Instacart Market Basket Analysis

Name: Nikita Vispute **Netid**: nxv170005

Name: Arpita Dutta **Netid**: axd170025

The dataset is a relational set of files describing customers' orders over time. We are working with below 3 datasets:

1. Order_products_prior.csv
2. Products.csv
3. Departments.csv

Packages required:

- Arules
- readr
- Plyr

Data Pre-processing:

Used read_csv function to load the datasets into R. Order_products_prior dataset looks like below:

```
> head(order_products_prior)
  order_id product_id add_to_cart_order reordered
1        2      33120                 1         1
2        2      28985                 2         1
3        2       9327                 3         0
4        2      45918                 4         1
5        2      30035                 5         0
6        2      17794                 6         1
```

```
> head(products)
  product_id                                                product_name aisle_id department_id
1          1                                  Chocolate Sandwich Cookies       61            19
2          2                                            All-Seasons Salt      104            13
3          3                        Robust Golden Unsweetened Oolong Tea       94             7
4          4 Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce       38             1
5          5                                   Green Chile Anytime Sauce        5            13
6          6                                                Dry Nose Oil       11            11
> |
```

```
> head(products)
# A tibble: 6 x 4
  product_id product_name                                                    aisle_id department_id
       <dbl> <chr>                                                              <dbl>         <dbl>
1          1 Chocolate Sandwich Cookies                                            61            19
2          2 All-Seasons Salt                                                     104            13
3          3 Robust Golden Unsweetened Oolong Tea                                  94             7
4          4 Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce     38             1
5          5 Green Chile Anytime Sauce                                              5            13
6          6 Dry Nose Oil                                                          11            11
> |
```

Order_products_prior dataset contains below fields:

- Order_id
- Product_id
- Add_to_cart_order
- Reordered

The products.csv file contains below fields:

- Product_id
- Product_name
- Aisle_id
- Department_id

The departments.csv file contains the below fields:

- Department_id
- Department

We merged order_products_prior and products dataset to get the corresponding product name and grouped the item names by order_id and store in transactionData and convert it to transaction object.

```
transactions as itemMatrix in sparse format with
 3214875 rows (elements/itemsets/transactions) and
 291600 columns (items) and a density of 3.381286e-05

most frequent items:
                Banana Bag of Organic Bananas   Organic Strawberries   Organic Baby Spinach   Organic Hass Avocado
                444134                 372129                 256230                 235489                 191943
               (Other)
              30198196

element (itemset/transaction) length distribution:
sizes
     1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17     18
176763 200371 215192 225877 228550 226274 216798 200868 182823 163882 145703 129726 115236 101690  89355  78800  68917  60533
    19     20     21     22     23     24     25     26     27     28     29     30     31     32     33     34     35     36
 52999  45961  40215  34705  30186  25930  22501  19418  16690  14326  12364  10650   8969   7611   6628   5636   4946   4107
    37     38     39     40     41     42     43     44     45     46     47     48     49     50     51     52     53     54
  3476   2993   2547   2164   1772   1518   1336   1102    999    858    711    604    498    453    363    325    236    223
    55     56     57     58     59     60     61     62     63     64     65     66     67     68     69     70     71     72
   194    171    151    119    110     83     78     74     64     53     48     39     34     29     36     23     11     13
    73     74     75     76     77     78     79     80     81     82     83     84     85     86     87     88     89     90
    25     16     17     11     10     11      6      6      7      3      3      5      6      3      2      2      3      1
    91     92     93     94     95     98     99    100    102    105    110    111    112    113    114    119    139    150
     2      1      2      1      3      2      3      1      4      2      1      2      1      1      1      1      1      1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    4.00    8.00    9.86   13.00  150.00

includes extended item information - examples:
                 labels
1                    #2
2        #2 Coffee Filters
3 #2 Cone White Coffee Filters
```

<u>Frequent itemsets for products in orders dataset. You have to output product names and not just product id</u>

From transaction Data object tr, we can run summary(tr) which gives very useful information about the transaction object.

We can get frequent product items by running eclat command on the transaction object.

The parameters used for frequent product items are:

