1 (a)

------------**RECIPE 1: White Sauce Pasta**------------------------------

Ingredients section:

2 cups, fusilli, cooked

2 cups, milk

2 tbsp, plain flour

2 tbsp, butter

2 tsp, garlic, finely chopped

¼ cup, yellow capsicum, striped

¼ cup, green capsicum, striped

¼ cup, red capsicum, striped

¼ cup, zucchini, sliced

¼ cup, broccoli florets, blanched

¼ cup, baby corn, diagonally cut and balanced

1 tsp, red chili flakes, dried

1 ½ tsp, mixed herbs, dried

¼ cup, processed cheese, grated

Cooking Instructions:

1. Combine the milk, plain flour, and salt in a deep bowl and whisk well till no lumps remain. Keep aside.
2. Heat the butter in a broad non-stick pan, add the garlic and saute on a medium flame for a few seconds.
3. Add the yellow, green, and red capsicum and zucchini and saute on a medium flame for two minutes.
4. Add the broccoli and baby corn and saute on a medium flame for one more minute.
5. Add the milk-plain flour mixture, chili flakes, mixed herbs, cheese, and a little salt, mix well and cook on a medium flame for two minutes while stirring continuously.
6. Add the fusilli, mix well and cook on a medium flame for one to two minutes while tossing gently. Serve white sauce pasta immediately.

------------------------**RECIPE 2: Upma**------------------------------

Ingredients section:

1 cup, semolina

1 ½ tbsp, oil

½ tsp, mustard seeds

1 tsp, split black lentils

4, curry leaves

2, green chilies, slit

¼ cup, onions, finely chopped

To taste, salt

2 tsp, lemon juice

2 tsp, sugar

1 tbsp, coriander, finely chopped

Cooking Instructions:

1. Heat the semolina in a kadhai and dry roast on a medium flame for four to five minutes, while stirring occasionally. Keep aside.
2. Heat the oil in a kadhai and add the mustard seeds.
3. When the seeds crackle, add the black lentils, curry leaves, and green chilies and saute on a medium flame for a few seconds.
4. Add the onions and saute on a medium flame for one to two minutes.
5. Add the roasted semolina, three cups of hot water, and salt and mix well, cover with a lid, and cook on a slow flame for three to four minutes while stirring occasionally.
6. Add the lemon juice and sugar, mix well and cook on a medium flame for another one minute while stirring constantly.
7. Fill the upma in a glass bowl and demould on a plate.
8. Serve the upma immediately garnished with coriander.

-------------------**RECIPE 3: Lemonade**------------------------------

Ingredients section:

4 tbsp, lemon juice

8 tbsp, sugar, powdered

1 tsp, black salt

2 tsp, cumin seeds, powdered

½ tsp, salt

Cooking Instructions: Combine all the ingredients in a deep bowl and mix well. Add four cups of chilled water and mix well. Serve the lemonade chilled.

----------------**RECIPE 4: Poha Chivda**-----------------------------

Ingredients section:

3 cups, beaten rice, thin

2 tbsp, oil

1 tsp, mustard seeds

4 to 6, curry leaves

¼ cup, peanuts, raw

¼ cup, chana dal, roasted

2 tbsp, cashew nut halves

½ tsp, turmeric powder

1 tsp, chili powder

1 tbsp, sugar, powdered

To taste, salt

Cooking Instructions:

1. Heat a deep non-stick pan and dry roast the thin beaten rice on a medium flame for two to three minutes while stirring continuously or till crisp. Keep aside.
2. Heat the oil in a deep non-stick pan and add the mustard seeds and curry leaves.
3. Once the seeds crackle and the peanuts, roasted chana dal, and cashew nuts, mix well and saute on a medium flame for two to three minutes or till they turn light brown.
4. Add the turmeric powder, mix well and saute on a medium flame for thirty seconds.
5. Add the roasted thin beaten rice, chili powder, powdered sugar, and salt mix well and cook on a medium flame for two to three minutes while stirring continuously. Keep aside to cool completely.
6. Store the poha chivda in an airtight container and use it as required.

