Fall 2022 MAT265 Random Samplings

- Shaokang Li

Introduction:

Random Sampling is a method to estimate characteristics of the whole population by sample a subset within the whole population randomly. For this week's assignment, I am interested in questions below:

- By using random sampling technique, I try to find if there is any pattern within the sampled data? Is the pattern any different from the whole population?
- If the pattern sampled in CD is any different from that in DVD or Books?
    - For each media type, sample 3 times in a row, compare their differences.
- Note that the bibNumber and itemNumber follows a linear pattern within a lower itemNumber range. If we randomly sample the bibNumber and itemNumber within the same range, will the same pattern appear?

Question 01:

For the first question. I first tried out the general query to random sampling in the whole `inraw` table by using query below.

Initial Query:

This query should randomly return 100 records from the `inraw` table. However, a error message was returned instead.

```
SELECT *
FROM spl_2016.inraw
order by rand(42) LIMIT 100;
```

> Message:
>
> SQL: SELECT * FROM spl_2016.inraw order by rand(42) LIMIT 100
>
> The table '/tmp/#sql_8a1_20' is full

After searching on stackoverflow, the error is likely due to a limit in the size of temporary table. If we don't apply any restriction to the query, the temporary table will exceeds the size limit.

Thus, I decided to do the random sampling in the CD category and set up a time scope from 2022 to today.

Updated Query

Query below is used to find the checkout times of a single title, with a randomly selection size of 100.

```
SELECT
    count(cout) as checkouts,
    bibNumber,
    title
FROM spl_2016.inraw
where
    itemtype in (
        'arcd',
        'nacd',
        'jrcd',
        'accd',
        'cacd',
        'cccd',
        'jccd',
        'nccd'
    )
    and cout > '2022-01-01'
group by bibNumber,title
order by rand(42)
LIMIT 100;
```

Result:

> See attached CD_100_#.csv for results.

After getting the result, I applied the deviation-based outlier method from last week to find out the ratio of outliers in this randomly sampled data. The outliers are the CDs with extremely low or high checkout times.And I'm interested in the difference between sampled data and the whole data.

```
import pandas as pd
import numpy as np

df = pd.read_csv('CD_100_01.csv')
df.head()
print(df.shape)

diff = df.checkouts.quantile(0.75)-df.checkouts.quantile(0.25)
upper_limit = df.checkouts.quantile(0.75) + 1.5*diff
lower_limit = df.checkouts.quantile(0.25) - 1.5*diff
print(upper_limit, lower_limit)

filtered_data = df[df.checkouts>upper_limit].sort_values(by=
['checkouts'],ascending=False)
ratio = filtered_data.shape[0]/df.shape[0]
print(ratio)
```

```
filtered_data.head()
```

Results are listed below:

Sample Times	Upper Limit for Non-outlier	Lower Limit for Non-outliers	Outlier Ratio
01	9.125	-3.875	7%
02	11.0	-5.0	5%
03	11.0	-5.0	10%
Whole data	11.0	-5.0	7.224%

Even though we only perform the random sample 3 times, the average outlier ratio is $(7+5+10)/3 \approx 7.3$, which is very close to the result on whole data (7.224%). In 2 out of 3 queries, the upper limit and lower limit for non-outliers are the same as those on the whole data.

Problem 02:

I would like to perform the procedure above on DVD and Books, trying to find out if the randomly sampling also produce a correct estimation of the outlier ratio.

