# Analyzing Big Data- Final Project

MSBA-Menghong Han

**Assumptions:**

1) We have used the data for the year 2004 and the results reflect analysis only of that year.
2) We have assumed Low income group from 3-15, Medium Income group from 16-21, High Income group from 22-27.

**Question (a)**

**How many:**

1) **Store shopping trips are recorded in your database?**

7501302

2) **Households appear in your database?**

39577

3) **Stores of different retailers appear in our data base?**

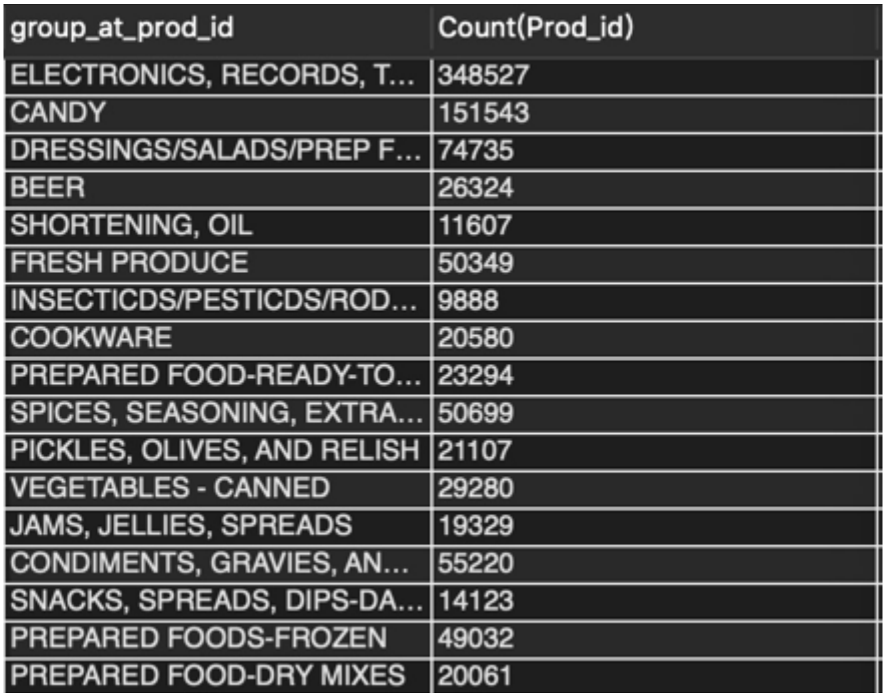
860

4) **Different products are recorded?**

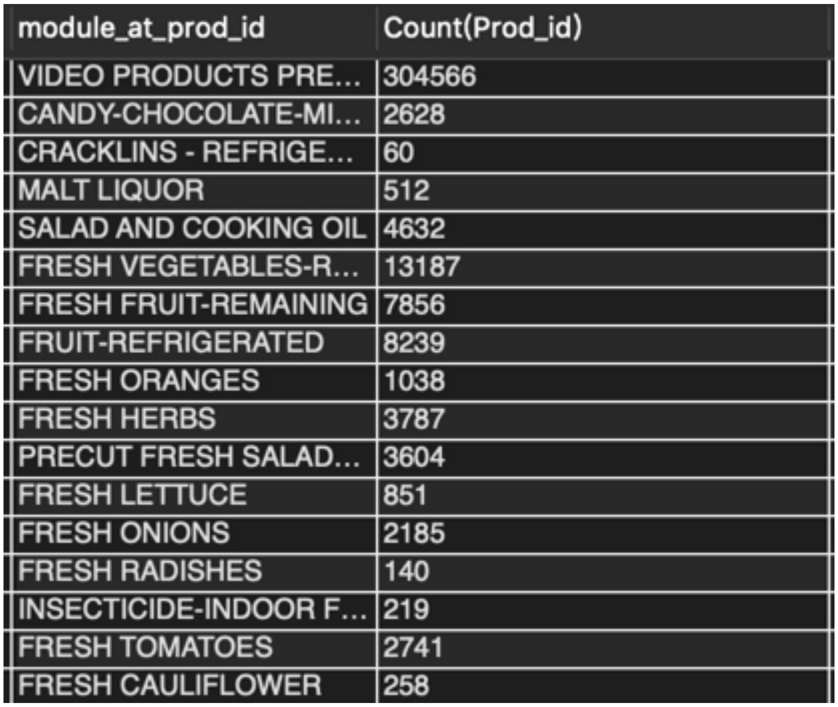
   i) **Products per category and products per module**

   Per Category- 119 rows returned

   Per Module- 1225 rows returned

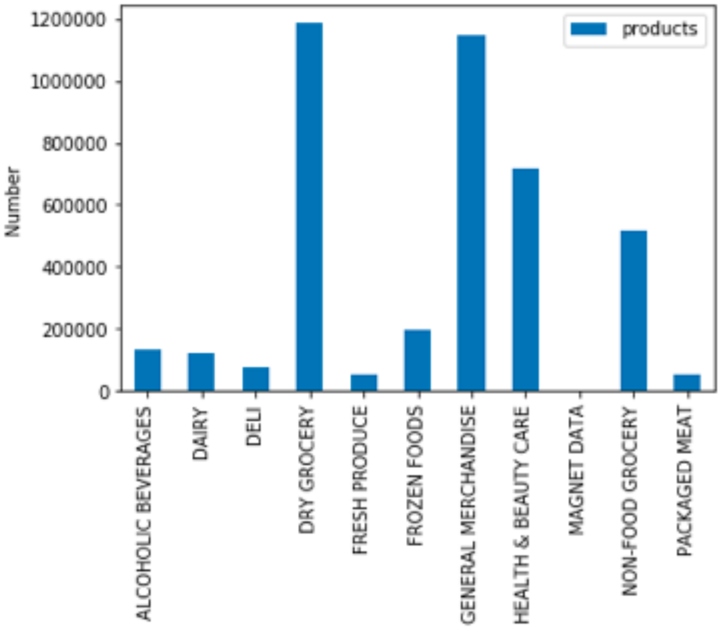


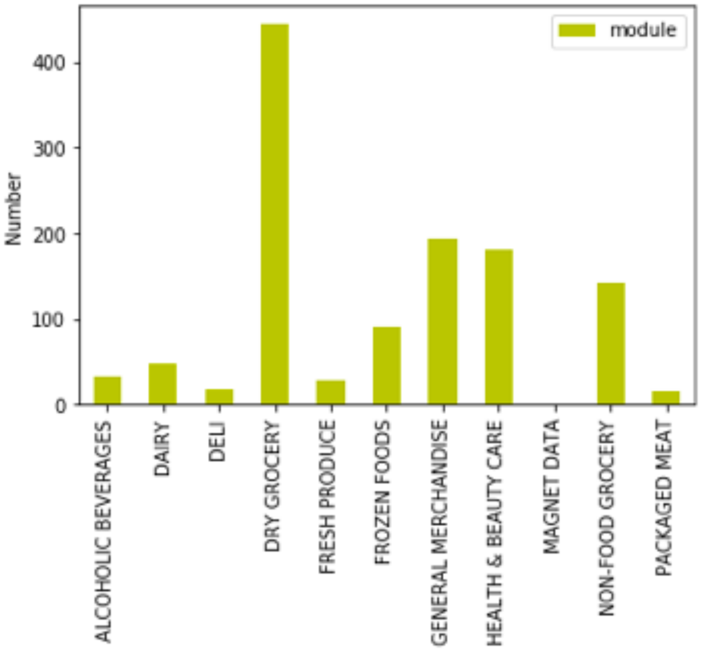
| group_at_prod_id | Count(Prod_id) |
|---|---|
| ELECTRONICS, RECORDS, T… | 348527 |
| CANDY | 151543 |
| DRESSINGS/SALADS/PREP F… | 74735 |
| BEER | 26324 |
| SHORTENING, OIL | 11607 |
| FRESH PRODUCE | 50349 |
| INSECTICDS/PESTICDS/ROD… | 9888 |
| COOKWARE | 20580 |
| PREPARED FOOD-READY-TO… | 23294 |
| SPICES, SEASONING, EXTRA… | 50699 |
| PICKLES, OLIVES, AND RELISH | 21107 |
| VEGETABLES - CANNED | 29280 |
| JAMS, JELLIES, SPREADS | 19329 |
| CONDIMENTS, GRAVIES, AN… | 55220 |
| SNACKS, SPREADS, DIPS-DA… | 14123 |
| PREPARED FOODS-FROZEN | 49032 |
| PREPARED FOOD-DRY MIXES | 20061 |



| module_at_prod_id | Count(Prod_id) |
|---|---|
| VIDEO PRODUCTS PRE… | 304566 |
| CANDY-CHOCOLATE-MI… | 2628 |
| CRACKLINS - REFRIGE… | 60 |
| MALT LIQUOR | 512 |
| SALAD AND COOKING OIL | 4632 |
| FRESH VEGETABLES-R… | 13187 |
| FRESH FRUIT-REMAINING | 7856 |
| FRUIT-REFRIGERATED | 8239 |
| FRESH ORANGES | 1038 |
| FRESH HERBS | 3787 |
| PRECUT FRESH SALAD… | 3604 |
| FRESH LETTUCE | 851 |
| FRESH ONIONS | 2185 |
| FRESH RADISHES | 140 |
| INSECTICIDE-INDOOR F… | 219 |
| FRESH TOMATOES | 2741 |
| FRESH CAULIFLOWER | 258 |