-----------**RECIPE 5: Hot Garlic Sauce**-----------------------------

Ingredients section:

1 ½ tbsp, garlic, finely chopped

¼ cup, schezwan sauce

¼ cup, red chili sauce

1 tbsp, cornflour

1 ½ tbsp, oil

½ tsp, ginger, finely chopped

1 tsp, green chilies, finely chopped

½ cup, spring onions whites and greens, finely chopped

½ tsp, soy sauce

To taste, salt

Cooking Instructions:

1. Combine the cornflour and ¾ cup of water in a bowl and mix very well. Keep aside.
2. Heat the oil in a broad non-stick pan, add the garlic, ginger, and green chilies and saute on a medium flame for thirty seconds.
3. Add the spring onions, schezwan sauce, chili sauce, soy sauce, mix well, and cook on a medium flame for one minute while stirring occasionally.
4. Add the cornflour-water mixture and salt, mix well and cook on a medium flame for one minute while stirring continuously. Use the hot garlic sauce as required.

1 (b)

Recipe1 - fusilli

Recipe1 - milk

Recipe1 - plain flour

Recipe1 - butter

Recipe1 - garlic

Recipe1 - yellow capsicum

Recipe1 - green capsicum

Recipe1 - red capsicum

Recipe1 - zucchini

Recipe1 - broccoli florets

Recipe1 - baby corn

Recipe1 - red chili flakes

Recipe1 - mixed herbs

```
Recipe1 - processed cheese
Recipe2 - semolina
Recipe2 - oil
Recipe2 - black lentils
Recipe2 - curry leaves
Recipe2 - green chilies
Recipe2 - onions
Recipe2 - salt
Recipe2 - lemon juice
Recipe2 - sugar
Recipe2 - coriander
Recipe3 - lemon juice
Recipe3 - sugar
Recipe3 - black salt
Recipe3 - cumin seeds
Recipe3 - salt
Recipe4 - beaten rice
Recipe4 - oil
Recipe4 - mustard seeds
Recipe4 - curry leaves
Recipe4 - peanuts
Recipe4 - chana dal
Recipe4 - cashew nut
Recipe4 - turmeric powder
Recipe4 - chili powder
Recipe4 - sugar
Recipe4 - salt
Recipe5 - garlic
Recipe5 - schezwan sauce
```

Recipe5 - red chili sauce

Recipe5 - cornflour

Recipe5 - oil

Recipe5 - ginger

Recipe5 - green chilies

Recipe5 - spring onions whites

Recipe5 - spring onions greens

Recipe5 - soy sauce

Recipe5 - salt

1 (c)

1. In coarse-graining the recipe data, many aspects of the recipe are being lost. We do not know **the form of the ingredient** being used in the recipes. For example, chana dal can be used raw or roasted. In Recipe4, roasted chana dal is used, but the data does not give any information.
2. The details of the **processing** are missing in the data. Due to this one does not know the **time** it would take to cook the recipe.
3. We can not know when a particular ingredient needs to be cooked and combined with some other elements.
   The **amount of each ingredient** is also missing the data.
4. From the above style of data, one cannot predict the **alternatives** for any ingredient.
5. We cannot predict the **taste** of the dish. For example, in Recipe2, salt and sugar are both added. Though Upma is salty yet after seeing that sugar is also added, one might believe that it is sweet.
6. Since there is no information about the processing of ingredients, we do not know the **resources and the utensils** required for the recipes. For example, in Recipe3 above, no heating is needed, whereas, in all the other recipes, we need to heat the ingredients.

1 (d)

1. Along with storing ingredients, we should store the form of the ingredient and their amount.
2. Each recipe should attribute the time required and the resources needed like heat, refrigeration, etc.
3. Their processing instructions should also accompany the ingredients, but this will not ensure the steps in which they will be processed.
4. The recipe should have another attribute that justifies the processing details to tackle the problem presented in the previous point.
5. Every ingredient can have another attribute that stores the alternative used when the particular element is absent.

