15-Final Project Data Description

April 15, 2019

1 Description of FritoLays Dataset

This document contains the latest description of all the data files for the DSO-570 final project.

1.1 1. Overview of Files

Note that almost all numbers have been altered (sanitized) by FritoLay to protect company trade secrets. (This sanitation process might have created errors and inconsistencies in the data.) However, you can assume that the picture the data portray is reasonably accurate.

- **2018 FLNA 10 RegionMap_DECEMBER.pptx**: a map of the 10 sales regions of FritoLays North America.
- **2018_P12_inventory_report.xlsx**: a report of the amount of inventory of each product in each warehouse in 2018 sales period 12.
- **california_eaches.csv**: a snapshot of the 250 SKUs that are available for eaches picking (individual packages sent to small retailers, instead of large boxes to large chains) at a distribution center in California in March 2019.
- **chain_information.csv**: information on the sales channel of each retail chain to customers. (Examples of channels include convenience stores, small grocery, supermarket, etc.) This can be used to classify chains, although certain chains may have multiple sales channels.

store的种类

- **distribution_cost_cleaned.xlsx**: a report about the transportation cost for shipping each product between warehouses. Such shipping is needed because certain products are only made in certain warehouses.
- **estimated_margins.csv**: estimates of the profit margins of FritoLays for various products. For legal reasons, these are not the true numbers, but are reasonable estimates.
- **Guest_Lecture_Slides_2-26.pptx**: the powerpoint slides Gretchen used in her in-class presentation on Feb 26.
- **manufacturing_platform_capacity.csv**: what percentage of production capacity is each type of manufacturing process currently using.
- **percent_of_stores_carrying_by_region.csv**: on average in 2018, what percentage of retail stores in each sales region carries each product, and whether the product is carried in a region constantly, seasonal, or an innovation product.

- **prices_cleaned.csv.gz**: a list of retail prices of FritoLays product in various price regions, as well as how much FritoLays charges the stores.

- **RalphsIRI-cleaned.csv.gz**: sales registry data of FritoLays product from the supermarket chain Ralphs, collected by an independent organization called IRI. This data contains actual customer demand as well as promotional patterns.

- **sales2018.csv.gz**: all sales of FritoLays products in 2018 in each sales period (each period is 4 weeks). This data records sales in each region to each retail chain. (Examples of retail chain include Ralphs, Seven Eleven, and Ranch 99.) Due to customer privacy, the identity of the chain is hidden and you only have a numerical ID.

- **sales2018_products.csv.gz**: more detailed information about each product in the sales2018.csv.gz dataset.

- **sales2018_SOUTHWEST_deepdive.csv.gz**: more detailed sales data for the Southwest region (which includes Los Angeles). This shows data from each warehouse to each retail chain.

- **sales2018_SOUTHWEST_warehouses.csv.gz**: more detailed information about the warehouses in the above dataset.

- **video_picking_boxes.mp4**: Video of warehouse worker picking large boxes for shipment to large retailers.

- **video_picking_eaches.mp4**: Video of warehouse worker picking "eaches" (individual packages) for shipment to small stores.

- **warehouses_north_america.xlsx**: location information on every FritoLays warehouse in North America. The types of warehouses include production plants, large distribution centers (DCs), and smaller warehouses called "bins" serving remote areas.

1.2 2. Detailed Description of Each File

The following contains a description of every column of every file, as well as code of how to load the data using Pandas.

1.2.1 3.1 National Sales Data (sales2018.csv.gz)

Overview: This contains (altered/sanitized) retail sales data for all of US in 2018, from each of the 10 sales regions to each chain for each product in each sales period.

Columns:

- sales_period: 4 weeks are in one sales period. There are 13 periods per year.
- region: name of the sales region. There are 10 regions.
- chain_id: a unique identifier of the retail chain to which the product was sold. Examples of chains include Walmart, 7-11, 99 Ranch, or Ralphs.
- BDC: a product code that is used internally by Fritolays.
- BDC_description: an abbreviated description of the product.
- GTIN: a product code that is used for communicating with clients.