**ii) Plot the distribution of products and modules per department**





### 5) Transactions?

**i) Total transactions and transactions realized under some kind of promotion.**

Total transaction: 38587942

Transactions realized under a promotion: 11384077

**Question (b)**

1) **How many households do not shop at least once on a 3 month periods?**

There are 32 households who have not been recorded shopping for 3 months

**i) Is it reasonable?**

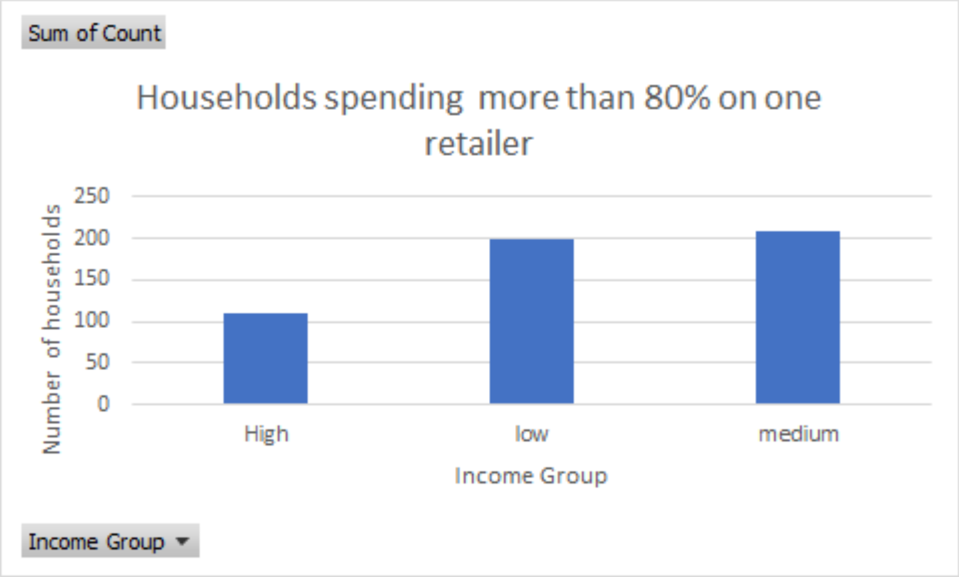
It is unreasonable that any household could go more than 3 months without needing to shop. However, compared to the 390000 total households, this very small number of households is reasonable as these households could have only been recorded on one shopping trip for the period for which we have data. This could be for any number of plausible reasons.

**ii) Why do you think this is occurring?**

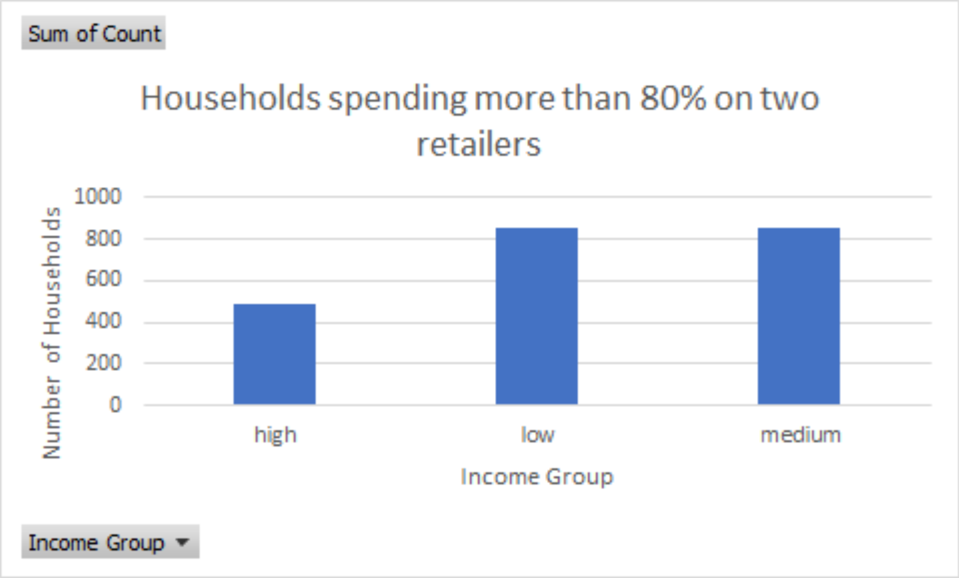
There are several plausible reasons why some households will only have been recorded having one shopping trip. This could be because they shopped for the final time within the first three months of the data's period. They could move out of the country, or die, or change household for example if an elderly person moved into their children's home or a care home. Alternatively, a family could move into the country towards the end of the period for which we have data and therefore not recorded as shopping more than once in a 3 month period.

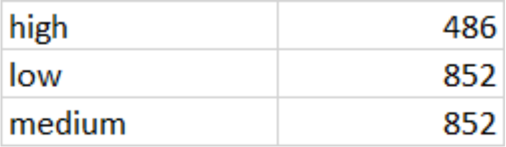
2) **Loyalism: Among the households who shop at least once a month, which % of them concentrate at least 80% of their grocery expenditure (on average) on single retailer? And among 2 retailers?**

**i) Are their demographics remarkably different? Are these people richer? Poorer?**



111 high income  (>$60k)  out of a total of 12805 (0.867%) spend more than 80% of their expenditure in one retailer. This is a lower fraction than the 1.345% and 1.764% of medium ($30 to $59.9k) and low income (<$30k) households respectively that spend more than 80% of their expenditure on one retail. This could be surprising if one thinks that lower income households would be more willing to 'shop around' different retailers to get the best price for the same products. But it could be explained by lower income households having less access to a variety of retailers.





| | |
|---|---|
| high | 486 |
| low | 852 |
| medium | 852 |

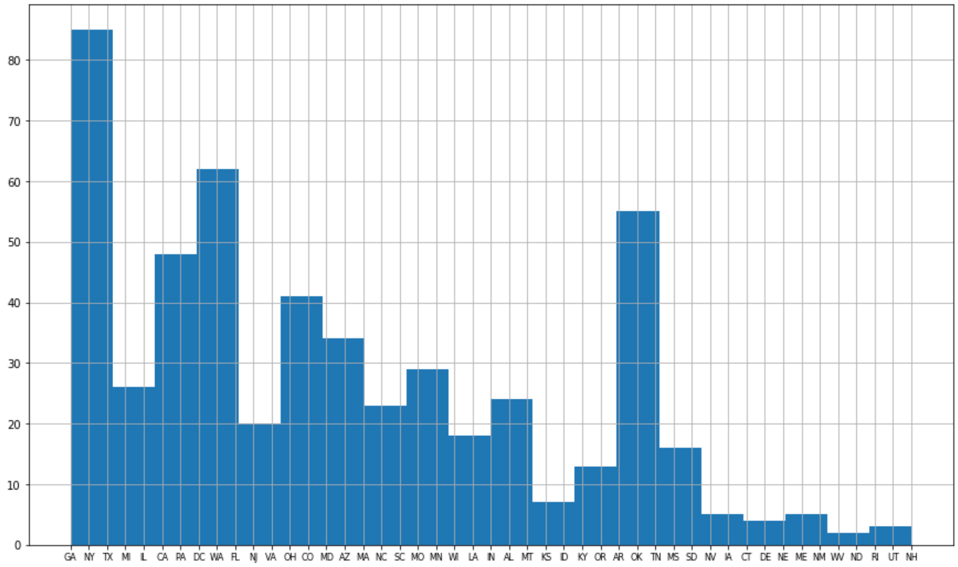
Again following the above pattern, the lowest fraction of households spending more than 80% on expenditure on two retailers is the high income group (3.795%). The medium and low income groups have 5.521% and 7.513% respectively spending more than 80% of expenditure on two retailers. The explanation should be similar to above and this consistency increases lends confidence to the findings.

### ii) What is the retailer that has more loyalists?

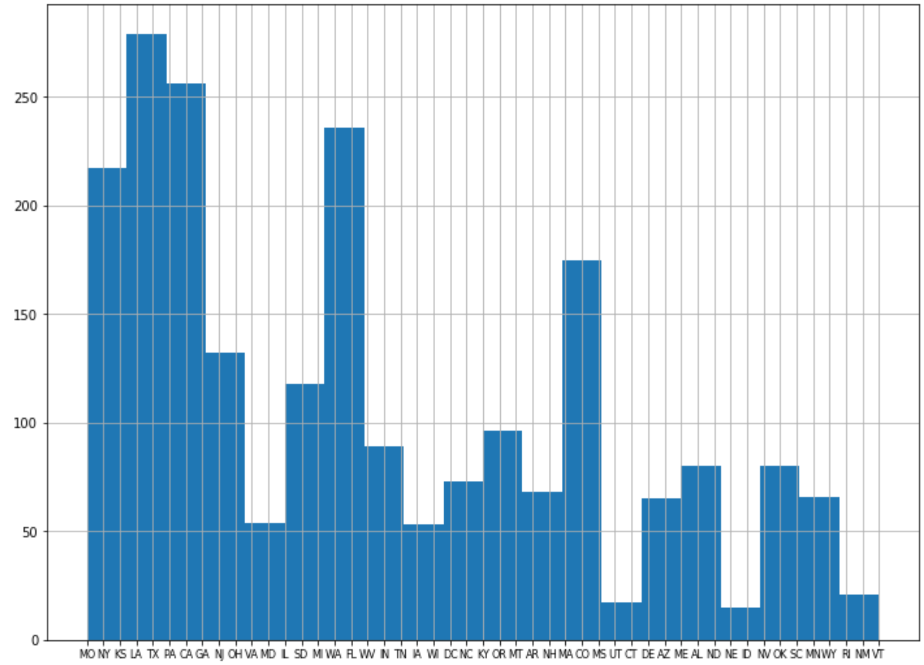
The top five retailers with more loyalists are: 6920, 181, 239, 32 , 42

### iii) Where do they live? Plot the distribution by state.

Households distribution for a single retailer:

