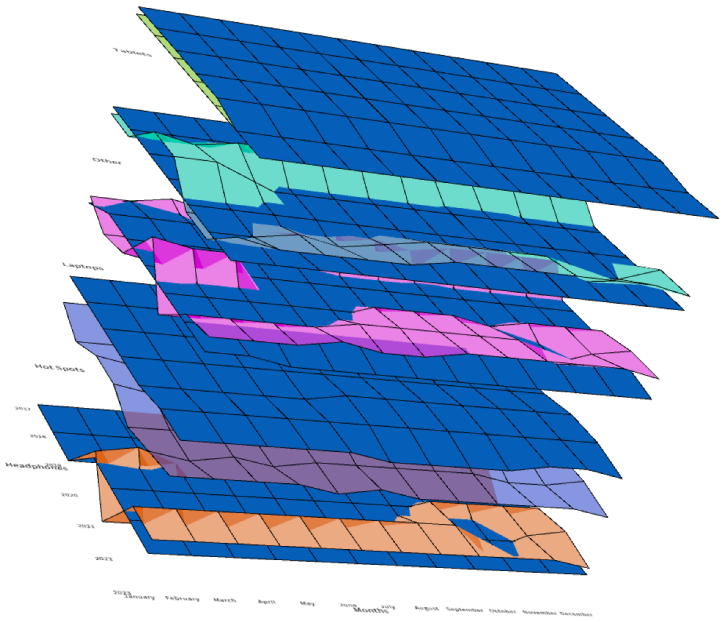
Process

I borrowed a similar idea from Li Zheng in the visualization since it allowed for a visualization of two different things on the same plane. Since I wanted to compare the inventory to checkouts data, I thought this would be a nice way of seeing this comparison.

I split the view into five different layers because I had five different equipment types so one could see the comparison of how many times certain pieces of equipment were checked out vs. how many were available, by month. Every time the non-blue graph goes above the flatter inventory line, that equipment type was checked out more times than the library had in inventory. This makes sense because the data is by month, and each item can be checked out multiple times.

Paul Kim

Below is a screenshot of my visualization.



If you mouse over the peaks of each graph, you will see the number of checkouts and the number of inventory at each month for each equipment type. I will be working on adding more labels, figuring out a better color scheme, the ordering of the data layers, and some other design choices. The final visualization is to come, but you can definitely still see some interesting insights in this visualization as it is. Some insights:

1. How the library responds to demand over time. It seems to generally lag, but they also generally responded to demand in some way. One interesting insight is that the library seemed to reduce the inventory of tablets due to lack of interest.
2. Headphones and laptops are often checked out at high frequencies, whereas tablets and hot spots are hardly checked out at all.
3. There is a huge number of hot spots available for checkout, but they are rarely checked out.
4. You can see a predictable trough in each of these materials during COVID.