- sales: total monetary amount sold for this item from this region to this chain during this period, in dollars. (Certain values are negative, but you should view this as error from the data sanitization.)
- returns: total monetary amount returned that period because of item expiration. (Fritolay calls these stale items.)

```
In [17]: import pandas as pd
         sales=pd.read_csv('project_data/sales2018.csv.gz')
         sales.head()
```

```
Out[17]:    sales_period     region  chain_id      BDC     BDC_description  \
         0             1  SOUTHWEST       6.0  2015006    LAY'S XL HON BBQ
         1             1  SOUTHWEST       6.0  2015009        LAY'S XL KCM
         2             1  SOUTHWEST       6.0  2015011       LAY'S  XL REG
         3             1  SOUTHWEST       6.0  2015014  LAY'S XL  REG WAVY
         4             1  SOUTHWEST       6.0  2015014  LAY'S XL  REG WAVY

                   GTIN    sales  returns
         0  28400200684   40.656    0.000
         1  28400199612  269.104    1.936
         2  28400199148  425.920    3.872
         3  28400043809    7.744    0.000
         4  28400199544  114.224    5.808
```

3.1.1 Sales Products (sales2018_products.csv.gz) **Overview:** This contains information about each product from the sales data above. This table was created by grouping by BDC and GTIN and pulling the first of the other fields. You can match information here back to the sales data by merging by BDC and GTIN.

Columns: - BDC: as in the national sales data. - GTIN: as in the national sales data. - BDC_description: as in the national sales data. - UPC: another product code system that is a part of the GTINs. - category_code, category_description, subcategory_code, subcategory_description: internal classification of products by FritoLay. - package_type: how the product is packaged when shipped to the retailer. - business_unit_code, business_unit_description: which subdivision of FritoLay is responsible for this product. - trademark_code, trademark_description: which brand is the product under. - alt_description: longer description of the product merged from another source.

```
In [20]: import pandas as pd
         sales_products=pd.read_csv('project_data/sales2018_products.csv.gz')
         sales_products.head()
```

```
Out[20]:          BDC         GTIN      BDC_description    UPC  category_code  \
         0  702321337  15300200012   RAR CUPS LARGE 12CS  20001             10
         1  702321448  15300200029  RAR CUPS LARGE 4CHEE  20002             10
         2  702321499  15300200036  RAR CUPS LARGE CHD B  20003             10
         3  702321596  15300200135  PASTA RONI CUPS 2.24  20013             10
         4  702321593  15300200142   ECOMM PAST RONI CUP  20014             10
```

```
  category_description  subcategory_code subcategory_description package_type  \
0   OTHER QUAKER - DRY                40       GOLDEN GRAIN PASTA          NaN
1   OTHER QUAKER - DRY                40       GOLDEN GRAIN PASTA          NaN
2   OTHER QUAKER - DRY                40       GOLDEN GRAIN PASTA          NaN
3   OTHER QUAKER - DRY                40       GOLDEN GRAIN PASTA          NaN
4   OTHER QUAKER - DRY                40       GOLDEN GRAIN PASTA          NaN

   business_unit_code business_unit_description trademark_code  \
0                 702        GOLDEN GRAIN PASTA             GG
1                 702        GOLDEN GRAIN PASTA             GG
2                 702        GOLDEN GRAIN PASTA             GG
3                 702        GOLDEN GRAIN PASTA             GG
4                 702        GOLDEN GRAIN PASTA             GG

  trademark_description alt_description
0          GOLDEN GRAIN             NaN
1          GOLDEN GRAIN             NaN
2          GOLDEN GRAIN             NaN
3          GOLDEN GRAIN             NaN
4          GOLDEN GRAIN             NaN
```

1.2.2 3.2 Deep Dive Sales Data (sales2018_SOUTHWEST_deepdive.csv.gz)

Overview: This is more detailed data for the Southwest region, aggregated not at a region level but at a warehouse level.