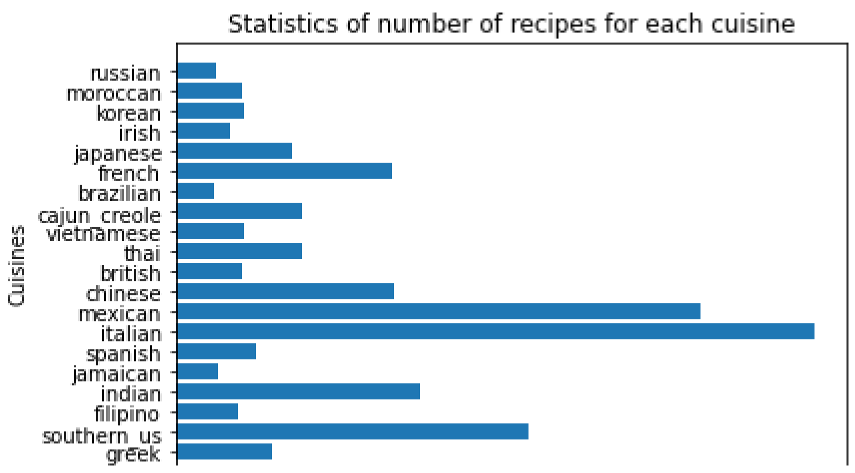
2 (a)

```
# Importing required libraries
import numpy as np
import matplotlib.pyplot as plt
import json
import operator

f = open('train.json')

# This data is list of dictionaries
data = json.load(f)

print()
# Q2
# Printing the number of recipes
print("The number of recipes:")
numOfRecipes = len(data)
print(numOfRecipes)

uniqueIngredients = []
uniqueCuisines = []
cuisine_recipes = {}
ingredient_frequency = {}

# stores the number of recipes corresponding to each recipe size
recipeSize_recipes = {}

for i in range(len(data)):
    dict = data[i]
    list_ingredients = dict['ingredients']

    # Finding out the frequency of each ingredient
    for ingredient in list_ingredients:
        if ingredient in ingredient_frequency:
            ingredient_frequency[ingredient] = ingredient_frequency[ingredient]+1
        else:
            ingredient_frequency[ingredient] = 1

    recipe_size = len(list_ingredients)
    if recipe_size in recipeSize_recipes:
        recipeSize_recipes[recipe_size] = recipeSize_recipes[recipe_size]+1
    else:
        recipeSize_recipes[recipe_size] = 1

    cuisine = dict['cuisine']

    # Finding the number of cuisines and storing number of recipes for each cuisine
    if cuisine in uniqueCuisines:
        cuisine_recipes[cuisine] = cuisine_recipes[cuisine]+1

    else:
        uniqueCuisines.append(cuisine)
        cuisine_recipes[cuisine] = 1

    # Finding the number of ingredients
    for j in range(len(list_ingredients)):
        element = list_ingredients[j]
        if element in uniqueIngredients:
            pass
        else:
            uniqueIngredients.append(element)

print()
# Printing the number of unique ingredients
print("The number of unique ingredients:")
print(len(uniqueIngredients))

print()
# Printing the number of cuisines
print("The number of cuisines:")
print(len(uniqueCuisines))
```

```
The number of recipes:
39774

The number of unique ingredients:
6714

The number of cuisines:
20
```

2 (b)