Query:

```
SELECT
    count(cout) as checkouts,
    bibNumber,
    title
FROM spl_2016.inraw
where
    itemtype in (
        'cadvdnf',
        'ncdvdnf',
        'nalndvd',
        'jrdvd',
        'jcdvd',
        'acdvd',
        'nadvd',
        'ardvd',
        'ccdvdnf',
        'ncdvd',
        'nadvdnf'
    )
    and cout > '2022-01-01'
group by bibNumber, title
order by rand(42)
LIMIT 100;
```

```
24
25  SELECT
26      count(cout) as checkouts,
27      bibNumber,
28      title
29  FROM spl_2016.inraw
30  where
31      itemtype in (
32          'cabknf',
33          'ncbknf',
34          'nalnbk',
35          'jrbk',
36          'jcbk',
37          'acbk',
38          'nabk',
39          'arbk',
40          'ccbknf',
41          'ncbk',
42          'nabknf',
43          'jcbknf'
44      )
45      and cout > '2022-01-01'
46  group by bibNumber, title
47  order by rand(44)
48  LIMIT 100;
```

RESULT:

> See attached DVD_100_#.csv / Book_100_#.csvfor results.

Sample Times	Upper Limit for Non-outlier	Lower Limit for Non-outliers	Outlier Ratio
Book_100_01	16.0	-8.0	10%
Book_100_02	20.125	-8.875	14%
Book_100_03	18.5	-9.5	14%
Book_all	18.5	-9.5	11.9%
DVD_100_01	38.0	-18.0	5%
DVD_100_02	41.75	-20.25	5%
DVD_100_03	59.625	-29.375	4%
DVD_all	44.5	-22.5	8.9%

The data on books performed similarity to CDs. The randomly sampled average outlier ratio is approximately the same as the overall outlier ratio.