Columns:

- sales_period: as in the national data.
- sub_region: a smaller partition of the sales region. Also known as "zone".
- warehouse: name of the warehouse from which the product is shipped to retailer.
- chain_id: as in the national sales data.
- BDC, BDC_description, GTIN: information about the product, as in the national sales data.
- sales, returns: as in the national sales data.

```
In [2]: import pandas as pd
        deep_dive=pd.read_csv('project_data/sales2018_SOUTHWEST_deepdive.csv.gz')
        deep_dive.head()
```

```
Out[2]:    sales_period    sub_region       warehouse  chain_id        BDC  \
        0             1  NORTH VALLEY  BAKERSFIELD DC    9634.0    4192051
        1             1  NORTH VALLEY  BAKERSFIELD DC    9634.0   23192019
        2             1  NORTH VALLEY  BAKERSFIELD DC    9634.0   45192130
        3             1  NORTH VALLEY  BAKERSFIELD DC    9634.0  105192883
        4             1  NORTH VALLEY  BAKERSFIELD DC    9634.0    1026006

                BDC_description         GTIN   sales  returns
        0  BN     VAL M HOT-N-S  28400595971  3.9984      0.0
        1  CS HOT VALMID SALVRD  28400596008  3.3516      0.0
        2        HISPAN XL SABRI  28400190800  3.3516      0.0
```

```
3            CS PUF CH  28400002288   2.9988      0.0
4   FRITOS X-XVL FRITOS  28400417723  20.0600      0.0
```

1.2.3 3.2.1 Deep Dive Warehouses (sales2018_SOUTHWEST_warehouses.csv.gz)

Overview: This contains detailed location information about the warehouses in the deep dive sales data for the Southwest region. It can be matched back to the sales deep dive data using the column warehouse.

Columns:

- warehouse: as in the deep dive sales data.
- address, city, state: location of the warehouse.

```
In [18]: deep_dive_warehouses=pd.read_csv('project_data/sales2018_SOUTHWEST_warehouses.csv.gz'
         deep_dive_warehouses.head()
```

```
Out[18]:         warehouse          address        city       state
         0         AJO AZ BIN     650 N 2ND AVE         AJO     ARIZONA
         1         ALAMEDA DC    1450 S LOOP RD     ALAMEDA  CALIFORNIA
         2      ALPINE TX BIN            HWY 90      ALPINE       TEXAS
         3  ANGELS CAMP CA BIN   2245 HIGHWAY 49  ANGELS CAMP  CALIFORNIA
         4     ARTESIA NM BIN  105 W HERMOSA DR     ARTESIA  NEW MEXICO
```

1.2.4 3.3 All Warehouses (warehouses_north_america.xlsx)

Overview: This contains more information about various warehouses, including the latitude and longitude.

Columns:

- combo_code, region_code, subregion_code: various codes used by FritoLay internally, encoding the location of the warehouse.
- LMSID: the unique ID for the warehouse, which matches the distribution cost and inventory data.
- warehouse: name of the warehouse.
- address, city, state, zipcode, latitude, longitude: location of the warehouse.
- crossdock: cross docking describes intermediate locations in which trucks dump their load, which is sorted by warehouse and carried by other trucks to each individual warehouse. This is a practice to pool shipments for several warehouses into the same truck, and save transportation cost.
- type: whether the warehouse is a production plant, a distribution center (DC), or a bin (small warehouse serving remove an area).

```
In [19]: warehouses=pd.read_excel('project_data/warehouses_north_america.xlsx')
         warehouses.head()
```

```
Out[19]:    combo_code  region_code  subregion_code  LMSID          warehouse  \
         0      311666           31             311    666  OAK FOREST IL DC
         1      312666           31             312    666  OAK FOREST IL DC
         2      313666           31             313    666  OAK FOREST IL DC
```

```
3      323666          32           323    666  OAK FOREST IL DC
4      328666          32           328    666  OAK FOREST IL DC

            address    crossdock        city state  zipcode   latitude  \
0  4170 W 166TH ST  KILBOURN AV  OAK FOREST    IL  60452.0  41.588672
1  4170 W 166TH ST  KILBOURN AV  OAK FOREST    IL  60452.0  41.588672
2  4170 W 166TH ST  KILBOURN AV  OAK FOREST    IL  60452.0  41.588672
3  4170 W 166TH ST  KILBOURN AV  OAK FOREST    IL  60452.0  41.588672
4  4170 W 166TH ST  KILBOURN AV  OAK FOREST    IL  60452.0  41.588672

   longitude type
0 -87.718196   DC
1 -87.718196   DC
2 -87.718196   DC
3 -87.718196   DC
4 -87.718196   DC
```