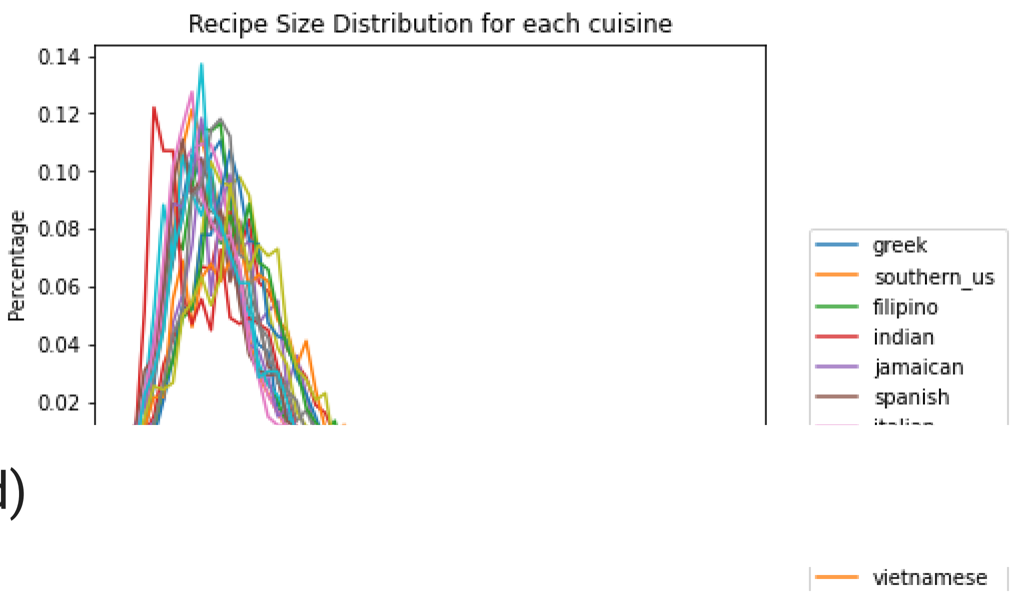
```
# Defining the measures for x and y axes
x_axis = list(cuisine_recipes.keys())
y_axis = list(cuisine_recipes.values())

# Horizontal bar plot as some names are not completely visible
plt.title("Statistics of number of recipes for each cuisine")
plt.xlabel("No. of recipes")
plt.ylabel("Cuisines")
plt.barh(x_axis, y_axis)

# Bar plot
bar_plot = plt.figure(figsize = (10,8))
plt.bar(x_axis,y_axis,color = 'green', width = 0.4)
plt.xticks(rotation = 45)
plt.xlabel("Cuisines")
plt.ylabel("No. of recipes")
plt.title("Statistics of number of recipes for each cuisine")
plt.show()
```



2 (c)