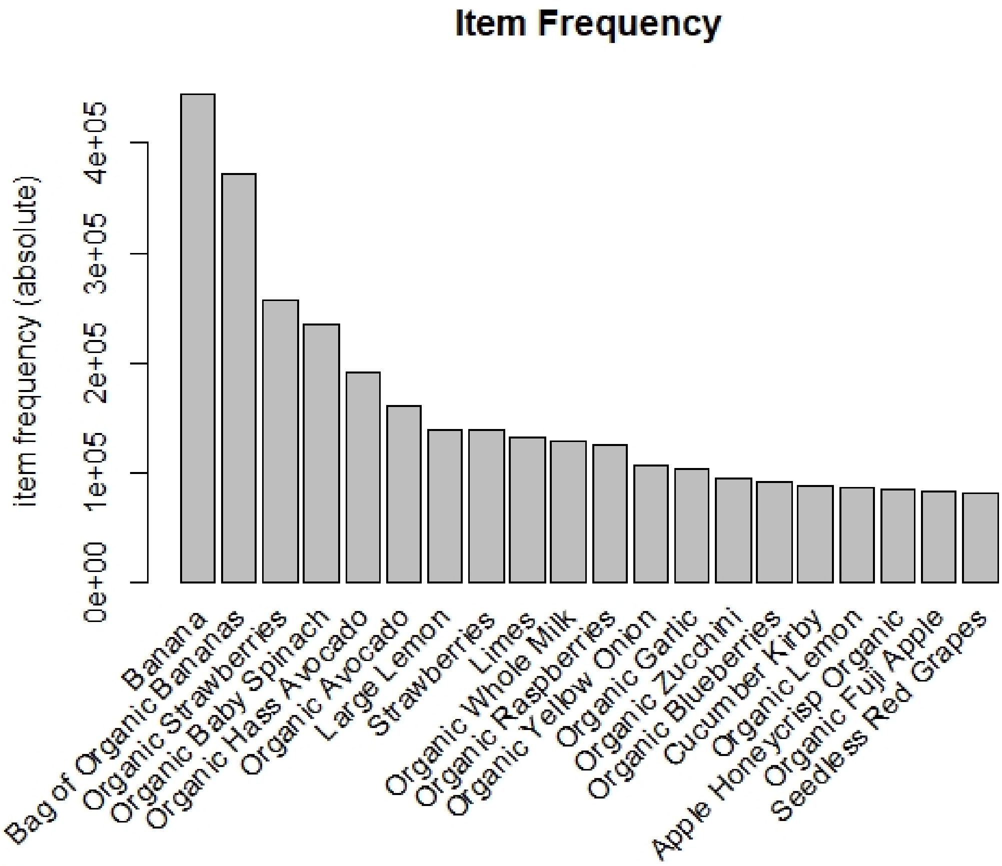
Support: 0.03

Maxlen: 15

The frequent products look like below:

```
      items                     support    count
[1]   {Banana}                  0.13814969 444134
[2]   {Bag of Organic Bananas}  0.11575225 372129
[3]   {Organic Strawberries}    0.07970139 256230
[4]   {Organic Baby Spinach}    0.07324982 235489
[5]   {Organic Hass Avocado}    0.05970465 191943
[6]   {Organic Avocado}         0.05021844 161446
[7]   {Large Lemon}             0.04311210 138600
[8]   {Limes}                   0.04095525 131666
[9]   {Organic Raspberries}     0.03879311 124715
[10]  {Strawberries}            0.04294226 138054
[11]  {Organic Whole Milk}      0.03990357 128285
[12]  {Organic Yellow Onion}    0.03300439 106105
[13]  {Organic Garlic}          0.03219130 103491
```

We can see the frequent items graphically by using the item frequency plot.



<u>Association rules for products in orders dataset. You have to output product names and not just product id</u>

Next step is to mine the rules using the APRIORI algorithm. The function apriori() is from package arules. The apriori() takes the transaction object on which mining to be applied along with parameter values of support and confidence.

The parameters for item_rules are :

Support: 0.001

Confidence: 0.5

The first 20 rules for frequent product items look like below:

```
     lhs                                           rhs                                   support     confidence lift      count
[1]  {Country Stand Juice}                      => {Medium Pulp}                         0.001090245 1          917.22539 3505
[2]  {Medium Pulp}                              => {Country Stand Juice}                 0.001090245 1          917.22539 3505
[3]  {Chocolate Chip Walnut}                    => {Cookies}                             0.001055718 1          701.63138 3394
[4]  {Twin Pack}                                => {Take & Bake}                         0.001005016 1          995.00929 3231
[5]  {Take & Bake}                              => {Twin Pack}                           0.001005016 1          995.00929 3231
[6]  {Twin Pack}                                => {French Baguettes}                    0.001005016 1          995.00929 3231
[7]  {French Baguettes}                         => {Twin Pack}                           0.001005016 1          995.00929 3231
[8]  {Take & Bake}                              => {French Baguettes}                    0.001005016 1          995.00929 3231
[9]  {French Baguettes}                         => {Take & Bake}                         0.001005016 1          995.00929 3231
[10] {2 Huge Rolls = 5 Regular Rolls  Towels/Napkins} => {Select-A-Size Paper Towels}    0.001302383 1          579.98827 4187
[11] {2 Huge Rolls = 5 Regular Rolls  Towels/Napkins} => {White}                         0.001302383 1          457.63345 4187
[12] {Three Cheese}                             => {Pizza Poppers}                       0.001001594 1          998.40839 3220
[13] {Pizza Poppers}                            => {Three Cheese}                        0.001001594 1          998.40839 3220
[14] {Deliciously Hydrating Watermelon Water}   => {Cold-pressed}                        0.001258525 1          794.58107 4046
[15] {Cold-pressed}                             => {Deliciously Hydrating Watermelon Water} 0.001258525 1       794.58107 4046
[16] {Ginger Root Beer}                         => {Naturally Flavored Zero Calorie Soda} 0.001221509 1         614.11175 3927
[17] {Ginger Root Beer}                         => {Caffeine Free}                       0.001221509 1          511.35279 3927
[18] {Sunkissed in the Mediterranean}           => {Wild Non-Pareil Capers}              0.001091800 1          915.91880 3510
[19] {Wild Non-Pareil Capers}                   => {Sunkissed in the Mediterranean}      0.001091800 1          915.91880 3510
[20] {Organic Snack Mix Bunnies Snack Mix}      => {Organic}                             0.001038921 1           47.54537 3340
```