1.2.5 3.4 Chain Information (chain_information.csv)

Overview: While the identity of each retail chain is hidden, this file contains information that can be used to classify chains.
Columns:

- chain_id: the identifier of the chain which matches the sales data.
- sales_channel: how the chain sells products to consumers.
- percentage_by_sales: for the majority of chains, this is equal to 1, meaning that the chain only has one channel. However, certain chains operate a variety of stores and may have multiple sales_channels. This adds up to 1 for every chain_id.

```
In [15]: chain_info=pd.read_csv('project_data/chain_information.csv')
         chain_info.head()
```

```
Out[15]:    chain_id sales_channel  percentage_by_sales
         0         6       C-STORE                  1.0
         1         7 SMALL GROCERY                  1.0
         2        12       C-STORE                  1.0
         3        13 SMALL GROCERY                  1.0
         4        15 SMALL GROCERY                  1.0
```

1.2.6 3.5 Inventory Data (2018_P12_inventory_report.xlsx)

Overview: This is a snapshot of the inventory at all warehouses in 2018 sales period 12.
Columns:

- LMSID: identifier of the warehouse.
- warehouse: name of the warehouse.
- type: type of the warehouse.
- region: sales region.
- BDC, BDC_description,class_code,class_description,produced?: product information.

For the following 7 columns, the data file first expresses them in number of cases, and then in monetary amounts.

- total: the total amount of inventory for the product, which equals cycle+safety_stock+other+in_transit.
- cycle: the minimum amount of inventory needed even if demand is absolutely steady.
- safety_stock: a stockpile of inventory calculated by the supply chain team to prepare for large spikes in demand.
- in_transit: the amount of inventory being carried by trucks and trains during transportation.
- other: other types of inventory not classified as cycle, safety_stock, or in_transit.
- in_building: the sum of cycle, safety_stock, and other.
- daily_demand: the average demand for the product per day.

```
In [37]: inventory=pd.read_excel('project_data/2018_P12_inventory_report.xlsx',skiprows=2)
         inventory.head()
```

```
Out[37]:    LMSID              warehouse   type region       BDC      BDC_description  \
         0  36370  ABERDEEN PLANT-RETAIL  PLANT   EAST  20016097  TOSTITOS XXL SCOOPS
         1  36370  ABERDEEN PLANT-RETAIL  PLANT   EAST   3015009      CT PUF XL JUMBO
         2  36370  ABERDEEN PLANT-RETAIL  PLANT   EAST  90055031  TO DIP XLGBT CONQUE
         3  36370  ABERDEEN PLANT-RETAIL  PLANT   EAST  85021063     VPK SUPER 24CTVPK
         4  36370  ABERDEEN PLANT-RETAIL  PLANT   EAST   6123030        MU PSZ CHS FIX

            class_code class_description produced?     total       ...        \
         0          20          TOSTITOS         P  28944.60       ...
         1           3            CT PUF         P  13146.95       ...
         2          90            TO DIP        NP   6687.25       ...
         3          85        MULTI PACK        NP   6417.00       ...
         4           6            RG SNK         P   5578.55       ...

            in_transit  in_building  daily_demand   total.1  cycle.1  safety_stock.1  \
         0     3971.00     24973.60       5491.95  97115.65  1166.60        19033.25
         1     3968.65      9178.30       3106.75  61764.40   892.50         7833.60
         2     1589.30      5097.95        940.70  73036.50  7691.20        12499.35
         3     1614.60      4802.40        796.95  72502.90  9202.30        17868.70
         4     3734.90      1843.65        838.95  40008.65  3463.75            0.00

             other.1  in_transit.1  in_building.1  daily_demand.1
         0  63592.05      13323.75       83791.90        18425.25
         1  34395.25      18643.05       43121.35        14595.35
         2  35486.70      17359.25       55677.25        10278.70
         3  27189.45      18242.45       54260.45         8997.60
         4   9758.00      26786.90       13221.75         6013.75

         [5 rows x 23 columns]
```

1.2.7 3.6 Distribution Costs (distribution_cost_cleaned.xlsx)

Overview: This Excel workbook contains data regarding the cost of transportation of various products from one warehouse to another. The first worksheet ("cost_calculations") combines information all of the other worksheets and illustrate the relationship.