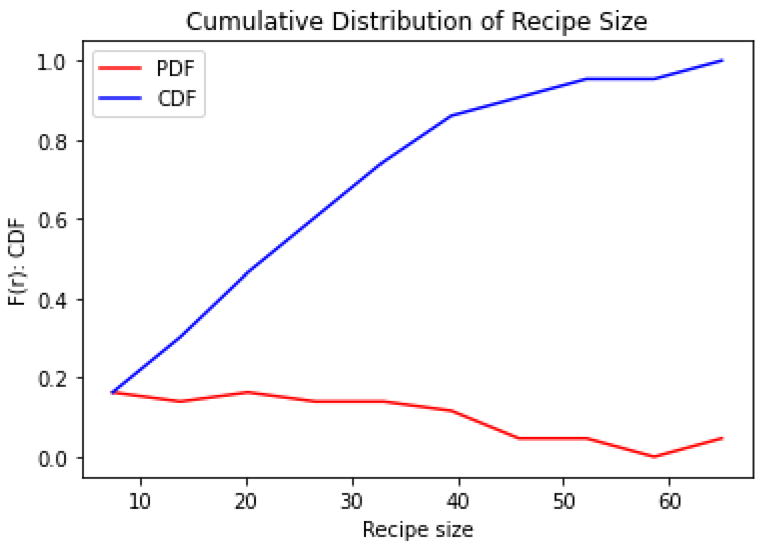
```
# Extracting recipe size and number of cuisines for each cuisine
# to distinguish every cuisine
Legend = []
i = 0

for cuisine in uniqueCuisines:

    # Stores the number of recipes corresponding to each recipe and cuisine
    recipeSize_numberOfCuisies = {}
    Legend.append(cuisine)

    # to calculate percentage of recipe size
    numOfRecipesInCuisine = 0.0
    for recipe in data:
        recipe_size = len(recipe['ingredients'])

        if cuisine==recipe['cuisine']:
            numOfRecipesInCuisine = numOfRecipesInCuisine+1
            if recipe_size in recipeSize_numberOfCuisies:
                recipeSize_numberOfCuisies[recipe_size] = recipeSize_numberOfCuisies[recipe_s
            else:
                recipeSize_numberOfCuisies[recipe_size] = 1

    # Calculating the recipe size percentage for each cuisine
    for recipe_size in recipeSize_numberOfCuisies:
        recipeSize_numberOfCuisies[recipe_size] = recipeSize_numberOfCuisies[recipe_size]/num

    sorted_list = sorted(recipeSize_numberOfCuisies.keys())
    y_list = []
    for r in sorted_list:
        y_list.append(recipeSize_numberOfCuisies[r])
    # Plotting every cuisine
    x_axis = sorted_list
    y_axis = y_list
```

```
        plt.plot(x_axis, y_axis)

    plt.xlabel("Recipe size")
    plt.ylabel("Percentage")
    plt.title("Recipe Size Distribution for each cuisine")
    plt.legend(Legend, bbox_to_anchor = (1.05, 0.6))
    plt.show()

    # Computing the percentage of each recipe size
    for recipe_size in recipeSize_recipes:
        recipeSize_recipes[recipe_size] = recipeSize_recipes[recipe_size]/float(len(data))

    # Plotting the graph
    sorted_list = sorted(recipeSize_recipes.keys())
    y_list = []
    x_axis = sorted_list
    for r in sorted_list:
        y_list.append(recipeSize_recipes[r])
    y_axis = y_list
    plt.plot(x_axis, y_axis)
    plt.xlabel("Recipe size")
    plt.ylabel("Percentage")
    plt.title("Recipe Size Distribution for all recipes")
    plt.show()
```



2 (d)

```
# Plotting CDF
cdfData = x_axis
count, bins_count = np.histogram(cdfData, bins = 10)
pdf = count/sum(count)
cdf = np.cumsum(pdf)
plt.plot(bins_count[1:], pdf, color="red", label="PDF")
plt.plot(bins_count[1:], cdf, color="blue", label="CDF")
plt.legend()
plt.xlabel("Recipe size")
plt.ylabel("F(r): CDF")
plt.title("Cumulative Distribution of Recipe Size")
plt.show()
```



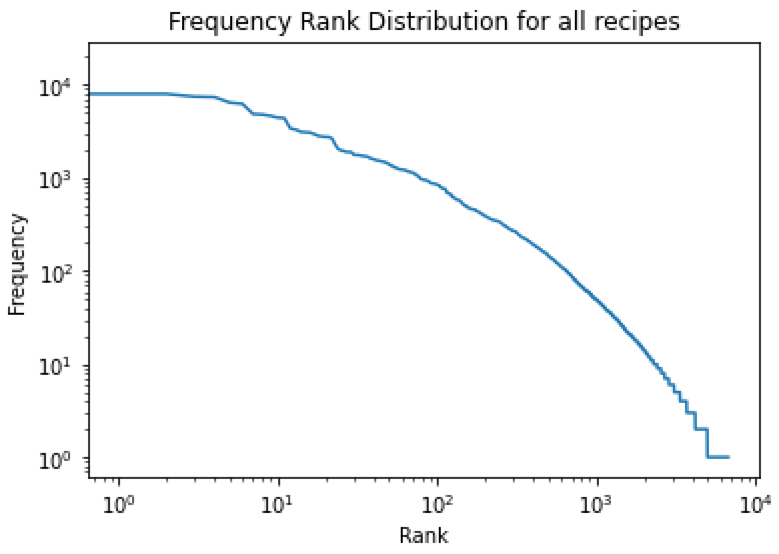
Interpretation

We can see that the plot is monotonically increasing. At the end it is flattened as there are not many data points. It tells that there not many recipes of size more than 40 because their instruction details are hard to transmit. These recipes are nuanced and complex. They are made only when there is some special occasion.