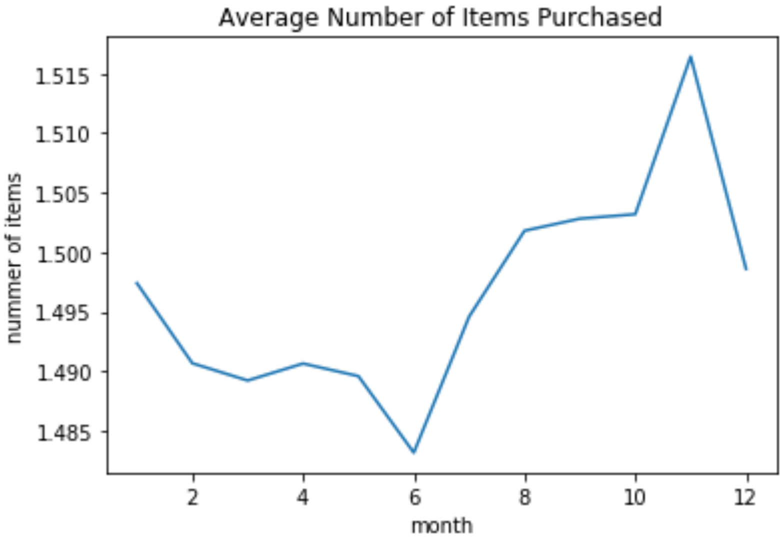


Households distribution for two retailers:

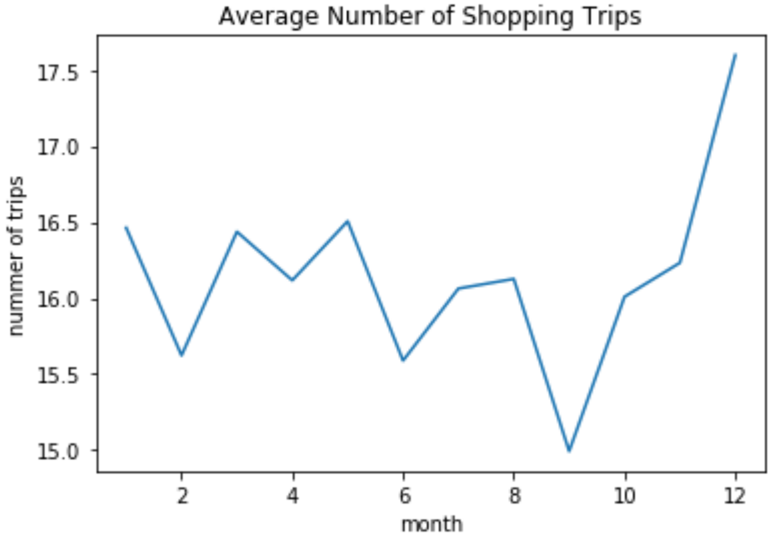


**3) Plot with the distributions:**

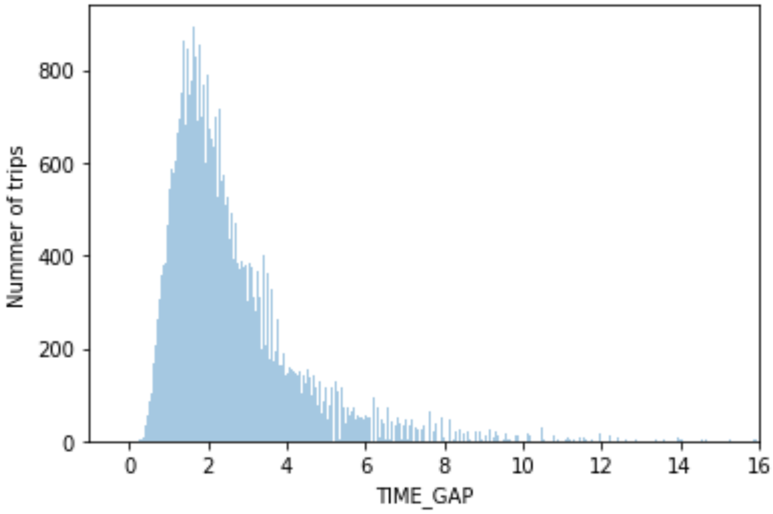
i) **Average number of items purchased in a given month.**



**ii) Average number of shopping trips per month.**



**iii) Average number of days between 2 consecutive shopping trips.**

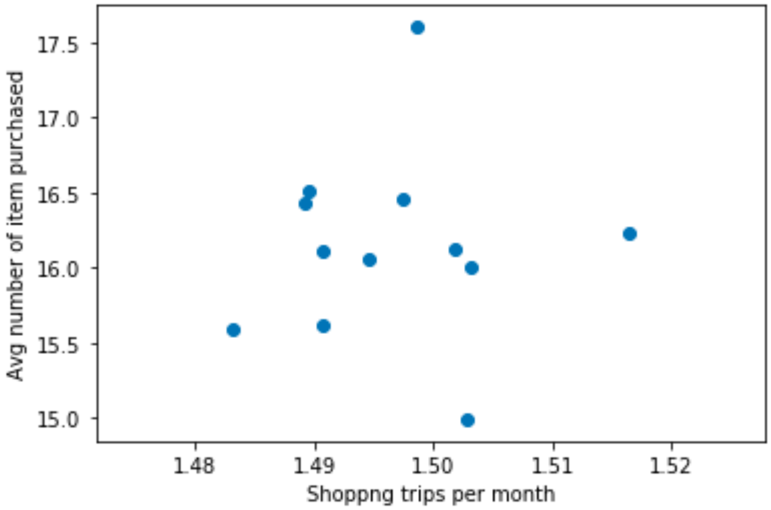


**Question (c)**

**Answer and reason the following questions: (Make informative visualizations)**

### 1) Is the number of shopping trips per month correlated with the average number of items purchased?

Please note, we have interpreted this question as asking whether the average number of shopping trips per month is correlated with the average number of items purchased on each trip.

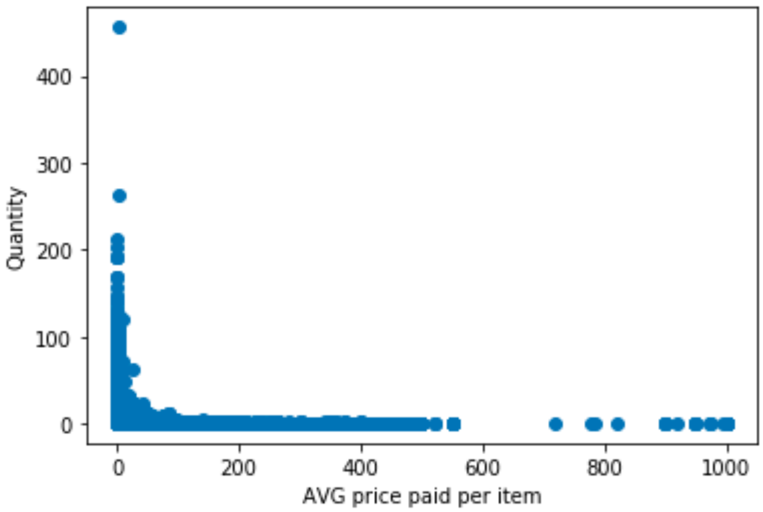


The correlation coefficient between the number of items bought and the average number of shopping trips per month is 0.04818. This correlation is small, meaning that on average the number of items bought increases very slightly with more shopping trips. This is a little counter-intuitive if one thinks that fewer shopping trips would mean that more items are bought on each trip, so that the shopping supplies purchased would last longer until the next shopping trip. It is still plausible if we think that more shopping trips occur in busier months, like holiday periods, in which people just buy more things, then the number of items bought can increase even as the number of trips increases, but this effect is minimal, given the very small correlation. The low outlier of under 15 items purchased is September and the high outlier of over 17.5 items purchased on average is December.

### 2) Is the average price paid per item correlated with the number of items purchased?

38587942 rows returned

The correlation is -0.0903. This is to be expected from looking at the graph because all of the trips with higher average price per item are very low quantities. It is also highly reasonable as the shopping trips with higher average price per item are likely to be for one off purchases of high value items like electronics.



3) **Private Labeled products are the products with the same brand as the supermarket. In the data set they appear labeled as ‘CTL BR’**

**i). What are the product categories that have proven to be more “Private labelled”**

113 rows returned

| group_at_prod_id | count_pl | count_all | count_pl/count_all |
|---|---:|---:|---:|
| DISPOSABLE DIAPERS | 8693 | 13723 | 0.6335 |
| VEGETABLES-FROZEN | 13549 | 21487 | 0.6306 |
| COUGH AND COLD REMEDIES | 15429 | 25068 | 0.6155 |
| DOUGH PRODUCTS | 2771 | 4555 | 0.6083 |
| JUICES, DRINKS-FROZEN | 1359 | 2297 | 0.5916 |

| | | | |
|---|---|---|---|
| **WRAPPING MATERIALS AND BAGS** | 11979 | 20278 | 0.5907 |
| **FIRST AID** | 12617 | 22698 | 0.5559 |
| **FRUIT - CANNED** | 7012 | 13006 | 0.5391 |
| **BREAD AND BAKED GOODS** | 84716 | 157793 | 0.5369 |
| **CEREAL** | 13119 | 24507 | 0.5353 |
| **VEGETABLES - CANNED** | 15256 | 29280 | 0.521 |
| **BREAKFAST FOODS-FROZEN** | 4330 | 8314 | 0.5208 |
| **SUGAR, SWEETENERS** | 2246 | 4442 | 0.5056 |

Here are the top thirteen product groups by the proportion of private label brands to the total number of brands for each product group. These are the only product groups, from 113 different groups, with over 50% of the products being private label brands.