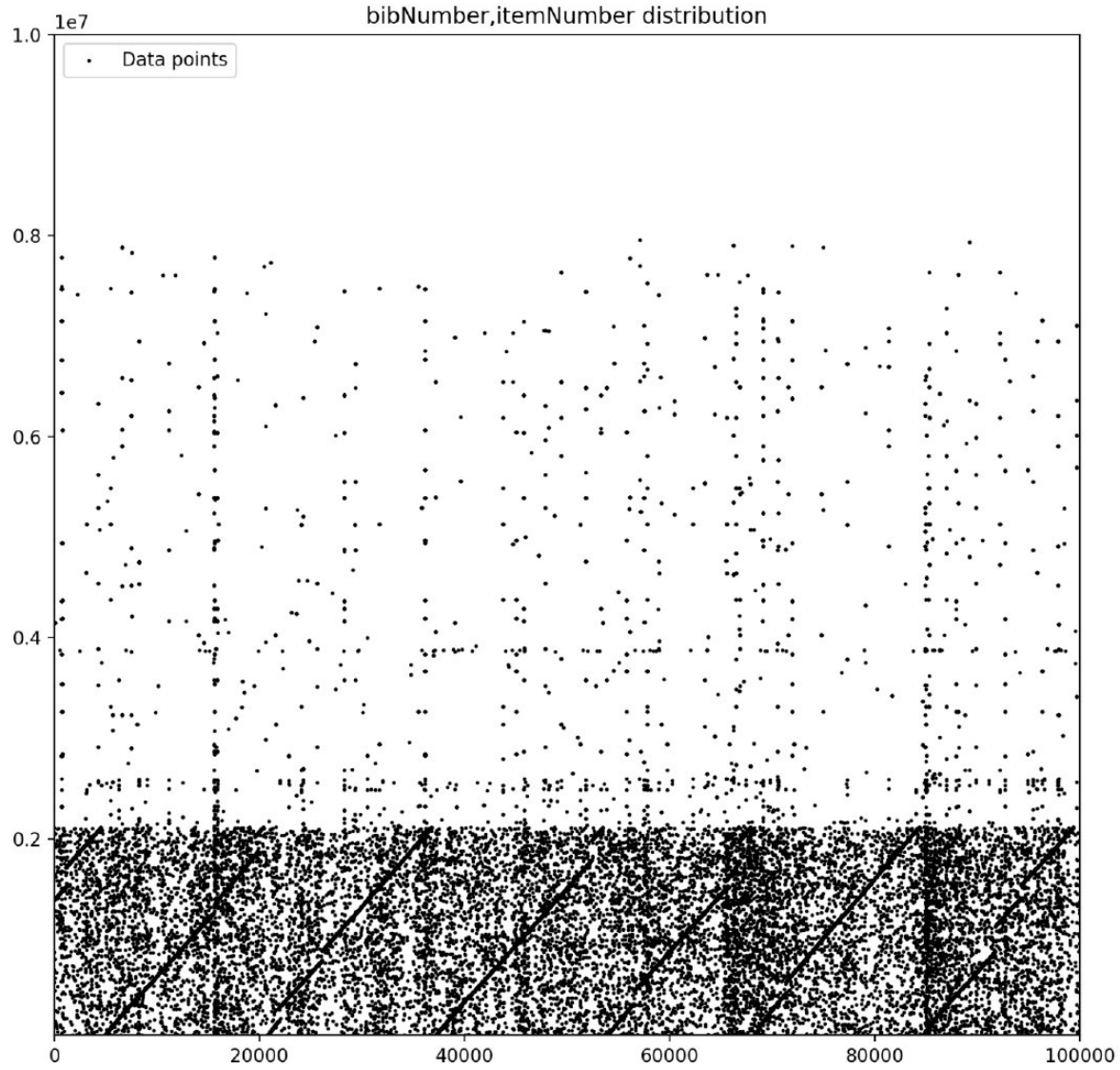
$(38\%/3 \approx 12.7)$

However, this doesn't apply to the DVD category. For the DVD category, the randomly sampled average outlier ratio is 4.6%, whereas the overall outlier ratio is 8.9%, nearly 2 times difference. This is can be the sampling error or the dataset for DVD category is skewed.

If we compare the limits for each media types, we find that DVD has a higher limit, then it is Book and then CD. The limits difference indicate the deviation of the items' checkout times. In this case, DVD has a much higher deviation than the rest category.