"shipment_quantities" worksheet This worksheet contains the total amount of each product shipped from one warehouse to another in 2018.
Columns:

- BDC, product_description, department, shape, flavor, product_category: information about the product.
- source_LMSID, source_warehouse,source_region: where the product is shipped from.
- destination_LMSID, destination_warehouse, destination_type, destination_region, destination_zone: where the product is shipped to.
- cases_ordered: how many cases of the product was shipped, calculated in terms of cases used when stored in a warehouse.
- standard_cases_ordered: before loading onto a truck, all cases are converted to standard_cases for shipment calculations. This column converts the cases_ordered by a certain multiplier for shipment cost calculations.

```
In [7]: cost_workbook=pd.read_excel('project_data/distribution_cost_cleaned.xlsx',sheet_name=N
        cost_workbook.keys()

Out[7]: odict_keys(['cost_calculations', 'shipment_quantities', 'transportation_cost', 'source_

In [10]: shipment=cost_workbook['shipment_quantities']
         shipment.head()

Out[10]:       BDC               product_description        department         shape  \
         0  1003006  LVL Fritos Flavor Twist Honey BBQ  FRIED CORN CHIPS  FRITOS TWIST
         1  1003006  LVL Fritos Flavor Twist Honey BBQ  FRIED CORN CHIPS  FRITOS TWIST
         2  1003006  LVL Fritos Flavor Twist Honey BBQ  FRIED CORN CHIPS  FRITOS TWIST
         3  1005006      SVL-R Fritos Honey BBQ Twists  FRIED CORN CHIPS  FRITOS TWIST
         4  1005006      SVL-R Fritos Honey BBQ Twists  FRIED CORN CHIPS  FRITOS TWIST

                     flavor product_category  source_LMSID source_warehouse  \
         0  FCC-TW HONEY BBQ           FRITOS          6360  KILLINGLY PLANT
         1  FCC-TW HONEY BBQ           FRITOS          4350  CUCAMONGA PLANT
         2  FCC-TW HONEY BBQ           FRITOS          4350  CUCAMONGA PLANT
         3  FCC-TW HONEY BBQ           FRITOS          7310  CHARLOTTE PLANT
         4  FCC-TW HONEY BBQ           FRITOS          7310  CHARLOTTE PLANT

                           source_region  destination_LMSID  \
         0  NORTHEAST                                  6360
         1  SOUTHWEST                                  4350
         2  SOUTHWEST                                 34319
         3  CAROLINAS                                  7310
         4  CAROLINAS                                  3942
        destination_warehouse destination_type destination_region  \
0             KILLINGLY PLANT                P          NORTHEAST
1             CUCAMONGA PLANT                P          SOUTHWEST
2  MODESTO EXCHANGE WAREHOUSE                P          SOUTHWEST
3             CHARLOTTE PLANT                P          CAROLINAS
4        CHESAPEAKE(NORFOLK)DC              DC          CAROLINAS

                   destination_zone  cases_ordered  standard_cases_ordered
0                    Plant to Plant            0.0                  0.0000
1                    Plant to Plant            0.0                  0.0000
2                    Plant to Plant         6776.1               4302.8235
3                    Plant to Plant            0.0                  0.0000
4  CHESAPEAKE                               1623.6               1030.9860
```

"transportation_cost" worksheet This contains the cost of transporting one standard case of a product from one warehouse to another.

Columns:

- source_LMSID, source_warehouse: where the product is shipped from.
- destination_LMSID, destination_warehouse: where the product is shipped to.
- transportation_cost_per_standard_case: the monetary amount needed for transportation based on trucking distance, not counting the cost of sending and receiving the shipment by the source and destination warehouses.

```
In [12]: transportation_cost=cost_workbook['transportation_cost']
         transportation_cost.head()
```

```
Out[12]:    source_LMSID source_warehouse  destination_LMSID    destination_warehouse  \
         0          1310     BELOIT PLANT                666        OAK FOREST IL DC
         1          1310     BELOIT PLANT               1224       CAROL STREAM IL DC
         2          1310     BELOIT PLANT               1232            SUMMITT IL DC
         3          1310     BELOIT PLANT               1310             BELOIT PLANT
         4          1310     BELOIT PLANT               1720  DAVENPORT DC (UNASST DF)

            transportation_cost_per_standard_case
         0                               0.111462
         1                               0.067900
         2                               0.097134
         3                               0.000000
         4                               0.124585
```

"source_warehouse_cost" worksheet This contains the cost needed to send a case of a product from a warehouse.