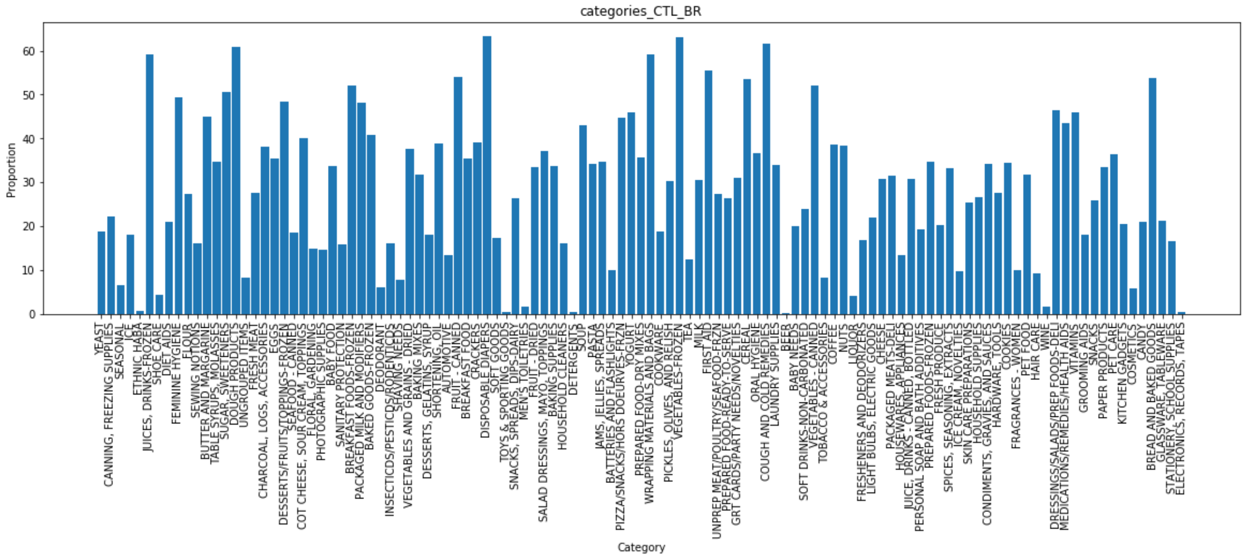
| group_at_prod_id | count_pl | count_all | count_pl/count_all |
|---|---|---|---|
| **BEER** | 56 | 26324 | 0.0021 |
| **TOYS & SPORTING GOODS** | 49 | 14012 | 0.0035 |

| | | | |
|---|---|---|---|
| **ELECTRONICS, RECORDS, TAPES** | 1333 | 348526 | 0.0038 |
| **DETERGENTS** | 78 | 17242 | 0.0045 |
| **ETHNIC HABA** | 11 | 1984 | 0.0055 |
| **WINE** | 1149 | 69421 | 0.0166 |
| **MEN'S TOILETRIES** | 265 | 15149 | 0.0175 |
| **LIQUOR** | 1395 | 34916 | 0.04 |
| **SHOE CARE** | 107 | 2405 | 0.0445 |
| **COSMETICS** | 8011 | 140017 | 0.0572 |
| **DEODORANT** | 598 | 9749 | 0.0613 |

Above are the product groups with the lowest proportion of private label brands.

The product groups with a high proportion of private label brands are lower-value, basic products like canned/frozen foods, bread, cereals, and dough products, which are bought regularly and in high volumes. There are generic products like first aid, cold remedies, diapers, wrappings materials and sugar, for which brand-power and marketing campaigns are less important.

The product groups with the lowest proportion of private label brands are mainly luxury goods like alcohol, cosmetics, electronics and shoe care. The premium prices of these products mean that there is more incentive for companies to produce these products and spent money on building brand-power through marketing campaigns. Similarly, these products, like cosmetics, beer and wine, and toys tend to be less generic and will respond well to marketing campaigns.



**ii) Is the expenditure share in Private Labeled products constant across months?**

12 rows returned

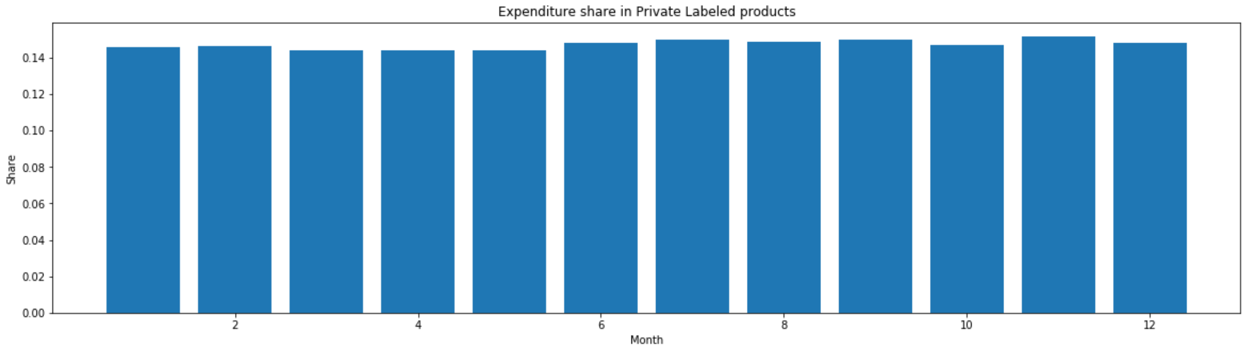
The proportion of expenditure on private label products to all products varies between 14.4% and 15.2% across months, with the 14.4% low occurring consistently in March, April and May and the 15.2% high occurring in November, the only month in which the proportion exceeds 15%. The standard deviation of the share of expenditure on private label products across months is 0.276%.

It is difficult to say if this is systematic variation across or just random variation. A case could be made for people increasing their expenditure share of better value, private label products towards the end of the year as budgets are stretched further around summer holidays and then Thanksgiving and Christmas holidays.

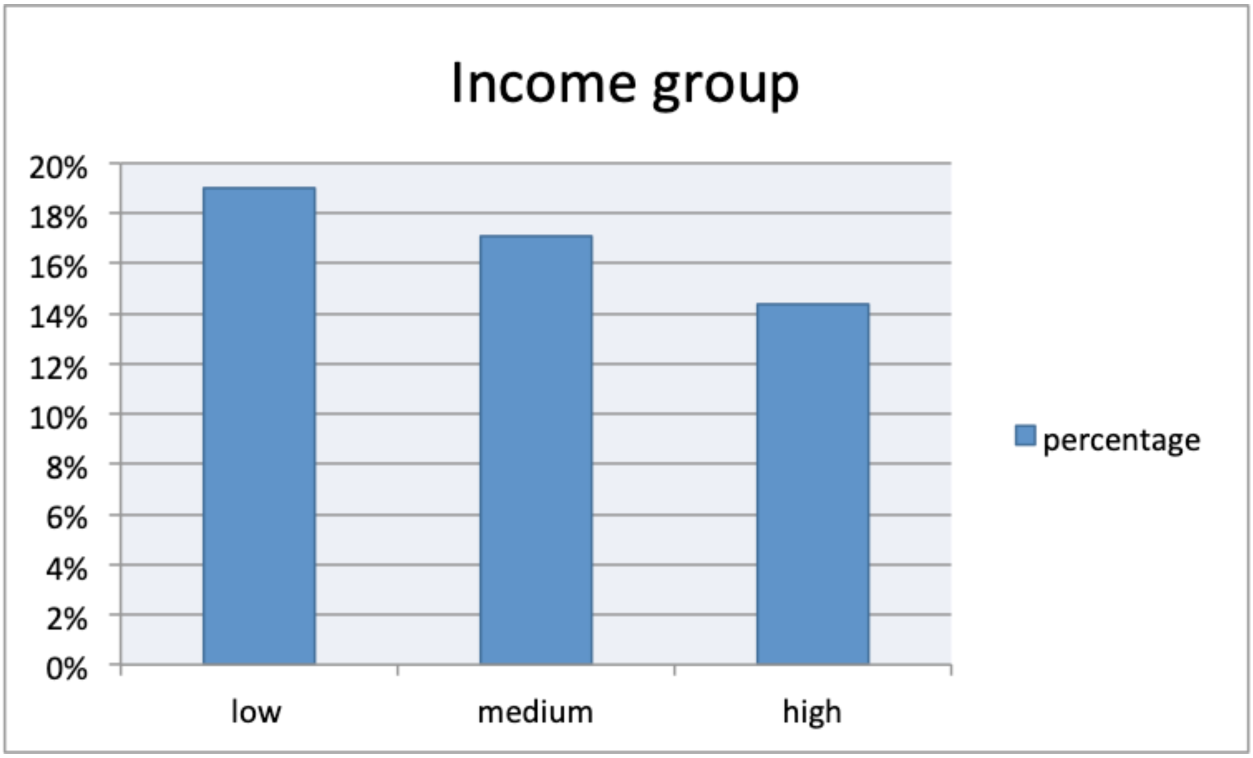
| pl_prods | all_prods | month | pl_prods/all_prods |
|---|---|---|---|
| **1648900.179997260** | 11288459.09010790 | 1 | 14.607% |
| **1563416.16999763** | 10674294.050103400 | 2 | 14.647% |

| | | | |
|---|---|---|---|
| **1607488.7599974200** | 11149783.340106100 | 3 | 14.417% |
| **1576996.3799974700** | 10940467.430109900 | 4 | 14.414% |
| **1648352.1399974700** | 11437012.420107200 | 5 | 14.412% |
| **1559165.7199975400** | 10541174.420102600 | 6 | 14.791% |
| **1643114.609997420** | 10964126.740104700 | 7 | 14.986% |
| **1669861.129997210** | 11236454.480105000 | 8 | 14.861% |
| **1570918.1899975900** | 10478079.200101800 | 9 | 14.992% |
| **1670569.9899972300** | 11342733.4901081 | 10 | 14.728% |
| **1673163.7299972000** | 11025383.97010670 | 11 | 15.176% |
| **1684078.509997170** | 11421330.150106400 | 12 | 14.745% |

| | |
|---|---|
| Standard deviation | 0.2476% |



**iii) Cluster households in three income groups, Low, Medium and High. Report the average monthly expenditure on grocery. Study the % of private label share in their monthly expenditures. Use visuals to represent the intuition you are suggesting.**



The low income group spends a higher proportion of their total expenditure on private label products (19%) as expected. The medium income group spends the next highest share of their expenditure on private label products (17%) and the highest income group spends the lowest portion (14%). This is to be expected because private label products tend to be cheaper than branded products and lower income households have tighter budgets.

