UMD Data Challenge

David Tonne and Ariana Bryant
DC21049

Introduction

- First year students with no experience in data science
- Decided to participate because it sounded like an interesting challenge
- Used MATLAB because it was the only data processor we have any experience with

Problem Statement

We set out to answer the questions:

- What are the most popular ingredient in each meal category?
- What are the common combinations appearing together?

In order to:

- Help food scientists improve health and longevity of packaged foods

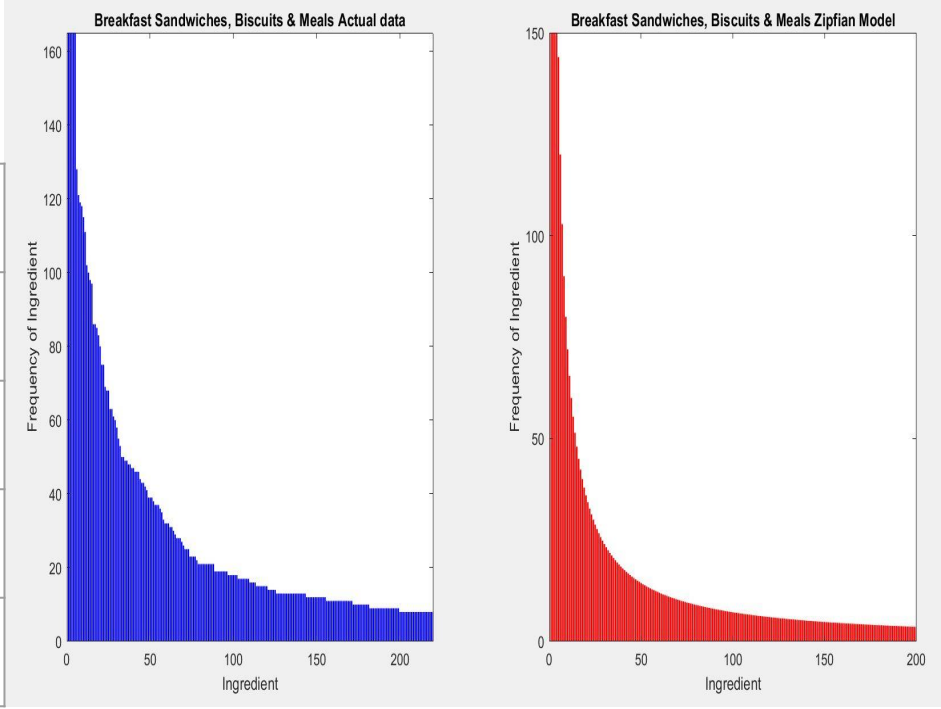
Dataset

- Spreadsheet with messy ingredients formatting
- Mixed capitalization and a variety of list formatting
- No quantitative data on the ingredient makeup
- We made all ingredients lowercase
- Removed parentheses, brackets, semicolons, and varieties of “contains 2% or less of:”

Common Ingredients (Breakfast Sandwiches and Meals)

- 238 food items
- 10612 total ingredients

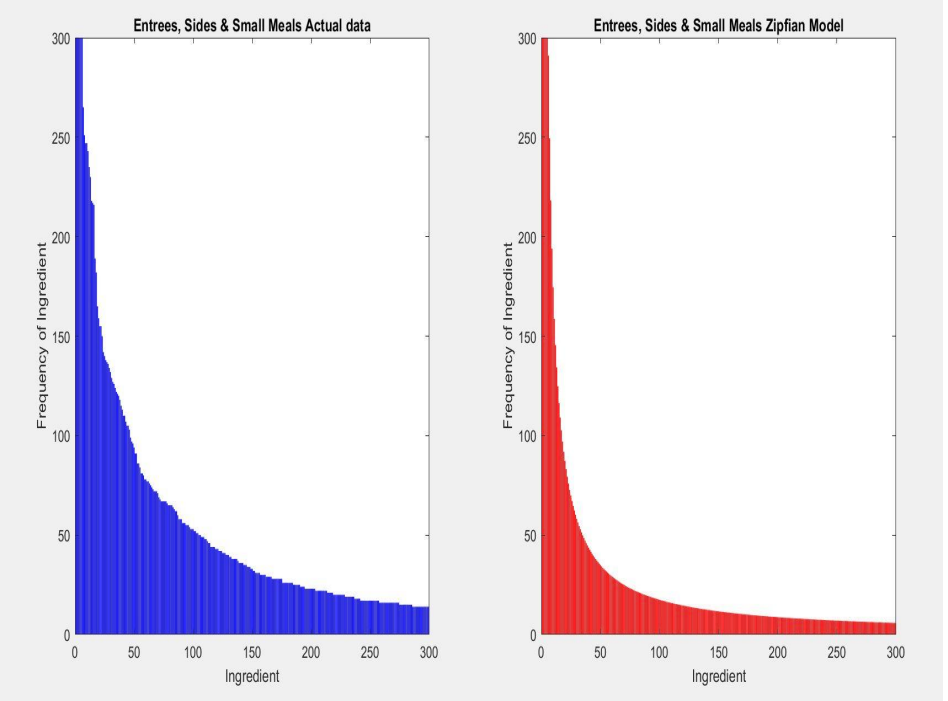
Salt	720
Water	410
Sugar	265
Enzymes	220
Citric acid	188