Columns:

- source_LMSID, source_warehouse: where the product is shipped from.
- source_cost_per_case: the cost incurred at the source warehouse for preparing the shipment.

```
In [13]: source_cost=cost_workbook['source_warehouse_cost']
         source_cost.head()
```

```
Out[13]:    source_LMSID   source_warehouse  source_cost_per_case
         0          1310       BELOIT PLANT              0.264789
         1          1320      WOOSTER PLANT              0.315841
         2          1390    FRANKFORT PLANT              0.280322
         3          2310  SAN ANTONIO PLANT              0.333829
         4          2350       TOPEKA PLANT              0.353731
```

"destination_warehouse_cost" worksheet This contains the cost needed to receive a case of a product at a warehouse.
Columns:

- destination_LMSID, destination_warehouse: where the product is shipped to.
- destination_cost_per_case: the cost incurred at the receiving warehouse for taking the shipment, and unpacking it for storage.

```
In [14]: destination_cost=cost_workbook['destination_warehouse_cost']
         destination_cost.head()
```

```
Out[14]:    destination_LMSID      destination_warehouse  destination_cost_per_case
         0                666          OAK FOREST IL DC                        0.4
         1               1107        GRAND RAPIDS MI DC                        0.4
         2               1111     STERLING HEIGHTS MI DC                        0.4
         3               1132              LANSING MI DC                        0.4
         4               1141  SOUTHGATE MI DC (  APCS )                       0.4
```

1.2.8 3.7 Manufacturing Capacity (manufacturing_platform_capacity.csv)

Overview: This file contains information about how each product is manufactured (which is referred to as the manufacturing platform) and what percentage of capacity that platform is currently operating at. For platforms over 95% capacity, you should add another percentage of cost (or reduce the margins by another percent).
Columns:

- brand, business_unit, BDC, BDC_description: information about the product.
- manufacturing_platform: the process by which the product is manufactured by FritoLay. If the product is not manufactured but only packed by FritoLay, it is called "Copack."
- platform_capacity: what percentage of maximum capacity is the given manufacturing_platform operating at.

```
In [28]: platform=pd.read_csv('project_data/manufacturing_platform_capacity.csv')
         platform.head()
```

```
Out[28]:        brand  business_unit      BDC  BDC_description manufacturing_platform  \
         0  Baken-Ets              4  4148051   1.69 PL BN STP                 Copack
         1  Baken-Ets              4  4026546  1.69 SHP BN SWT                 Copack
         2  Baken-Ets              4  4026051  1.89 BN HT SPCY                 Copack
```

```
3  Baken-Ets              4  4026011      1.89 BN REG              Copack
4  Baken-Ets              4  4148011   1.89 PL STP BN              Copack

   platform_capacity
0                0.8
1                0.8
2                0.8
3                0.8
4                0.8
```

1.2.9 3.8 Product Data

Unfortunately, none of the following files match the sales data exactly (there are BDCs which are present here and not there, and vice versa). However, they do match for most of the products.

3.8.1 Price to Chains (prices_cleaned.csv.gz) **Overview:** This contains (sanitized) pricing data for all products, at a snapshot in time in 2018.