## Code:

## Use Python to import the large scale data:

```
import numpy as np
import pymysql
import pandas as pd
from sqlalchemy import create_engine
db = pymysql.connect(host='localhost',
user='root',
password='3p2a1o2333',
db='db_consumer_panel2',
charset='utf8mb4',
cursorclass=pymysql.cursors.DictCursor)
engine = create_engine('mysql+pymysql://root:3p2a1o2333@localhost/db_consumer_panel2')


dta_at_hh = pd.read_csv("/Users/tmh/Desktop/bdm/dta_at_hh.csv")
b1=dta_at_hh.drop('Unnamed: 0', axis=1)
filter=[2070841,8034573,8089790,8127731,8210330,8215790,8236635,8278811,9001556,9156682]
b1_sample=b1[b1['hh_id'].isin(filter)]
pd.io.sql.to_sql(b1_sample,'households', con=engine, index=False, if_exists='append')

dta_at_prod_id = pd.read_csv("/Users/tmh/Desktop/bdm/dta_at_prod_id.csv")
b2=dta_at_prod_id.drop('Unnamed: 0', axis=1)
filter_prod=[100001087957081,100003814221599,100003886102970,100005830621411,100007500102178,100008165210077,100071722680513,100073772952220,100088392936336,100967582824162]
b2_sample=b2[b2['prod_id'].isin(filter_prod)]
pd.io.sql.to_sql(b2_sample,'products', con=engine, index=False, if_exists='append')

dta_at_TC = pd.read_csv("/Users/tmh/Desktop/bdm/dta_at_TC.csv")
b3=dta_at_TC.drop('Unnamed: 0', axis=1)
filter=[2070841,8034573,8089790,8127731,8210330,8215790,8236635,8278811,9001556,9156682]
b3_sample=b3[b3['hh_id'].isin(filter)]
```

```
pd.io.sql.to_sql(b3_sample,'trips', con=engine, index=False, if_exists='append')
```

```
dta_at_tc_upc = pd.read_csv("/Users/tmh/Desktop/bdm/dta_at_tc_upc.csv")
b4=dta_at_tc_upc.drop('Unnamed: 0', axis=1)
filter_tcid=[13848,19929,31768,32754,41828,41829,41830,51028,51222,51223,52311,52312,6
2378,72483,73647,80249,80250,88536,100492,100493,104546,105627,115488,120983,12661
0,126611,140191,140192,140193,140194,142370,142371,151450,151451,151452,151453,157
640,157641,165629,165630,166556,166557,174944,182301,182302,182303,183148,183149,1
87004,199511,203410,214996,215164,215165,216101,221087,233194,233195,233196,233197
,242742,242743,250119,254231,280299,280300,286845,286846,286847,294222,294223,3079
81,311361,311362,312255,320305,320306,328424,331920,340934,342820,346501,354683,35
6489,360055,361045,370681,382267,388984,392777,403654,403655,407176,419353,421595,
422551,438568,439427,441910,441911,447486,454758,463761,463762,466278,471033,47208
8,480405,483243,489651,489652,489653,493925,499393,503102,515158,515159,550586,550
587,555421,555422,564969,564970,573094,573095,573096,580661,584469,588360,595660,5
99318,608178,610080,610081,614776,614777,633412,636525,636526,637921,637922,645405
,661270,665024,665025,674088,678711,678712,683491,684601,695785,704032,711614,7118
06,711807,711808,711809,711810,723640,723641,723642,729474,744883,755285,755286,75
8779,761058,762784,762785,773655,777794,777795,788345,793543,793544,797157,802938,
811480,819705,824713,828176,829688,834029,834030,853267,853268,858037,859127,85912
8,864406,864407,864408,872259,873651,875910,879774,889919,919610,919611,919612,929
728,932549,937916,951287,954540,954541,954542,954543,955417,960500,963410,986638,9
87681,1003901,1004849,1004850,1014334,1019994,1019995,1021635,1021636,1025624,105
1286,1051287,1074290,1081535,1081536,1082342,1082343,1090258,1098283,1098284,1098
285,1106227,1108791,1108792,1113343,1114411,1118388,1119492,1128557,1130815,11308
16,1141719,1141720,1145251,1145252,1149334,1152218,1153534,1168373,1176304,117993
4,1186842,1195485,1211532,1212464,1212465,1220860,1223465,1228396,1229312,1229313,
1237703,1237704,1237705,1237706,1237707,1245327,1246455,1246456,1248950,1257709,1
257710,1264050,1269675,1279708,1279709,1285356,1292119,1306401,1307657,1307658,13
18024,1325406,1326249,1326250,1326251,1326252,1326253,1334448,1338703,1338704,134
2911,1351042,1351043,1351044,1353549,1353550,1359140,1359141,1359142,1361567,1374
701,1374889,1390161,1393341,1397549,1406172,1406173,1406174,1408350,1410053,14100
54,1417212,1420480,1420481,1424584,1430313,1434652,1435819,1439294,1445906,144864
6,1453661,1454657,1454658,1467127,1467128,1470287,1471132,1473638,1479107,1486056,
1486057,1502027,1511491,1511492,1511493,1514146,1514147,1519477,1519478,1524795,1
535390,1561507,1562735,1562736,1562737,1562738,1569828,1575628,1575629,1581517,15
81518,1597463,1597464,1597465,1600852,1603548,1603549,1606488,1613746,1614559,161
7012,1617013,1622571,1629518,1630537,1634405,1634406,1647591,1647592,1664085,1664
086,1671120,1671121,1689789,1702422,1713526,1713527,1716403,1723980,1723981,17300
00,1733131,1757371,1757530,1757531,1757532,1758433,1758434,1786420,1786421,179386
4,1797890,1812639,1812640,1814989,1814990,1828556,1828557,1831678,1831679,1834468,
1837691,1840883,1840884,1857359,1858268,1860782,1862418,1866773,1866774,1866775,1
866776,1869565,1874848,1874849,1878465,1892481,1892669,1899108,1902897,1908477,19
```

```
12761,1913820,1920569,1928780,1928781,1937593,1937594,1937595,1941429,1941430,194
1431,1945189,1948082,1972166,1979885,1984157,1987121,1995799,1998181,2000856,2003
691,2003692,2018779,2018954,2024733,2030438,2032237,2055125,2066522,2076483,20777
43,2096732,2099189,2101922,2101923,2104890,2104891,2113301,2118575,2124132,212864
9,2129444,2144564,2145509,2149549,2153753,2162742,2165513,2165514,2165515,2179368,
2183046,2193466,2193467,2199103,2207814,2207815,2207816,2216956,2216957,2221749,2
221750,2221751,2227952,2238082,2238904,2238905,2238906,2243940,2247009,2247010,22
60105,2262943,2262944,2265544,2269991,2281962,2287179,2287180,2287181,2287182,228
8146,2292521,2300856,2300857,2318463,2318464,2329871,2347738,2347925,2348798,2351
492,2351493,2351494,2351495,2357444,2361934,2366073,2366074,2374369,2376903,23769
04,2376905,2382494,2382495,2397867,2414760,2420213,2434923,2439196,2439197,243919
8,2439199,2456693,2468565,2468566,2473970,2489691,2502590,2504459,2515595,2520134,
2531971,2531972,2552504,2597889,2597890,2600375,2600376,2600377,2606145,2608639,2
608640,2616742,2624802,2630212,2638127,2638128,2645103,2645282,2645283,2648457,26
48458,2675554,2675555,2679030,2679031,2698587,2702684,2713870,2713871,2713872,271
6396,2716397,2753601,2753602,2754405,2754406,2758520,2769836,2788659,2791477,2791
478,2793267,2805038,2807686,2808867,2818890,2818891,2822261,2822262,2822263,28284
46,2829605,2829606,2839170,2841747,2847330,2857639,2859354,2862548,2862549,286255
0,2875745,2888062,2893666,2893667,2895502,2895503,2896624,2896625,2896626,2896627,
2916636,2916637,2918331,2942834,2942835,2942836,2942837,2953001,2954173,2971537,2
980492,2987583,2988380,3003775,3006007,3011619,3011620,3011621,3011622,3016978,30
18798,3019816,3033542,3054148,3055184,3061233,3081904,3093564,3093565,3094382,309
4383,3096688,3104900,3104901,3104902,3104903,3118760,3126921,3126922,3132095,3134
963,3142437,3154394,3154395,3160574,3163161,3163162,3169863,3179450,3179451,31962
83,3201413,3213814,3218387,3218388,3219215,3224160,3227060,3227061,3235444,324334
8,3243349,3243350,3246182,3251513,3253791,3255462,3270169,3282065,3282066,3284241,
3284242,3294317,3294318,3294319,3310612,3324365,3324366,3324367,3331065,3338371,3
338372,3338373,3344156,3346497,3349355,3355083,3355084,3360100,3360941,3369219,33
69220,3372118,3377825,3377826,3380232,3394469,3411523,3413931,3414947,3414948,341
7816,3435149,3435150,3435151,3435152,3449601,3467712,3467713,3467714,3467715,3467
716,3481138,3481139,3488918,3489823,3500617,3500618,3506151,3506152,3508513,35102
02,3514248,3521815,3521979,3522814,3530892,3539117,3539118,3539119,3565123,358091
0,3586733,3601959,3601960,3610433,3610434,3613102,3615010,3615800,3626610,3631175,
3631176,3639862,3639863,3645822,3647569,3665155,3665156,3666141,3671575,3679247,3
686186,3696132,3702997,3707279,3708450,3708451,3718171,3721263,3721264,3729110,37
34943,3743557,3751321,3754536,3758997,3758998,3764825,3770898,3775537,3776524,377
6525,3793225,3813068,3813069,3814146,3828894,3843451,3849616,3857664,3857665,3857
666,3857667,3866806,3872384,3876381,3886252,3888794,3900062,3900063,3904703,39048
55,3904856,3910836,3917669,3920782,3920783,3923054,3923055,3945099,3945100,395545
3,3955454,3955456,3979001,3979002,3979003,3982613,3988436,3988437,3990790,3992578,
4000274,4001983,4001984,4005288,4006151,4014377,4014378,4035563,4035564,4040103,4
040265,4040266,4041079,4041080,4045144,4046237,4059437,4064980,4080142,4103509,41
06726,4117469,4122265,4123138,4128422,4134833,4136435,4139655,4139656,4148502,415
2029,4154791,4159917,4159918,4159919,4161701,4162790,4162791,4165611,4169494,4174
```