Common Ingredients (Entrees, Sides, and Small Meals)

- 1410 food items
- 27354 total ingredients

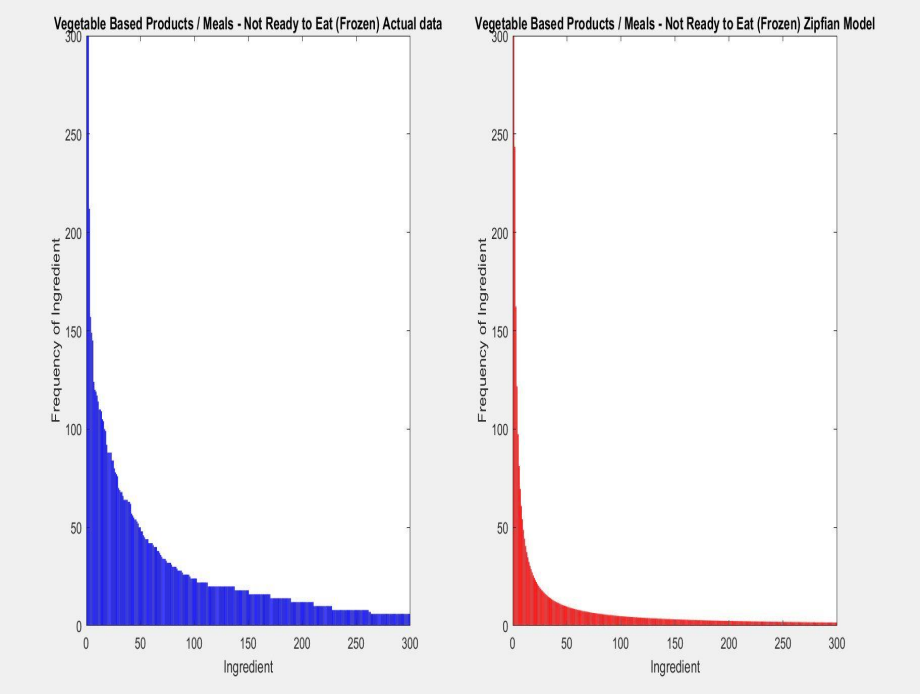
Salt	1746 6.4%
Water	1094 4.0%
Sugar	440 1.6%
Citric acid	430 1.6%
Maltodextrin	365 1.3%



Common Ingredients (Frozen Vegetarian Meals)

- 227 food items
- 10388 total ingredients

Water	487 4.7%
Salt	448 4.3%
Wheat gluten	212 2.0%
Spices	157 1.5%
Garlic Powder	149 1.4%