Question 03:

From last week's assignment, there is a linear pattern in the distribution of itemNumber and bibNumber. (As shown in the graph below)



If we randomly sampled some datapoints within the same range of bibNumber, will the sampled data follow the same pattern or produce a different distribution pattern?

Query:

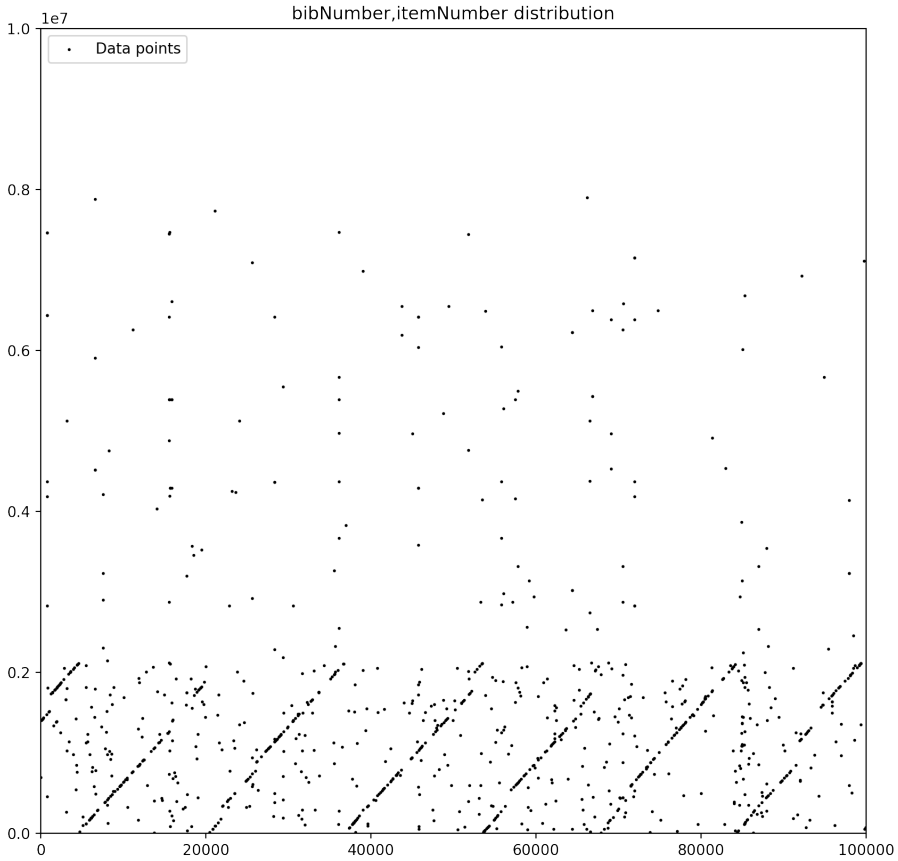
```
select
    distinct bibNumber,
    itemNumber
from spl_2016.inraw
where `bibNumber` < 100000
order by rand(44),bibNumber
LIMIT 1000;
```