Columns:

- division: East or West.
- GTIN, BDC, UPC, product_code: various product identifiers.
- description: a short description of the product.
- brand: the overal brand which the product falls under.
- alt_description: a more informative description (which matches the column under the same name in the "sales2018_products.csv.gz" file).
- price_area: for different price_areas, the price may differ for the same product.
- price_to_store: the price FritoLay charges to each retail store. This is a sanitized estimate. True prices may be subject to individual contracts with a retailer.
- price_on_bag: the price each retail store charges customer without any price promotion. (True price may be subject to promotions.)
- ounces: the weight of each unit of the product in ounces.
- carton_type: how the product is packaged in boxes.
- count_per_carton: how many units are put into each box.
- price_per_carton: the price_to_store multiplied by the count_per_carton.
- cube: the volume each carton takes in cubit feet (for use in storage).
- shelf_life: the number of days before the product expire from the time of manufacture.

```
In [4]: prices=pd.read_csv('project_data/prices_cleaned.csv.gz')
        prices.head()
```

```
Out[4]:   division         GTIN    UPC      BDC  product_code      description  \
        0     West  28400161848  16184  1005006       7178301   .50 FR HNY BBQ
        1     West  28400161855  16185  1005007       7178401   .50 FR CHI CHS
        2     West  28400161862  16186  1005011       7178501   .50 REG FRITOS
        3     West  28400161879  16187  1005182       7178601    .50 FR TURBOS
        4     West  28400047944   4794  1009007       7731901  LSS FRITO CHILI

             brand                              alt_description price_area  \
0     Fritos  Fritos Flavor Twists Honey BBQ Flavored Corn S...       NATL
1     Fritos  Fritos Chili Cheese Flavored Corn Chips 1.125 ...       NATL
2     Fritos  Fritos The Original Corn chips 1.125 Ounce Pla...       NATL
3   Sabritas  Sabritas Turbos Flamas Flavored Corn Snacks 1...       NATL
4     Fritos  Fritos Chili Cheese Corn Chips 2.0 Ounce Plast...        EOR

   price_to_store  price_on_bag  ounces carton_type  count_per_carton  \
0        0.395025        0.5000   1.125         REG                48
1        0.395025        0.5000   1.125         REG                48
2        0.395025        0.5000   1.125         REG                48
3        0.395025        0.5000   1.125         REG                48
4        0.437324        0.4537   2.000         VFS                64

   price_per_carton  cube shelf_life
0         18.961200  1.27         70
1         18.961200  1.27         70
2         18.961200  1.27         70
3         18.961200  1.27         70
4         27.988709  2.02         70
```

3.8.2 Estimated Margins (estimated_margins.csv) **Overview:** A very rough estimate of how much profit margins FritoLay has for each product (which have been altered to product company secrets). This represents a national average does not account for differential transportation cost in each region.

Columns:

- brand: the overall brand the product falls under.
- BDC, description: a descriptive name for the product.
- estimated_magins: the estimated profit margins of FritoLay for this product. This is a very rough estimate based on the main ingredient of the product as well as the business unit.
- business_unit_code: generally equal to the BDC divided by a million.

```
In [41]: margins=pd.read_csv('project_data/estimated_margins.csv')
         margins.head()
```

```
Out[41]:            brand                                     description  \
         0  Baked Cheetos                         XL Baked Cheetos Crunchy
         1  Baked Cheetos                XL Baked Cheetos Crunchy Flamin Hot
         2  Baked Cheetos                      XXVL Baked Cheetos Flamin' Hot
         3  Baked Cheetos          XXVL Baked Cheetos Flamin' Hot Stick Strip
         4  Baked Cheetos  SVL Baked Cheetos Crunchy Whole Grain Rich Fla...

                  BDC  estimated_margins  business_unit_code
         0  172015026               0.15               172.0
         1  172015071               0.15               172.0
         2  172026071               0.15               172.0
         3  172148071               0.15               172.0
         4  172012651               0.15               172.0
```

3.8.3 Distribution to stores (percent_of_stores_carrying_by_region.csv) **Overview:** A rough estimate of what percentage of stores in each region carries each product in 2018.