Frequent itemsets for departments in orders dataset. You have to output product names and not just product id

We merged order_products_prior,products and departments dataset to get the corresponding department names of the items purchased per order and grouped the departments names by order_id and store in transactionData1 and convert it to transaction object tr1.

From transaction Data object tr1, we can run summary(tr1) which gives very useful information about the transaction object.

```
transactions as itemMatrix in sparse format with
 3214875 rows (elements/itemsets/transactions) and
 22 columns (items) and a density of 0.2152805

most frequent items:
   produce dairy eggs   beverages      snacks      frozen     (Other)
   2409320     2177338     1457351     1391447     1181018     6609725

element (itemset/transaction) length distribution:
sizes
     1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17     18
287949 395701 468955 488113 447455 367284 279920 198390 130101  78397  41778  19579   7880   2589    651    108     22      3

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   4.000   4.736   6.000  18.000

includes extended item information - examples:
   labels
1 alcohol
2  babies
3  bakery
```

We can get frequent product items by running eclat command on the transaction object.

The parameters used for frequent departmenrs are:

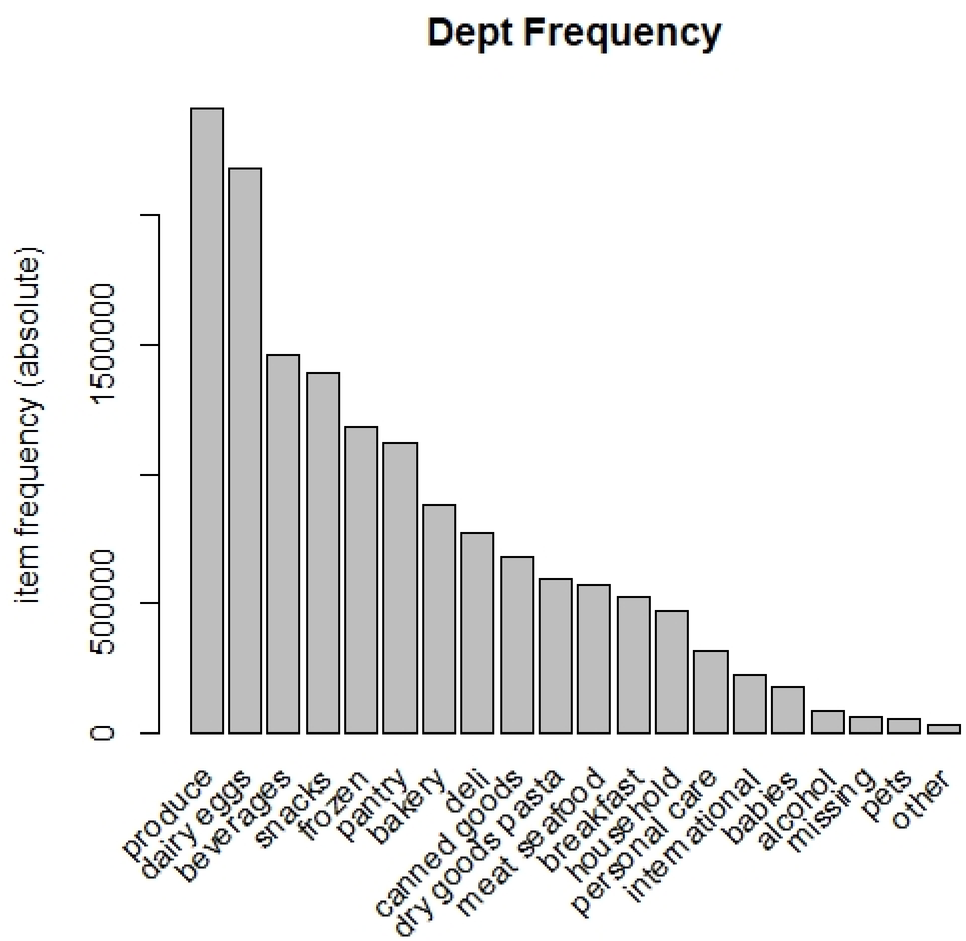
Support: 0.2

Maxlen: 15

The frequent departments look like below:

```
     items                            support   count
[1]  {deli,produce}                   0.2060590  662454
[2]  {bakery,produce}                 0.2289678  736103
[3]  {bakery,dairy eggs}              0.2249972  723338
[4]  {dairy eggs,pantry,produce}      0.2324585  747325
[5]  {pantry,produce}                 0.2869402  922477
[6]  {dairy eggs,pantry}              0.2692615  865642
[7]  {dairy eggs,frozen,produce}      0.2425799  779864
[8]  {frozen,produce}                 0.2981183  958413
[9]  {dairy eggs,frozen}              0.2833796  911030
[10] {beverages,dairy eggs,produce}   0.2611629  839606
[11] {beverages,produce}              0.3335635 1072365
[12] {beverages,dairy eggs}           0.3207378 1031132
[13] {beverages,snacks}               0.2291035  736539
[14] {dairy eggs,produce,snacks}      0.2704659  869514
[15] {produce,snacks}                 0.3372616 1084254
[16] {dairy eggs,snacks}              0.3225478 1036951
[17] {dairy eggs,produce}             0.5504833 1769735
[18] {produce}                        0.7494288 2409320
[19] {dairy eggs}                     0.6772699 2177338