However, there is peak for the values in between 10 and 40. This is becuase they are less complex and can be cooked to eat in daily life.

3 (a)

```
# Q3 a
sorted_matrix = sorted(ingredient_frequency.items(),key=operator.itemgetter(1),reverse=True)
sorted_dict = {}
topTenIngredients = []
j = 0

for i in range(len(sorted_matrix)):
    sorted_dict[i] = sorted_matrix[i][1]
    if(j<10):
        topTenIngredients.append(sorted_matrix[i][0])
    j = j+1

# print(sorted_dict)
x_axis = list(sorted_dict.keys())
y_axis = list(sorted_dict.values())

# Line plot
plt.loglog(x_axis,y_axis)
plt.xlabel("Rank")
plt.ylabel("Frequency")
plt.title("Frequency Rank Distribution for all recipes")
plt.show()
```



Interpretation

The powerlog of frequency rank distribution is a little flattened for the lower ranks. This is because these ingredients are widely used in many recipes like salt, onions, water etc. However, it falls sharply as the ingredients with higher ranks are not frequently used due to availablitiy, cost and their taste profiles.

3 (b)

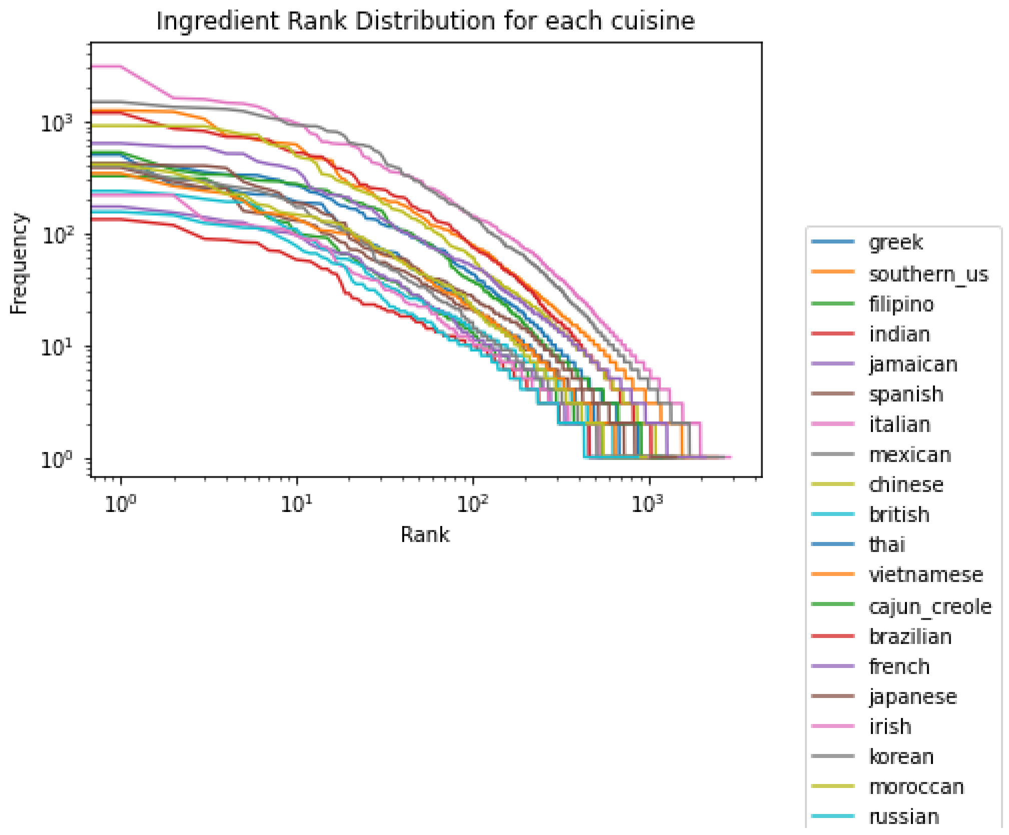
```
print()
print("The 10 most popular ingredients in the recipes are: ")
for i in range(len(topTenIngredients)):
    print(i+1,end = ". ")
    print(topTenIngredients[i])
```

```
The 10 most popular ingredients in the recipes are:
1. salt
2. onions
3. olive oil
4. water
5. garlic
6. sugar
7. garlic cloves
8. butter
9. ground black pepper
10. all-purpose flour
```