Columns:

- region: the sales region.
- BDC, description: identifying information for the product.
- strategy: whether the product is an option chosen for the region, an innovation, an ethnic product, and in/out (seasonal product), an experimental product, etc.
- pct_stores_carrying: an estimate of the percentage of all stores in the region carrying that product. A low percentage implies that the local sales people are not putting these on shelves (which would explain a low sales volume).

```
In [27]: carrying=pd.read_csv('project_data/percent_of_stores_carrying_by_region.csv')
         carrying.head()
```

```
Out[27]:       region                   description         BDC       strategy  \
         0  Carolinas            Bulk Tostito 16.0  20050016.0  Region Option
         1  Carolinas                  Flush Items  98001001.0  Region Option
         2  Carolinas  Food/Service Unflav Ruffles  12050011.0  Region Option
         3  Carolinas      LSS Cheddar & SC Ruffles  12010045.0  Region Option
         4  Carolinas     LSS Onion Maui Style Chip  30010017.0  Region Option

            pct_stores_carrying
         0                  NaN
         1             0.176471
         2                  NaN
         3                  NaN
         4                  NaN
```

```
In [42]: carrying['strategy'].unique()
```

```
Out[42]: array(['Region Option', '(blank)', 'Club Only', 'Ethnic', 'Clip Strip',
                'Pallet', 'Shipper', 'In/Out', 'Innovation 2017',
                'Innovation 2016', 'Test Market', 'LIO (National)', 'Caddy',
                'Stick Strip', 'Stick Strips', 'Regional', 'Innovation 2018',
                'Weight Change 2018', nan], dtype=object)
```

1.2.10 3.9 Customer Demand Data for Ralphs (RalphsIRI-cleaned.csv.gz)

Overview: IRI is an independent marketing company which tracks sales to customers. This is data for all FritoLay product in the retail chain Ralphs in 2018. In contrast to the other sales data, this represents direct customer demand.

Columns:

- week: this data is indexed by week.
- GTIN, description, packaging, ounces: information about the product.
- revenue: the total monetary sales of this product in all Ralphs stores in that week
- sold: the number of units sold.

- average_price: revenue divided by sold. Note that this changes from week to week due to the presence of price promotions.

```
In [43]: import pandas as pd
         ralphs=pd.read_csv('project_data/RalphsIRI-cleaned.csv.gz')
         ralphs.head()
```

```
Out[43]:          week         GTIN                              description  \
         0  2017-12-31  28400596001   CHESTERS CORN & POTATO SNACK FLAMIN HOT
         1  2017-12-31  28400087691   CHESTERS CORN & POTATO SNACK FLAMIN HOT
         2  2017-12-31  28400437741   CHESTERS CORN & POTATO SNACK FLAMIN HOT
         3  2017-12-31  28400190801       SABRITONES WHEAT SNACK CHILI & LIME
         4  2017-12-31  28400183902  CHEETOS CHEESE SNACK CHEESE 50% LESS FAT

           packaging  ounces  revenue  sold  average_price
         0       BAG   5.500  9763.00  4882       1.999795
         1       BAG   1.125    77.50   155       0.500000
         2       BAG   4.000  2834.13  1677       1.690000
         3       BAG   4.250   446.00   223       2.000000
         4       BAG   7.625  1872.30   572       3.273252
```

1.2.11 3.10 California Eaches as of March 2019

Overview: This CSV file contains the 250 SKUs that are stocked at the MANTECA DC in Calfornia as of March 2019. This snapshot can be used as a sense of the status quo of what SKUs are selected for eaches (items delivered by individual packages to small retailers, instead of by large boxes).

Columns:

- product_code: an internal product code used by warehouses, which matches that of the prices_cleaned.csv.gz file.
- description: a description of the product.
- updated_at: when the supply at the distribution center was last updated at the eaches module for picking.
- pick_face: which location this product is stored in the eaches module in the distribution center.
- capacity: the target number of units of this product at the eaches module.

```
In [5]: import pandas as pd
        eaches=pd.read_csv('project_data/california_eaches.csv')
        eaches.head()
```

```
Out[5]:    product_code                description           updated_at pick_face  \
        0       8074901    (CF025) 1.89 SC HVST CHD  03/08/2019 02:41:01   BAY933F
        1       8838501      (COL 053) .50 BBQ LAYS  01/17/2019 04:44:52   BAY962F
        2       8975101     (096) 2.00 CS HOT FRIES  03/08/2019 02:42:11   BAY362F
        3       6614001    (194) 2.29 BLND SANTITAS  03/08/2019 02:41:00   BAY252F
        4       5009901  (CF125)  .50 JAPANESE NUTS  03/08/2019 02:42:32  BAY1122F

           capacity
```

```
0         10
1         10
2         10
3         10
4          4
```