124,4175079,4183543,4193315,4210541,4210542,4210543,4212941,4236154,4236155,42473
86,4251183,4256688,4256689,4256690,4262049,4264836,4264837,4268568,4274172,427646
6,4285977,4290636,4290637,4303254,4307758,4307913,4308727,4316862,4325845,4326731,
4340573,4346279,4346280,4347319,4347320,4365558,4365559,4374154,4374155,4376929,4
381102,4385028,4392942,4392943,4407151,4407152,4412961,4415804,4424981,4432949,44
32950,4441477,4442411,4447471,4450512,4454748,4454749,4454750,4465865,4480302,448
0519,4480520,4504388,4516026,4519923,4524767,4525547,4527719,4543414,4555676,4555
677,4560602,4560603,4563914,4569852,4569853,4578933,4585332,4593200,4598740,45987
41,4599696,4614257,4617632,4617633,4619986,4620999,4627317,4639017,4642744,466204
3,4662793,4674578,4679210,4687939,4691705,4696177,4699246,4704843,4708885,4711658,
4713447,4713627,4714439,4722389,4726548,4732402,4732403,4732404,4732405,4752051,4
752052,4769174,4769175,4795501,4798534,4798535,4807849,4812439,4821149,4829697,48
34466,4838233,4838234,4846462,4847325,4847326,4847327,4859282,4867398,4882482,488
5890,4896026,4896027,4896028,4904077,4908285,4908286,4909400,4912445,4923309,4926
059,4927923,4928696,4933783,4945541,4945542,4953153,4957003,4957004,4959607,49622
63,4962264,4964465,4978207,4987205,5006526,5006527,5006528,5013652,5017317,502458
3,5024584,5028220,5028221,5040253,5055183,5059663,5060477,5073745,5077447,5077448,
5077449,5085254,5093705,5094502,5094503,5094504,5096679,5099529,5099530,5109281,5
111064,5120020,5129790,5129791,5130699,5130700,5133289,5152017,5152018,5153107,51
76528,5177754,5183764,5183765,5185030,5194742,5196721,5199829,5214391,5214392,521
7423,5231551,5231552,5232317,5232318,5232319,5248338,5248513,5249267,5249268,5266
595,5266596,5266597,5267492,5267493,5276345,5276346,5285119,5285120,5285121,52851
22,5285123,5287535,5287536,5298588,5332238,5337546,5348806,5348807,5365585,536632
9,5368474,5383368,5383369,5400857,5404013,5410812,5410813,5422314,5423350,5428712,
5430048,5440674,5442893,5447090,5447091,5448233,5448234,5455613,5467013,5467014,5
467741,5470004,5470005,5471951,5484497,5485276,5504430,5507399,5510161,5510162,55
19822,5519823,5537706,5560864,5607293,5607294,5610240,5613018,5621836,5622620,563
0522,5642463,5648286,5652653,5657431,5657432,5658219,5658220,5674323,5676427,5694
839,5696410,5699059,5705820,5717810,5732895,5732896,5735790,5745921,5758454,57689
54,5802021,5808101,5808102,5813021,5813022,5819283,5836759,5836760,5841903,585924
5,5861405,5874378,5887323,5892539,5899861,5904630,5907508,5913257,5917568,5930962,
5940656,5940657,5941450,5954693,5960764,5960765,5982699,5982700,6019136,6019137,6
023570,6029279,6031552,6031553,6034576,6044706,6044707,6046685,6055460,6055461,60
64631,6064632,6065423,6065424,6065425,6067572,6069393,6069394,6073250,6083045,609
6363,6128761,6128762,6128763,6131298,6131299,6152758,6160909,6171435,6182191,6185
714,6195116,6197586,6198543,6208779,6208780,6208781,6208782,6214683,6214684,62208
44,6221054,6221055,6221056,6229129,6295879,6297029,6297030,6297031,6317320,631809
1,6322498,6331438,6331439,6331440,6336344,6339421,6339422,6345459,6355480,6361932,
6374840,6384668,6390369,6390370,6393661,6393662,6396945,6396946,6396947,6396948,6
396949,6396950,6399581,6399582,6399583,6403039,6446966,6446967,6446968,6464299,64
69607,6469608,6470464,6470465,6490248,6490249,6510064,6512469,6512470,6512471,651
4721,6518902,6528957,6544165,6544166,6549812,6550666,6560660,6560661,6560662,6571
368,6571369,6575271,6582482,6601830,6601831,6623095,6623919,6637806,6642958,66461
51,6652633,6662794,6663613,6663614,6663615,6663616,6668185,6672635,6682939,668400

3,6684004,6684005,6698495,6698496,6705064,6705065,6732693,6732694,6736822,6736823,
6755660,6776824,6779887,6779888,6779889,6779890,6779891,6782551,6784811,6789460,6
798831,6802263,6812773,6825834,6825835,6832494,6848712,6852840,6852841,6856753,68
72295,1000011645,1000025342,1000032415,1000037309,1000037310,1000046012,10000473
20,1000052654,1000070909,1000075820,1000088045,1000093761,1000094546,1000102997,
1000118249,1000118250,1000122721,1000129427,1000147724,1000157939,1000157940,100
0158544,1000171094,1000171095,1000184977,1000185930,1000188174,1000199976,100020
1381,1000208325,1000211073,1000211074,1000223523,1000239187,1000240899,100024183
8,1000245610,1000263246,1000267934,1000286340,1000294679,1000294680,1000294681,1
000296965,1000299255,1000301199,1000303382,1000304709,1000308958,1000312394,1000
325852,1000328968,1000332752,1000337541,1000338827,1000338828,1000339654,1000344
280,1000347346,1000349592,1000351514,1000369266,1000378329,1000389030,1000391175
,1000395455,1000397501,1000398202,1000403140,1000403141,1000407083,1000426889,10
00426890,1000435575,1000437595,1000439130,1000446381,1000450034,1000454582,10004
57583,1000458149,1000462117,1000465794,1000477427,1000486950,1000489314,10004900
12,1000491571,1000494972,1000496904,1000505741,1000505742,1000507875,1000513772,
1000515331,1000516376,1000516377,1000517883,1000521195,1000526296,1000530447,100
0564018,1000565628,1000565629,1000568750,1000568751,1000570382,1000570684,100058
8357,1000590882,1000591665,1000591666,1000605841,1000605842,1000607483,100062374
4,1000625427,1000629996,1000631422,1000635605,1000642738,1000644362,1000648490,1
000651043,1000663045,1000663848,1000665400,1000667787,1000670510,1000672890,1000
678050,1000691764,1000697231,1000711129,1000726716,1000740977,1000750118,1000767
208,1000778531,1000779270,1000779271,1000787987,1000787988,1000802956,1000807963
,1000815501,1000847443,1000849900,1000857670,1000861644,1000883499,1000895212,10
00895213,1000895214,1000901037,1000901846,1000909907,1000909908,1000910275,10009
10276,1000910277,1000914274,1000914941,1000916173,1000919788,1000928227,10009306
84,1000930685,1000931409,1000933561,1000941688,1000945548,1000945894,1000955146,
1000955147,1000955148,1000963767,1000964436,1000990151,1000990152,1000990153,100
0990154,1001001152,1001011093,1001018505,1001030258,1001030259,1001034800,100103
4801,1001048287,1001057058,1001059914,1001060579,1001080510,1001083949,100109001
6,1001095566,1001100210,1001109458,1001114254,1001124119,1001125170,1001127387,1
001138237,1001141244,1001142962,1001145879,1001147538,1001150630,1001158612,1001
162551,1001170065,1001173002,1001177604,1001182204,1001183883,1001203228,1001207
051,1001207052,1001207053,1001207054,1001208533,1001210119,1001210891,1001212910
,1001212911,1001217417,1001223643,1001229102,1001236378,1001238650,1001240351,10
01241078,1001247657,1001252154,1001252155,1001252156,1001255033,1001256521,10012
56522,1001258034,1001269874,1001273001,1001273002,1001274624,1001275281,10012752
82,1001275837,1001277313,1001281762,1001281763,1001286327,1001290779,1001292364,
1001292365,1001294127,1001307161,1001312096,1001313741,1001313742,1001316495,100
1322265,1001325020,1001325021,1001325022,1001325869,1001326748,1001338110,100134
3513,1001344778,1001345465,1001346824,1001347039,1001349111,1001351894,100135325
8,1001355438,1001358169,1001358170,1001359831,1001368859,1001374149,1001375734,1
001391133,1001400134,1001401961,1001406469,1001410634,1001413517,1001414105,1001
431851,1001432181,1001456294,1001456295,1001477869,1001490780,1001491023,1001495

```
691,1001507314,1001514429,1001522346,1001525363,1001530149,1001546640,1001547341
,1001550987,1001555381,1001555382,1001555383,1001559622,1001559623,1001559624,10
01568136,1001572455,1001578649,1001582883,1001582884,1001602309,1001604155,10016
08196,1001608197,1001608198,1001620931,1001622375,1001623062,1001637685,10016515
37,1001653324,1001654063,1001657985,1001657986,1001660133,1001660842,1001669693,
1001669694,1001677678,1001679046,1001700927,1001700928,1001702721,1001704681,100
1742093,1001742094,1001744392,1001745345,1001745346,1001746262,1001750784,100175
2689,1001753784,1001754671,1001754672,1001754673,1001754674,1001756972,100176008
6,1001763850,1001765785,1001765786,1001765787,1001772081,1001772082,1001772083,1
001772084,1001794530,1001801519,1001801520,1001803594,1001804772,1001805719,1001
810141,1001819005,1001824247,1001841089,1001843127,1001849164,1001852646,1001855
508,1001856677,1001870205,1001880892,1001882216,1001891781,1001897038,1001897039
,1001904763,1001917400,1001929818]
filter_prod=[100001087957081,100003814221599,100003886102970,100005830621411,1000
07500102178,100008165210077,100071722680513,100073772952220,100088392936336,100
967582824162]
b4_sample=b4[b4['TC_id'].isin(filter_tcid)]
pd.io.sql.to_sql(b4_sample,'purchases', con=engine, index=False, if_exists='replace')
```