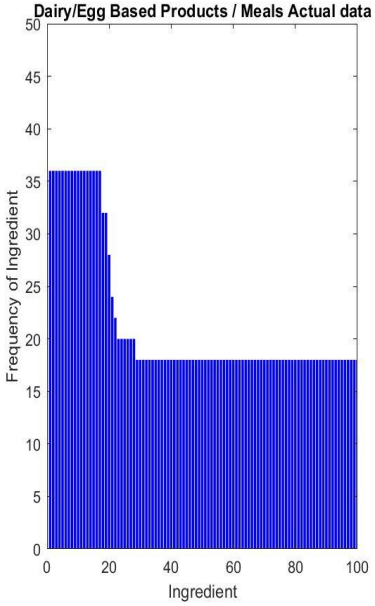
Conclusions

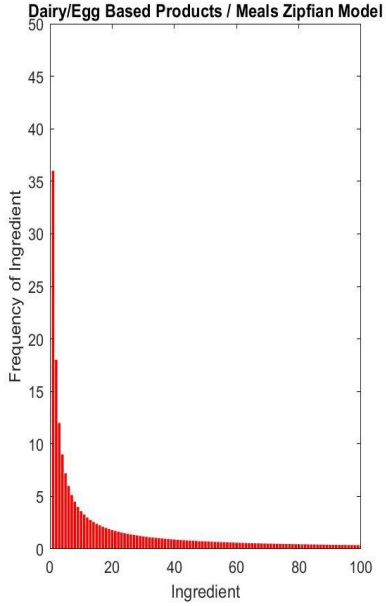
- High amounts of salt and sugar can be harmful
- Water is ubiquitous
- Common flavorings include citric acid and garlic
- Methodology is limited in ability to clean up data
- Lacking in specific quantities of ingredients

Common Ingredient Pairs (Dairy Based Products)

- 29 food items
- 11690 total ingredient pairs

Color, Sulfate	36 0.3%
Egg white, Sulfate	36 0.3%
Beta carotene, Sulfate	36 0.3%
Onion powder, Sulfate	36 0.3%
Salt, Sulfate	36 0.3%

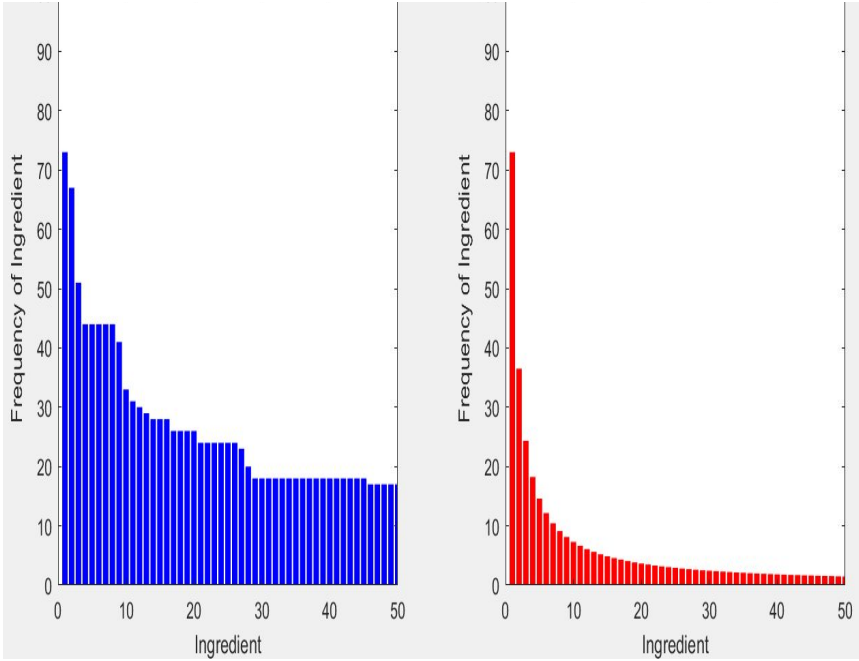




Common Ingredient Pairs (Savoury Dough Meals Not Ready to Eat)

- 8 food items
- 14183 total ingredient pairs

Salt, Salt	73 0.5%
Salt, Enzymes	67 0.5%
Salt, Cheese cultures	51 0.4%
Salt, Folic acid	44 0.3%
Salt, Niacin	44 0.3%



Conclusions

- Used nested for loops, processing time increases with n!
- Limited to performing analysis on smallest categories
- Data seems to suggest that there are not any common combinations to target
- Small categories may also mean these results are not useful