3 (c)

```
Legend = []

for cuisine in uniqueCuisines:
    # Stores the ingredient and their frequency
    Legend.append(cuisine)
    ingr_freq = {}
    for recipe in data:
        if cuisine==recipe['cuisine']:
            ingredientsUsed = recipe['ingredients']
            for ingr in ingredientsUsed:
                if ingr in ingr_freq:
                    ingr_freq[ingr] = ingr_freq[ingr]+1
                else:
                    ingr_freq[ingr] = 1
    sorted_matrix = sorted(ingr_freq.items(),key=operator.itemgetter(1),reverse=True)
    sorted_dict = {}
    topTenIngredients = []
    j = 0
    for i in range(len(sorted_matrix)):
        sorted_dict[i] = sorted_matrix[i][1]
        if(j<10):
            topTenIngredients.append(sorted_matrix[i][0])
        j = j+1

    x_axis = list(sorted_dict.keys())
    y_axis = list(sorted_dict.values())
    plt.loglog(x_axis, y_axis)
    print()
    print(cuisine)
    for ingredient in topTenIngredients:
        print(ingredient, end=", ")
    print()

plt.xlabel("Rank")
plt.ylabel("Frequency")
plt.legend(Legend, bbox_to_anchor = (1.05, 0.6))
plt.title("Ingredient Rank Distribution for each cuisine")
plt.show()
```

```
british
salt, all-purpose flour, butter, milk, eggs, unsalted butter, sugar, onions, baking pow

thai
fish sauce, garlic, salt, coconut milk, vegetable oil, soy sauce, sugar, water, garlic

vietnamese
fish sauce, sugar, salt, garlic, water, carrots, soy sauce, shallots, garlic cloves, ve

cajun_creole
salt, onions, garlic, green bell pepper, butter, olive oil, cayenne pepper, cajun seaso

brazilian
salt, onions, olive oil, lime, water, garlic cloves, garlic, cachaca, sugar, tomatoes,

french
salt, sugar, all-purpose flour, unsalted butter, olive oil, butter, water, large eggs,

japanese
soy sauce, salt, mirin, sugar, water, sake, rice vinegar, vegetable oil, scallions, gin

irish
salt, all-purpose flour, butter, onions, sugar, potatoes, baking soda, baking powder, m

korean
soy sauce, sesame oil, garlic, green onions, sugar, salt, water, sesame seeds, onions,

moroccan
salt, olive oil, ground cumin, onions, ground cinnamon, garlic cloves, water, ground gi

russian
salt, sugar, onions, all-purpose flour, sour cream, eggs, water, butter, unsalted butte
```



3 (d)

The power log of the frequency rank distribution is plotted.

1. The plot is flattened for the initial ranks. This is becuase some ingredients are way more omnipresent and used in many recipes.
2. They are used in so many recipes because they are cheap for the locals and easily accessible.
3. However, as the rank increases we see a downfall in the plot. This is becuase the frequency of use of these ingredients are not as great in number as the ingredient with smaller ranks.
4. The main reason for such a trend is the seasonal nature of the recipes. For example, Jamaican pumpkins are available from September until mid-January.
5. The cost and availability of an ingredient also affect the plot nature. For example, saffron is considered to be an expensive ingredient due to which it is not used in many recipes. Also, olive oil was not easily available in India in traditional recipes.