### Use MySQL to manipulate data:

```
create table trips_2004
select * from trips where year(TC_date)=2004;


#a. How many:
#1 Store shopping trips are recorded in your database?
Select count(hh_id)
FROM trips where year(TC_date)=2004 ;

#2 Households appear in your database?
Select Count(Distinct(hh_id))
from households;

#3 Stores of different retailers appear in our data base?
SELECT Count(distinct(TC_retailer_code))
FROM trips where year(TC_date)=2004 ;

#4 Different products are recorded?
#i. Products per category and products per module
Select group_at_prod_id,Count(Prod_id)
from products
```

```
group by group_at_prod_id;

Select module_at_prod_id,Count(Prod_id)
from products
group by module_at_prod_id;

#ii. Plot the distribution of products per module and products per category
drop table if exists department;
create table department
select a.department_at_prod_id as department, products, module
from (Select department_at_prod_id, Count(distinct prod_id) as products
  from products
    group by department_at_prod_id) as a inner join
    (select department_at_prod_id, count(distinct module_at_prod_id) as module
    from products
    group by department_at_prod_id) as b
    on a.department_at_prod_id= b.department_at_prod_id;


#5 Transactions
#i. Total transactions and transactions realized under some kind of promotion.
SELECT Count(TC_id)
FROM purchases;

SELECT Count(TC_id)
from purchases
where deal_flag_at_TC_prod_id = 1;

## b(1)

drop table if exists t1;
create table t1
SELECT hh_id, TC_date , ROW_NUMBER() OVER (ORDER BY TC_date) as ID  FROM trips
where year(TC_date)=2004  order by hh_id, TC_date;

drop table if exists t2;
create  table t2
        select *,  ID+1  as ID_2 from t1 order by hh_id, TC_date;


drop table if exists t3;
create  table t3
select
       A.hh_id    as  hh_id_0 ,
        A.TC_date    as  TC_date_0 ,
                B.hh_id    as  hh_id_1 ,
        B.TC_date    as  TC_date_1 ,
        datediff(B.TC_date   ,A.TC_date )/30 as TIME_WINDOW
from
         t2 as A
         inner join
         t1 as B
         on     A.ID_2 = B.ID;
select sum(TIME_WINDOW)
from
(select count(TIME_WINDOW) as TIME_WINDOW from t3 where TIME_WINDOW>3
) a;

#b(2)
# hh_id's with shopping every month
CREATE  TABLE HH_once_month as
(SELECT *
FROM
(Select hh_id,Count(hh_id) AS Months
from
(Select hh_id,month(TC_date)
from trips
group by hh_id,month(TC_date)) A
group by hh_id) B
WHERE months>11);

#Left joining trips data for those who shop once a month
Create  table HH_once_month_all as
(select A.*
from trips A
left join HH_once_month B
on A.hh_id=B.hh_id
where B.hh_id is not null);

#Calculating avg spent per household
DROP TABLE HH_AVG_SPENT;
CREATE TABLE HH_AVG_SPENT AS
select hh_id,sum(tc_total_spent)/12 AS AVG_SPENT
from HH_once_month_all
group by hh_id;
#having hh_id='9001556';

#Calculating average spent per retailer
```

```
DROP TABLE HH_AVG_SPENT_RETAILER;
CREATE TABLE HH_AVG_SPENT_RETAILER AS
select hh_id,tc_retailer_code,sum(tc_total_spent)/12 AS AVG_SPENT_PER_RETAILER
from HH_once_month_all
group by hh_id,tc_retailer_code;
#having hh_id='9001556';

#retailers percentage share
create table HH_PERCENTAGE_OF_SPENT AS
SELECT
A.hh_id,A.tc_retailer_code,A.AVG_SPENT_PER_RETAILER,B.AVG_SPENT,(A.AVG_SPENT_
PER_RETAILER/B.AVG_SPENT)*100 AS 'PERCENTAGE_OF_SPENT'
FROM HH_AVG_SPENT_RETAILER A
LEFT JOIN HH_AVG_SPENT B
ON A.HH_ID=B.HH_ID;

#one retailers with more than 80% share
select *
from HH_PERCENTAGE_OF_SPENT
where 'PERCENTAGE_of_spent'>=80;

#two retailers with more than 80% share
select
A.HH_ID,A.TC_RETAILER_CODE,A.AVG_SPENT_PER_RETAILER,B.AVG_SPENT_PER_RE
TAILER,(A.AVG_SPENT_PER_RETAILER+B.AVG_SPENT_PER_RETAILER),
((A.AVG_SPENT_PER_RETAILER+B.AVG_SPENT_PER_RETAILER)/A.AVG_SPENT)*100
from HH_PERCENTAGE_OF_SPENT A
LEFT JOIN HH_PERCENTAGE_OF_SPENT B
ON A.HH_ID=B.HH_ID
AND A.TC_RETAILER_CODE<>B.TC_RETAILER_CODE
WHERE
((A.AVG_SPENT_PER_RETAILER+B.AVG_SPENT_PER_RETAILER)/A.AVG_SPENT)*100>8
0

#i. Are their demographics remarkably different? Are these people richer? Poorer?
SELECT A.HH_ID,B.HH_INCOME
FROM HH_2_RETAILERS_80 A
LEFT JOIN HOUSEHOLDS B
ON A.HH_ID=B.HH_ID;

#ii. What is the retailer that has more loyalists?
SELECT TC_RETAILER_CODE,SUM(TC_TOTAL_SPENT)
FROM TRIPS
GROUP BY TC_RETAILER_CODE
```

```
ORDER BY SUM(TC_TOTAL_SPENT) DESC
LIMIT 1;

#iii. Where do they live? Plot the distribution by state.
SELECT A.HH_ID,B.HH_STATE
FROM HH_2_RETAILERS_80 A
LEFT JOIN HOUSEHOLDS B
ON A.HH_ID=B.HH_ID


#b(3)
#i
DROP  TABLE IF EXISTS Num_items;
CREATE TABLE Num_items
SELECT month, AVG(number) AS num_items
FROM (SELECT h.hh_id, MONTH(t.TC_date) AS month, AVG(p.quantity_at_TC_prod_id) AS number
  FROM trips_2004 t INNER JOIN households h on t.hh_id=h.hh_id
      INNER JOIN Purchases p on t.TC_id=p.TC_id
  GROUP BY h.hh_id, MONTH(t.TC_date)) AS A
GROUP BY month
ORDER BY month ASC;
#ii
DROP  TABLE IF EXISTS Num_trips;
CREATE TABLE Num_trips
SELECT month, AVG(number) AS num_trips
FROM (SELECT h.hh_id, MONTH(t.TC_date) AS month, COUNT(t.TC_id) AS number
  FROM trips_2004 t INNER JOIN households h on t.hh_id=h.hh_id
  GROUP BY h.hh_id, MONTH(t.TC_date)) AS B
GROUP BY month
ORDER BY month ASC;
#iii
DROP TABLE IF EXISTS t4;
CREATE  TABLE t4
SELECT hh_id, TC_date , ROW_NUMBER() OVER (ORDER BY hh_id, TC_date) as ID
FROM trips_2004
order by hh_id, TC_date;

DROP TABLE IF EXISTS t5;
CREATE  TABLE t5
SELECT *,  ID+1  AS ID_2 FROM t1 order by hh_id, TC_date;

DROP TABLE IF EXISTS t6;
CREATE  TABLE t6
```

```
SELECT A.hh_id as hh_id_0, A.TC_date as TC_date_0, B.hh_id as hh_id_1, B.TC_date as TC_date_1,
datediff(B.TC_date, A.TC_date ) AS time_window
FROM  t5 AS A INNER JOIN t4 AS B ON  A.ID_2 = B.ID;

SELECT hh_id_0 as household, AVG(time_window) AS days
FROM t6
WHERE hh_id_0=hh_id_1
GROUP BY hh_id_0;


#3(1)

SELECT t.month, i.num_items, t.num_trips
FROM Num_trips t INNER JOIN  Num_items i on i.month=t.month;


#3(2)
select total_price_paid_at_TC_prod_id/quantity_at_TC_prod_id as avg_price_paid_per_item ,quantity_at_TC_prod_id as quantity_at_TC_prod_id from purchases;


#c(3)i

# join in table and get ratio
# note that this doesn't include ratio, where there are no private label products belonging to that product group.
# This is not important in the question though as it asks for the groups with the highest fraction of private label goods.

drop table if exists categories_CTL_BR;
create  table categories_CTL_BR
select a.group_at_prod_id, a.count_pl, b.count_all, a.count_pl/b.count_all*100 as proportion
FROM
        (select count(brand_at_prod_id) as count_pl, group_at_prod_id
from products
WHERE brand_at_prod_id = "CTL BR"
group by group_at_prod_id
order by count(brand_at_prod_id)) AS a

INNER JOIN
        (select count(brand_at_prod_id) as count_all, group_at_prod_id
from products
group by group_at_prod_id
order by count(brand_at_prod_id)) AS b

ON a.group_at_prod_id = b.group_at_prod_id;


#c(3)ii

## part c, ii
# join the tables needed with the relevant info



# get total monthly expenditure by month
drop table if exists pl_expend;
create table pl_expend
select sum(total_price_paid_at_TC_prod_id) as "all_prods", month(TC_date) as month1
FROM (select  b.TC_date, b.TC_id, a.brand_at_prod_id, a.prod_id,
b.total_price_paid_at_TC_prod_id
FROM products as a INNER JOIN
(select a.total_price_paid_at_TC_prod_id, a.prod_id, a.TC_id, b.TC_date
FROM purchases as a INNER JOIN trips_2004 as b on a.TC_id = b.TC_id) AS b
ON a.prod_id = b.prod_id) as c
group by month(TC_date)
order by month(TC_date);


# get total expenditure by month on private labell branded products
drop table if exists private_expend;
create table private_expend
select sum(total_price_paid_at_TC_prod_id) as "pl_prods", month(TC_date) as month2
FROM (select  b.TC_date, b.TC_id, a.brand_at_prod_id, a.prod_id,
b.total_price_paid_at_TC_prod_id
FROM products as a INNER JOIN
(select a.total_price_paid_at_TC_prod_id, a.prod_id, a.TC_id, b.TC_date
FROM purchases as a INNER JOIN trips_2004 as b on a.TC_id = b.TC_id) AS b
ON a.prod_id = b.prod_id) as c
WHERE brand_at_prod_id = "CTL BR"
group by month(TC_date)