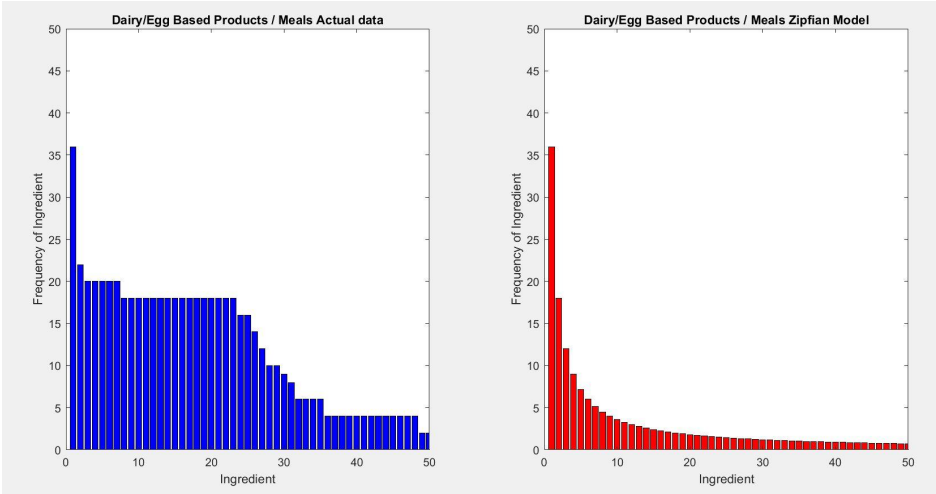
Future Possibilities

- Standardized data collection would greatly expedite analysis
- Quantitative data on ingredients would allow for more in depth conclusions
- Possibly develop less cumbersome method for ingredient combinations, or use more computing power
- Get more information on food science to make more informed conclusions

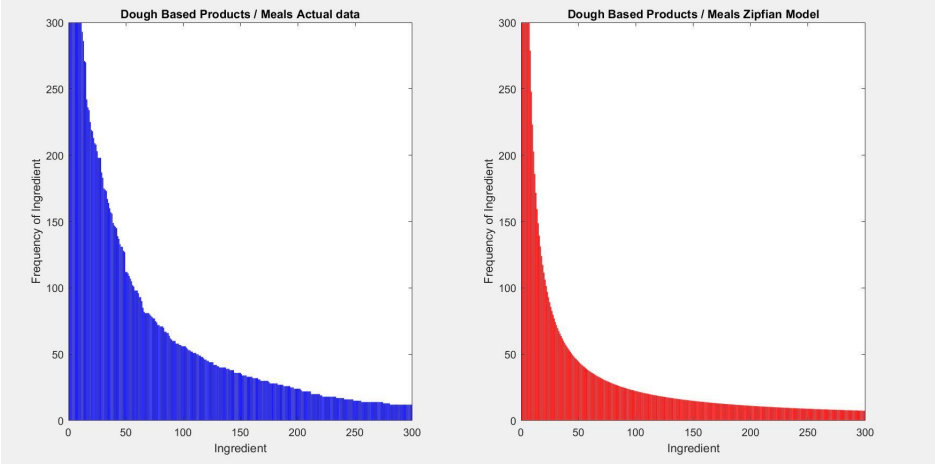
Thank you for your time!

Any questions?

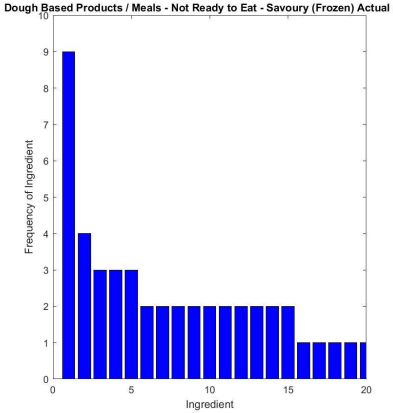
Extraneous Slides

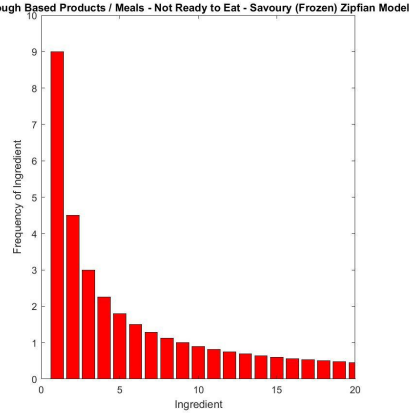

Dairy/Egg Based Products / Meals Actual data
Frequency of Ingredient
Ingredient
0
5
10
15
20
25
30
35
40
45
50
0
10
20
30
40
50