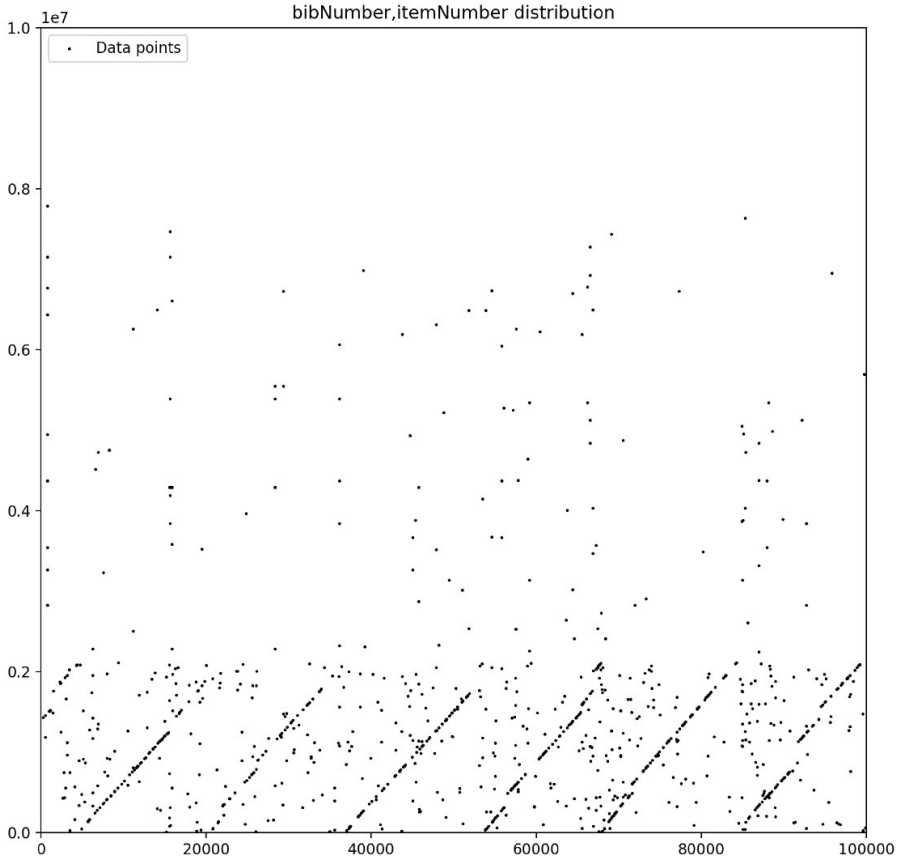
> See bibDist_#.csv for results

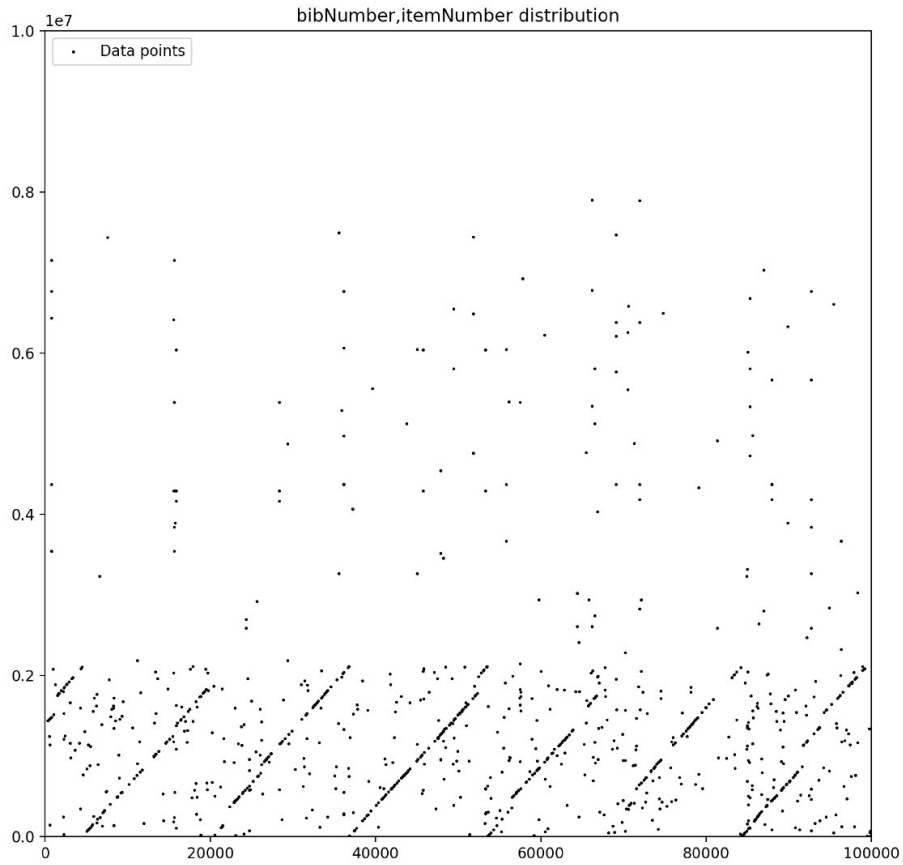
The random sampling is performed 3 times. Each time the data will be feed to below visualization code to generate the distribution of the bibNumber and itemNumber:

```
arr_dist = np.array(df_dist[['bibNumber','itemNumber']])
figure(figsize=(10, 10), dpi=200)
# Generate train data
X = arr_dist

plt.title("bibNumber,itemNumber distribution")
plt.scatter(X[:, 0], X[:, 1], color="k", s=1.0, label="Data points")

plt.axis("tight")
plt.xlim((0, 100000))
plt.ylim((50000, 10000000))
legend = plt.legend(loc="upper left")
# legend.legendHandles[0]._sizes = [10]
# legend.legendHandles[1]._sizes = [20]
plt.show()
plt.savefig('img.png')
```

Result:


bibNumber,itemNumber distribution
1e7
Data points
1.0
0.8
0.6
0.4
0.2
0.0
0
20000
40000
60000
80000
100000


bibNumber,itemNumber distribution
1e7
Data points
1.0
0.8
0.6
0.4
0.2
0.0
0
20000
40000
60000
80000
100000



The randomly sampled results are similar to the overall figure, 6 lines all appears in the 3 graphs and they share similar starting points and period, which means some of the itemNumbers start over after 18000 bibNumbers. The linear pattern is much more evident in the random sampled datapoints than the overall one.

Conclusion:

Comparing the data between random sampling and the whole data, here's some of my observations.

- The random sampled results are similar to the overall dataset on CD and Books, whereas for DVD, this doesn't hold. This may indicate that DVD checkout has a more uneven distribution.
- The result on the overall dataset is approximately the average of the randomly sampled results.
- For the bibNumber / itemNumber distribution, the random sample result follows the similar pattern from the overall dataset. But the pattern is more prominent on the sampled result.