order by month(TC_date);

drop table if exists CTL_BR_expenditure_share;
create table CTL_BR_expenditure_share
SELECT *, pl_prods/all_prods as share
FROM private_expend a inner join pl_expend b
```

```
    on a.month2=b.month1;

#c(3)iii
##low level income group
SELECT AVG(total_low), AVG(private_low), AVG(private_low)/AVG(total_low) AS percentage
FROM
(SELECT Y1.TC_date, total_low, private_low, (private_low/total_low) AS percentage FROM
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS total_low FROM
 (SELECT A1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS low_income FROM dta_at_hh WHERE hh_income BETWEEN
3 AND 15) AS A1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS A2
  ON A1.hh_id = A2.hh_id) AS B1
 INNER JOIN
  (SELECT TC_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS B2
 ON B1.TC_id = B2.TC_id
 GROUP BY TC_date) AS Y1
INNER JOIN
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS private_low FROM
 (SELECT AA1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS low_income FROM dta_at_hh WHERE hh_income BETWEEN
3 AND 15) AS AA1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS AA2
  ON AA1.hh_id = AA2.hh_id) AS BB1
 INNER JOIN
 (SELECT TC_id, total_price_paid_at_TC_prod_id FROM
  (SELECT TC_id, prod_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS W2
  INNER JOIN
  (SELECT prod_id FROM dta_at_prod_id WHERE brand_at_prod_id REGEXP "CTL BR") AS
W1
  ON W1.prod_id = W2.prod_id) AS BB2
 ON BB1.TC_id = BB2.TC_id
 GROUP BY TC_date) AS Y2
ON Y1.TC_date = Y2.TC_date) AS F1;
#middle level income group
SELECT AVG(total_mid), AVG(private_mid), AVG(private_mid)/AVG(total_mid) AS percentage
FROM
(SELECT Y1.TC_date, total_mid, private_mid, (private_mid/total_mid) AS percentage FROM
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS total_mid FROM
 (SELECT A1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS mid_income FROM dta_at_hh WHERE hh_income
BETWEEN 16 AND 21) AS A1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS A2
  ON A1.hh_id = A2.hh_id) AS B1
 INNER JOIN
  (SELECT TC_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS B2
 ON B1.TC_id = B2.TC_id
 GROUP BY TC_date) AS Y1
INNER JOIN
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS private_mid FROM
 (SELECT AA1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS mid_income FROM dta_at_hh WHERE hh_income
BETWEEN 16 AND 21) AS AA1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS AA2
  ON AA1.hh_id = AA2.hh_id) AS BB1
 INNER JOIN
 (SELECT TC_id, total_price_paid_at_TC_prod_id FROM
  (SELECT TC_id, prod_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS W2
  INNER JOIN
  (SELECT prod_id FROM dta_at_prod_id WHERE brand_at_prod_id REGEXP "CTL BR") AS
W1
  ON W1.prod_id = W2.prod_id) AS BB2
 ON BB1.TC_id = BB2.TC_id
 GROUP BY TC_date) AS Y2
ON Y1.TC_date = Y2.TC_date) AS F1;

#high level income group
SELECT AVG(total_high), AVG(private_high), AVG(private_high)/AVG(total_high) AS
percentage FROM
(SELECT Y1.TC_date, total_high, private_high, (private_high/total_high) AS percentage FROM
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS total_high FROM
 (SELECT A1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS high_income FROM dta_at_hh WHERE hh_income > 21) AS
A1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS A2
  ON A1.hh_id = A2.hh_id) AS B1
 INNER JOIN
  (SELECT TC_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS B2
 ON B1.TC_id = B2.TC_id
 GROUP BY TC_date) AS Y1
INNER JOIN
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS private_high FROM
 (SELECT AA1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS high_income FROM dta_at_hh WHERE hh_income > 21) AS
AA1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS AA2
  ON AA1.hh_id = AA2.hh_id) AS BB1
 INNER JOIN
 (SELECT TC_id, total_price_paid_at_TC_prod_id FROM
  (SELECT TC_id, prod_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS W2
  INNER JOIN
  (SELECT prod_id FROM dta_at_prod_id WHERE brand_at_prod_id REGEXP "CTL BR") AS
W1
  ON W1.prod_id = W2.prod_id) AS BB2
 ON BB1.TC_id = BB2.TC_id
 GROUP BY TC_date) AS Y2
ON Y1.TC_date = Y2.TC_date) AS F1;


#c(3)iii
##low level income group
SELECT AVG(total_low), AVG(private_low), AVG(private_low)/AVG(total_low) AS percentage
FROM
(SELECT Y1.TC_date, total_low, private_low, (private_low/total_low) AS percentage FROM
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS total_low FROM
 (SELECT A1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS low_income FROM dta_at_hh WHERE hh_income BETWEEN
3 AND 15) AS A1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS A2
  ON A1.hh_id = A2.hh_id) AS B1
 INNER JOIN
  (SELECT TC_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS B2
 ON B1.TC_id = B2.TC_id
 GROUP BY TC_date) AS Y1
INNER JOIN
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS private_low FROM
 (SELECT AA1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS low_income FROM dta_at_hh WHERE hh_income BETWEEN
3 AND 15) AS AA1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS AA2
  ON AA1.hh_id = AA2.hh_id) AS BB1
 INNER JOIN
 (SELECT TC_id, total_price_paid_at_TC_prod_id FROM
  (SELECT TC_id, prod_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS W2
  INNER JOIN
  (SELECT prod_id FROM dta_at_prod_id WHERE brand_at_prod_id REGEXP "CTL BR") AS
W1
  ON W1.prod_id = W2.prod_id) AS BB2
 ON BB1.TC_id = BB2.TC_id
 GROUP BY TC_date) AS Y2
ON Y1.TC_date = Y2.TC_date) AS F1;
#middle level income group
SELECT AVG(total_mid), AVG(private_mid), AVG(private_mid)/AVG(total_mid) AS percentage
FROM
(SELECT Y1.TC_date, total_mid, private_mid, (private_mid/total_mid) AS percentage FROM
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS total_mid FROM
 (SELECT A1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS mid_income FROM dta_at_hh WHERE hh_income
BETWEEN 16 AND 21) AS A1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS A2
  ON A1.hh_id = A2.hh_id) AS B1
 INNER JOIN
  (SELECT TC_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS B2
 ON B1.TC_id = B2.TC_id
 GROUP BY TC_date) AS Y1
INNER JOIN
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS private_mid FROM
 (SELECT AA1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS mid_income FROM dta_at_hh WHERE hh_income
BETWEEN 16 AND 21) AS AA1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS AA2
  ON AA1.hh_id = AA2.hh_id) AS BB1
 INNER JOIN
 (SELECT TC_id, total_price_paid_at_TC_prod_id FROM
  (SELECT TC_id, prod_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS W2
  INNER JOIN
  (SELECT prod_id FROM dta_at_prod_id WHERE brand_at_prod_id REGEXP "CTL BR") AS
W1
  ON W1.prod_id = W2.prod_id) AS BB2
 ON BB1.TC_id = BB2.TC_id
 GROUP BY TC_date) AS Y2
ON Y1.TC_date = Y2.TC_date) AS F1;

#high level income group
SELECT AVG(total_high), AVG(private_high), AVG(private_high)/AVG(total_high) AS
percentage FROM
(SELECT Y1.TC_date, total_high, private_high, (private_high/total_high) AS percentage FROM
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS total_high FROM
 (SELECT A1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS high_income FROM dta_at_hh WHERE hh_income > 21) AS
A1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS A2
  ON A1.hh_id = A2.hh_id) AS B1
 INNER JOIN
  (SELECT TC_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS B2
 ON B1.TC_id = B2.TC_id
 GROUP BY TC_date) AS Y1
INNER JOIN
(SELECT TC_date, SUM(total_price_paid_at_TC_prod_id) AS private_high FROM
 (SELECT AA1.hh_id, TC_date, TC_id FROM
  (SELECT hh_id, hh_income AS high_income FROM dta_at_hh WHERE hh_income > 21) AS
AA1
  INNER JOIN
  (SELECT hh_id, DATE_FORMAT(TC_date, '%Y-%m') AS TC_date, TC_id FROM dta_at_TC)
AS AA2
  ON AA1.hh_id = AA2.hh_id) AS BB1
 INNER JOIN
 (SELECT TC_id, total_price_paid_at_TC_prod_id FROM
  (SELECT TC_id, prod_id, total_price_paid_at_TC_prod_id FROM dta_at_TC_upc) AS W2
  INNER JOIN
  (SELECT prod_id FROM dta_at_prod_id WHERE brand_at_prod_id REGEXP "CTL BR") AS
W1
  ON W1.prod_id = W2.prod_id) AS BB2
 ON BB1.TC_id = BB2.TC_id
 GROUP BY TC_date) AS Y2
ON Y1.TC_date = Y2.TC_date) AS F1;
```