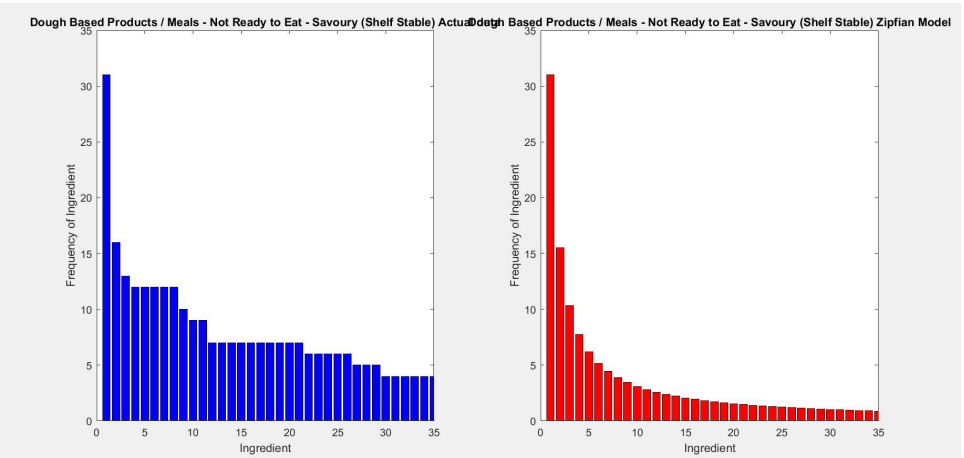
Dairy/Egg Based Products / Meals Zipfian Model
Frequency of Ingredient
Ingredient
0
5
10
15
20
25
30
35
40
45
50
0
10
20
30
40
50


Dough Based Products / Meals Actual data
Frequency of Ingredient
Ingredient
0
50
100
150
200
250
300
0
50
100
150
200
250
300

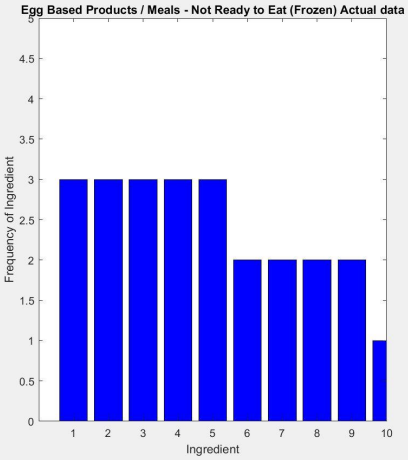
Dough Based Products / Meals Zipfian Model
Frequency of Ingredient
Ingredient
0
50
100
150
200
250
300
0
50
100
150
200
250
300

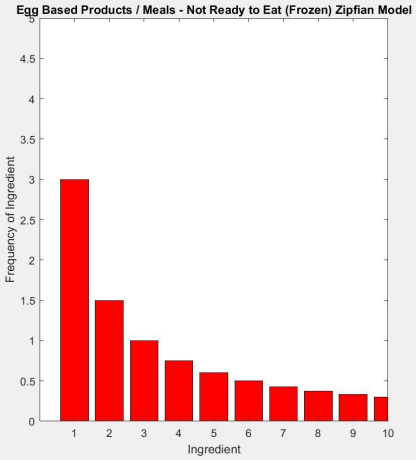

Dough Based Products / Meals - Not Ready to Eat - Savoury (Frozen) Actual data
Frequency of Ingredient
Ingredient
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20

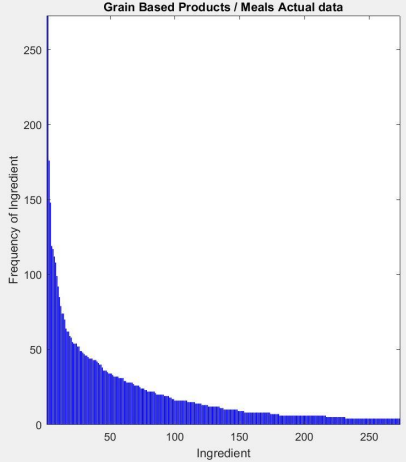

Dough Based Products / Meals - Not Ready to Eat - Savoury (Frozen) Zipfian Model
Frequency of Ingredient
Ingredient
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20

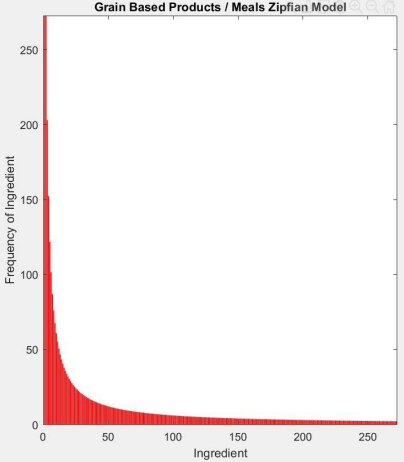

Dough Based Products / Meals - Not Ready to Eat - Savoury (Shelf Stable) Actual data
Frequency of Ingredient
Ingredient
0 5 10 15 20 25 30 35
0 5 10 15 20 25 30 35

