What are WIC households' total expenditures on whole wheat bread?

Import libraries and connect to database

In [ ]:
```
# pandas-related imports
import pandas as pd

# database interaction imports
from pyathenajdbc import connect
```

In [ ]:
```
conn = connect(s3_staging_dir = 's3://usda-iri-2019-queryresults/',
               region_name = 'us-gov-west-1',
               LogLevel = '0',
               workgroup = 'workgroup-iri_usda')
```

Products data

Begin to familiarize yourself with the products in the data, using table `iri_usda.pd_pos_all` .

In [ ]:
```
query = "select distinct category from iri_usda.pd_pos_all"
category_df = pd.read_sql(query, conn)
category_df.head()
```

Since our questions surround purchasing of a specific type of product (we use whole wheat bread, but you can change that), let's modify the SQL query using `like` to look for bread products.

In [ ]:
```
bread_query = "select distinct * from iri_usda.pd_pos_all where category
like '%BREAD%' limit 1000"
```

Let's take a moment to see how we would modify the query for other foods, using the `format` function.

In [ ]:
```
query_input = 'GRAPES'
query_template = """select distinct * from iri_usda.pd_pos_all
where category like '%{}%' limit 1000""".format(query_input)
print(query_template)
```

Back to Bread.

In [ ]:
```
bread_query = "select distinct * from iri_usda.pd_pos_all where category
like '%BREAD%' limit 1000"

bread_df = pd.read_sql(bread_query, conn)
bread_df.head()
```

It appears that the flavor field might also be helpful to us. Let's look at what's in there. Remember, this is only for the first 1000 rows of the data, so we may need to iterate on this.

In [ ]:
```
# bread_df.columns.values
# bread_df['style'].unique()
```

In [ ]:
```
bread_df.flavor.unique()
```

When we inspect the list see that '100% whole wheat bread' is indicated by a bunch of different flavors, for example, `whole rye` , `whole wheat` , `9 grain` . But, we also see that there are a lot of categories. Let's see how many.

In [ ]:
```
bread_flavors = bread_df.flavor.unique()
len(bread_flavors)
```

Let's modify our original bread_df query to filter only for category of `FRESH BREAD & ROLLS` to see if it narrows the list of flavors.

In [ ]:
```
bread_query = "select distinct * from iri_usda.pd_pos_all where category
= 'FRESH BREAD & ROLLS'"

bread_df = pd.read_sql(bread_query, conn)
bread_flavors = bread_df.flavor.unique()
```

It took some flavors off the list. Let's come back to the full dataframe and see if the other columns can help us. to work with those.

In [ ]:
```
bread_df['product'].unique()
```

We can go back and refine the query using the `product` field.

In [ ]:
```
bread_query = """
SELECT distinct upc, flavor,upcdesc
from iri_usda.pd_pos_all
where upcdesc like '%100%'
and product in ('FRESH BREAD','HAMBURGER AND HOT DOG BUNS','PITA BREA
D','BAGELS/BIALYS','BREAD','ROLL','BUN'
    ,'BAGEL')
and category = 'FRESH BREAD & ROLLS';"""

bread_df = pd.read_sql(bread_query, conn)
```

In [ ]:
```
len(bread_df.upc.unique().tolist())
```

And again let's take a look at the flavors and do some pattern matching to try to find whole wheat items. When we inspect the list, you can see that 'whole wheat bread' is indicated by a bunch of different flavors, for example, `whole rye` , `whole wheat` , `9 grain` . Let's identify some terms that might help us make a more focused query.

In [ ]:
```
ww_flavor_terms = ['WHOLE','WHEAT','WHOLE WHEAT','GRAIN','OAT']
ww_df = bread_df[bread_df.flavor.str.contains('|'.join(ww_flavor_terms
))]
```

Now we get the UPC codes.

In [ ]:
```
ww_upc_list = ww_df.upc.unique().tolist()
len(ww_upc_list)
```

Purchasing Datasets

We're going to use a cohort that looks only at 2017 and households that are wic-participating or wic-eligible. The table name is `project_q2_cohort` and the schema name is `iri_usda_2019_db` .

In [ ]:
```
wic_hh_query = """
SELECT distinct panid
from iri_usda_2019_db.project_q2_cohort
where projection61k > 0 and wic_june = 1;"""
wic_hh_df = pd.read_sql(wic_hh_query, conn)
```

We will want to join household level data to purchasing data, which is in the `trip_all` table. Let's read that data in.

In [ ]:
```
wic_hh_list = wic_hh_df.panid.unique().tolist()
```

In [ ]:
```
trip_query = """
SELECT distinct purdate,panid,mop,upc,dollarspaid
from iri_usda.trip_all
where year = '2017' and
panid in {} and
upc in {};""".format(tuple(wic_hh_list),tuple(ww_upc_list))
```

In [ ]:
```
bread_trip_df = pd.read_sql(trip_query,conn)
```

In [ ]:
```
joined = pd.merge(wic_hh_df,bread_trip_df, on = 'panid')
```

In [ ]:
```
wic_purchases = joined.loc[joined.mop == '7']
wic_purchases.shape
```

In [ ]:
```
len(wic_purchases.panid.unique().tolist())
```

In [ ]:
```
wic_purchases['month'] = wic_purchases['purdate'].apply(lambda x: x.mont
h)
```

In [ ]:
```
wic_purchases_sub = wic_purchases[['month','dollarspaid']]
```

In [ ]:
```
wic_purchases_agg = wic_purchases_sub.groupby(['month']).sum()
wic_purchases_agg
```

In [ ]:
```
Import visualization packages
import matplotlib.pyplot as plt
import seaborn as sn

# so images get plotted in the notebook
%matplotlib inline
```

In [ ]:
```
ax = wic_purchases_agg.plot(figsize = (12, 6))
ax.set(ylabel = '$ Spent', title = '$ Spent by WIC-Households on 100% Wh
ole Wheat Bread by month, 2017')
ax.get_legend().remove()
plt.annotate('Sources: IRI Consumer Network and InfoScan',
             xy=(0.75,-0.1));
```