[20] {snacks}                         0.4328153 1391447
[21] {beverages}                      0.4533150 1457351
[22] {frozen}                         0.3673605 1181018
[23] {pantry}                         0.3477249 1117892
[24] {bakery}                         0.2742116  881556
[25] {deli}                           0.2396050  770300
[26] {canned goods}                   0.2119227  681305
```

We can see the frequent departments graphically by using the item frequency plot.



<u>Association rules for departments in orders dataset. You have to output product names and not just product id</u>

Next step is to mine the rules using the APRIORI algorithm. The function apriori() is from package arules. The apriori() takes the transaction object on which mining to be applied along with parameter values of support and confidence.

The parameters for department_rules are :

Support: 0.07

Confidence: 0.5

The rules for frequent departments look like below:

	lhs		rhs	support	confidence	lift	count
[1]	{canned goods,dairy eggs,pantry}	=>	{produce}	0.08483067	0.9217936	1.229995	272720
[2]	{dairy eggs,deli,pantry}	=>	{produce}	0.08249123	0.9169900	1.223585	265199
[3]	{dairy eggs,dry goods pasta,pantry}	=>	{produce}	0.07342867	0.9137443	1.219254	236064
[4]	{canned goods,dairy eggs,frozen}	=>	{produce}	0.08240383	0.9130381	1.218312	264918
[5]	{canned goods,dairy eggs,snacks}	=>	{produce}	0.08535977	0.9123281	1.217365	274421
[6]	{bakery,dairy eggs,deli}	=>	{produce}	0.07833026	0.9085897	1.212376	251822
[7]	{dairy eggs,deli,frozen}	=>	{produce}	0.09052856	0.9073162	1.210677	291038
[8]	{dairy eggs,dry goods pasta,snacks}	=>	{produce}	0.07973343	0.9068887	1.210107	256333
[9]	{bakery,canned goods}	=>	{produce}	0.07238664	0.9068781	1.210092	232714
[10]	{canned goods,pantry}	=>	{produce}	0.09871053	0.9064324	1.209498	317342
[11]	{meat seafood,pantry}	=>	{produce}	0.07578273	0.9050391	1.207638	243632
[12]	{canned goods,dairy eggs}	=>	{produce}	0.15367409	0.9040459	1.206313	494043
[13]	{dairy eggs,dry goods pasta,frozen}	=>	{produce}	0.07794984	0.9035153	1.205605	250599
[14]	{dairy eggs,meat seafood}	=>	{produce}	0.12915774	0.9014228	1.202813	415226
[15]	{beverages,canned goods,dairy eggs}	=>	{produce}	0.07813679	0.9012985	1.202647	251200
[16]	{deli,pantry}	=>	{produce}	0.09426494	0.9000567	1.200990	303050
[17]	{dairy eggs,deli,snacks}	=>	{produce}	0.10425724	0.8996680	1.200472	335174
[18]	{dairy eggs,frozen,pantry,snacks}	=>	{produce}	0.07424363	0.8994555	1.200188	238684
[19]	{dry goods pasta,pantry}	=>	{produce}	0.08325549	0.8982468	1.198575	267656
[20]	{meat seafood,snacks}	=>	{produce}	0.07954275	0.8972348	1.197225	255720