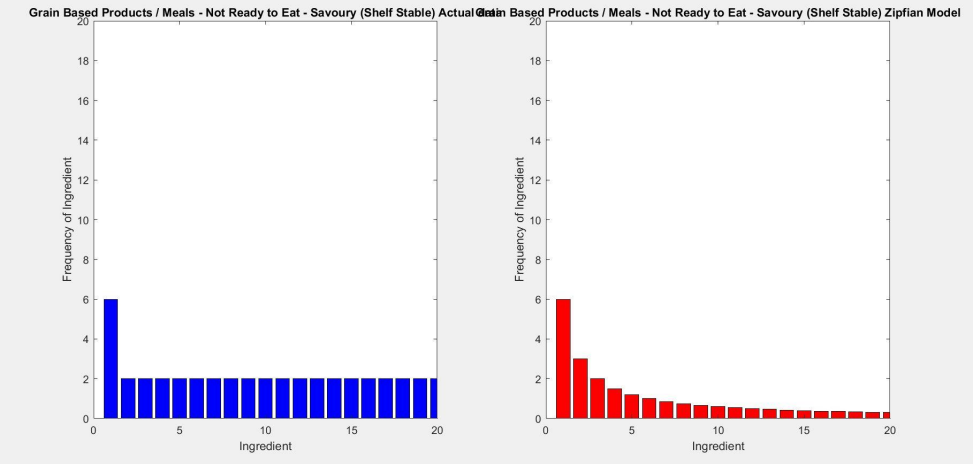
Dough Based Products / Meals - Not Ready to Eat - Savoury (Shelf Stable) Zipfian Model
Frequency of Ingredient
Ingredient
0 5 10 15 20 25 30 35
0 5 10 15 20 25 30 35


Egg Based Products / Meals - Not Ready to Eat (Frozen) Actual data
Frequency of Ingredient
Ingredient
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
1 2 3 4 5 6 7 8 9 10

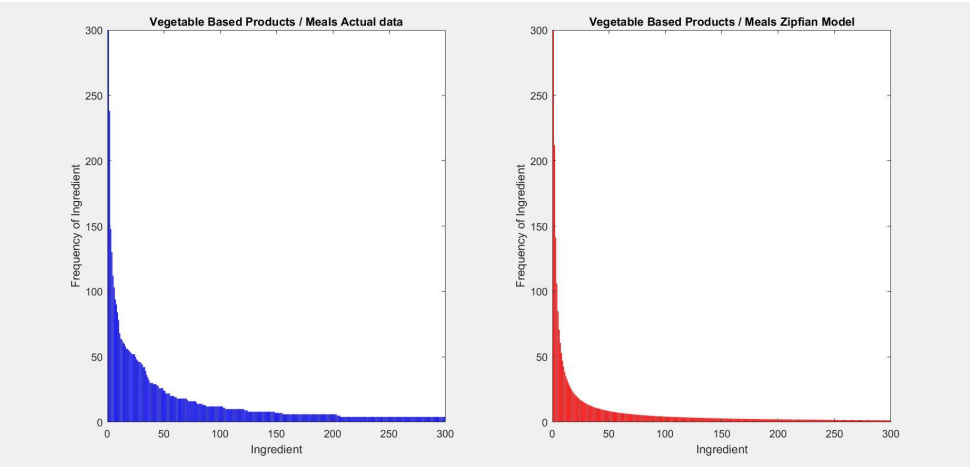

Egg Based Products / Meals - Not Ready to Eat (Frozen) Zipfian Model
Frequency of Ingredient
Ingredient
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5
1 2 3 4 5 6 7 8 9 10


Grain Based Products / Meals Actual data
Frequency of Ingredient
Ingredient
0 50 100 150 200 250
50 100 150 200 250

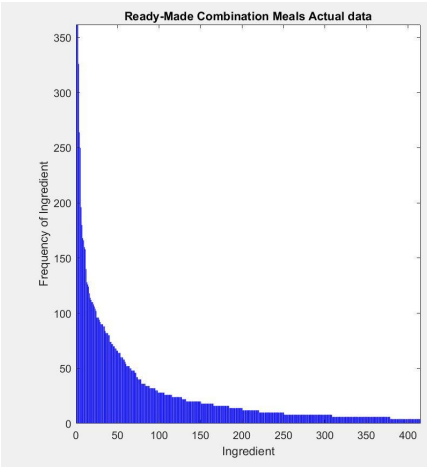

Grain Based Products / Meals Zipfian Model
Frequency of Ingredient
Ingredient
0 50 100 150 200 250
0 50 100 150 200 250

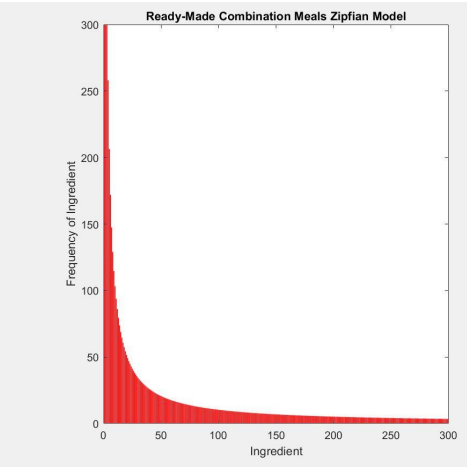

Grain Based Products / Meals - Not Ready to Eat - Savoury (Shelf Stable) Actual data
Frequency of Ingredient
Ingredient

Grain Based Products / Meals - Not Ready to Eat - Savoury (Shelf Stable) Zipfian Model
Frequency of Ingredient
Ingredient


Vegetable Based Products / Meals Actual data
Frequency of Ingredient
Ingredient

Vegetable Based Products / Meals Zipfian Model
Frequency of Ingredient
Ingredient


Ready-Made Combination Meals Actual data
Frequency of Ingredient
Ingredient


Ready-Made Combination Meals Zipfian Model
Frequency of Ingredient
Ingredient

Unique Ingredient Code

```
function [] = uniqueIng(a, xscale, yscale)
% Determines most common individual ingredients in a single category of
% food type
raw = importfile('Packaged Meals Dataset.xlsx');% Importing data
% Extracting the categories of food types
categories = unique(raw.branded_food_category);
% Pulling out the relevant data to the specific category
section = raw(raw.branded_food_category == categories(a), :);
% Creating a cell of string arrays for each ingredients list
ing = cell(size(section.ingredients));
for i = 1:size(section.ingredients)
    ing{i} = split(section.ingredients(i),',');
end
% Separating each cell into a long array of all ingredients
ingtot = strings(1, 177277);
k = 1;
for i = 1:size(ing)
    for j = 1:size(ing{i})
        ingtot(k) = ing{i}(j);
        k = k + 1;
    end
end
ingtot = ingtot(~(ingtot==''));
% Checking list of all ingredients against unique ingredient array and
% counting the number of occurences
uniqueIngreds = unique(ingtot);
count = zeros(size(uniqueIngreds));
for i = 1:size(uniqueIngreds, 2)
    count(i) =  sum(strcmp(uniqueIngreds(i), ingtot));
end
```

```
function [] = combos(a, xscale, yscale)
% Determines the frequency of unique combinations of ingredients from
% single food items in a category
raw = importfile('Packaged Meals Dataset.xlsx');% Importing data
categories = unique(raw.branded_food_category);% Extracting the categories of food types
% Pulling out the relevant data to the specific category
section = raw(raw.branded_food_category == categories(a), :);
% Creating a cell of string arrays for each ingredients list
ing = cell(size(section.ingredients));
for i = 1:size(section.ingredients)
    ing{i} = split(section.ingredients(i),',');
end

% Creating array of all combination of ingredients in each food
combo = strings(1, nchoosek(length(ing)*10, 2));
k = 1;
for i = 1:length(ing)
    transient = ing{i};
    for j = 1:length(transient)
        for w = j+1:length(transient)
            combo(k) = transient(j) + transient(w);
            k = k+1;
        end
    end
end
combo = combo(~(combo==''));

% Checking the list of combinations against the array of unique
% combinations and counting the number of occurences
uniqueCombo = unique(combo);
count = zeros(size(uniqueCombo));
for i = 1:size(uniqueCombo, 2)
    count(i) =  sum(strcmp(uniqueCombo(i), combo));
end
```