A. Research Question

Research Question: Does the consumption rate of various food products influence health outcomes?

Hypotheses: **H_0 :** Food consumption has no impact on health outcomes in developing nations, **H_1:** in nations with lower consumption of high-fat and high-caloric foods, there are more favorable health outcomes, **H_2:** nations with greater consumption of high-fat and high-caloric foods result in better health outcomes.

This research is designed to explore the intersection of food consumption and health outcomes in developing countries. This research analyzes foods divided into 32 categories: rice (in all forms except flour), bread (fresh bread and special bread), cheese (cheese and curd), and pasta products to name a few. Consumption rates from each category were gathered from The World Bank. The data for nutritional value was then gathered for each type of item in the category and the median of the individual food items was used to represent the category. Finally, the health outcomes for each developing country were gathered from the World Health Organization. The Health Outcomes and consumption rate datasets are from 2010 data, but the nutritional data, gathered from nutritionvalue.org, is 2020 data. This is a constraint on the accuracy of the results attained.

This research aims to achieve a predictive model for health outcomes in developing nations based on their current consumption rates. This will provide health and government officials with the information necessary to focus resources and maximize care.

B. Data Collection

The World Bank and World Health Organizations had easily accessible datasets that downloaded into comma-separated values. Initially, this project was designed to collect data from Food Data Central for nutritional values. However, Food Data Central did not have records for a large portion of the food items and would have resulted in skewed and inaccurate results. The project then shifted to utilizing nutritionvalue.org. This provided enough data for each subcategory listed within the category. Not every single item and its contribution to the total consumption were able to be included due to data unavailability, untraceable local products, and underground markets.

While reviewing the data, higher rates of health outcomes and higher consumption rates were skewing some results. For this reason, Data Hub's Population Figures by Country data set was used in order to create per capita figures that aided the practicability of the results.

An advantage to the data gathering process is that the generalized approach gives a broad view of the relationship at hand without getting into product specifics. This is a useful approach for focusing nutrition information and health resources on a large scale.

C. Data Extraction and Preparation

The first step necessary was to create a macro in SAS that would facilitate an iterative process through each csv file in the Food Data folder. There is a separate csv for each food item with the nutritional data in a column and the nutritional values in a separate column. The macro would

first read through the folder of csv files and create a list of the filenames in a temporary file called “list.” This is preceded by creating a library to reference throughout for the folder “Capstone.” The macro method was selected in order to reduce redundancy in the code and increase efficiency, while an aftereffect is that the coding became increasingly obscure.

```
%global caps;
%let caps=/folders/myshortcuts/myfolder/Capstone;
libname capstone "&caps";


%macro list_files(dir, ext);
    %local filrf rc did memcnt name i;
    %let rc=%sysfunc(filename(filrf, &dir));
    %let did=%sysfunc(dopen(&filrf));

    %if &did eq 0 %then
        %do;
            %put Directory &dir cannot be open or does not exist;
            %return;
        %end;

    %do i=1 %to %sysfunc(dnum(&did));
        %let name=%qsysfunc(dread(&did, &i));

        %if %qupcase(%qscan(&name, -1, .))=%upcase(&ext) %then
            %do;
                %put &dir\&name;
                %let file_name =  %qscan(&name, 1, .);
                %put &file_name;

                data _tmp;
                    length dir $512 name $100;
                    dir=symget("dir");
                    name=symget("name");
                    path=catx('\', dir, name);
                    the_name=substr(name, 1, find(name, '.')-1);
                run;

                proc append base=list data=_tmp out=work.list force;
                run;

                quit;

                proc sql;
                    drop table _tmp;
                quit;

            %end;
        %else %if %qscan(&name, 2, .)=%then
            %do;
                %list_files(&dir\&name, &ext)
            %end;
    %end;
    %let rc=%sysfunc(dclose(&did));
    %let rc=%sysfunc(filename(filrf));

%mend list_files;
```

The following steps will then call on the “list” file variables to iterate through each file within the folder. This file is shown below.

Obs	dir	name	path	the_name
1	/folders/myshortcuts/myfolder/food_data	pie_peach.csv	/folders/myshortcuts/myfolder/food_data\pie_peach.csv	pie_peach
2	/folders/myshortcuts/myfolder/food_data	pie_prepared_from_recipe_apple.csv	/folders/myshortcuts/myfolder/food_data\pie_prepared_from_recipe_apple.csv	pie_prepared_from_recipe_apple
3	/folders/myshortcuts/myfolder/food_data	pie_prepared_from_recipe_pecan.csv	/folders/myshortcuts/myfolder/food_data\pie_prepared_from_recipe_pecan.csv	pie_prepared_from_recipe_pecan
4	/folders/myshortcuts/myfolder/food_data	plums_raw.csv	/folders/myshortcuts/myfolder/food_data\plums_raw.csv	plums_raw
5	/folders/myshortcuts/myfolder/food_data	pork_raw_ground_fresh.csv	/folders/myshortcuts/myfolder/food_data\pork_raw_ground_fresh.csv	pork_raw_ground_fresh
6	/folders/myshortcuts/myfolder/food_data	pork_unprepared_bacon_cured.csv	/folders/myshortcuts/myfolder/food_data\pork_unprepared_bacon_cured.csv	pork_unprepared_bacon_cured
7	/folders/myshortcuts/myfolder/food_data	potatoes_baked_flesh_and_skin_russet.csv	/folders/myshortcuts/myfolder/food_data\potatoes_baked_flesh_and_skin_russet.csv	potatoes_baked_flesh_and_skin_russet
8	/folders/myshortcuts/myfolder/food_data	potatoes_raw_flesh_and_skin_russet.csv	/folders/myshortcuts/myfolder/food_data\potatoes_raw_flesh_and_skin_russet.csv	potatoes_raw_flesh_and_skin_russet
9	/folders/myshortcuts/myfolder/food_data	potatoes_skin_raw.csv	/folders/myshortcuts/myfolder/food_data\potatoes_skin_raw.csv	potatoes_skin_raw
10	/folders/myshortcuts/myfolder/food_data	prune_puree.csv	/folders/myshortcuts/myfolder/food_data\prune_puree.csv	prune_puree
11	/folders/myshortcuts/myfolder/food_data	puddings_prepared_with_2_milk_dry_mix_rice.csv	/folders/myshortcuts/myfolder/food_data\puddings_prepared_with_2_milk_dry_mix_rice.csv	puddings_prepared_with_2_milk_dry_mix_rice
12	/folders/myshortcuts/myfolder/food_data	pumpkin_raw.csv	/folders/myshortcuts/myfolder/food_data\pumpkin_raw.csv	pumpkin_raw
13	/folders/myshortcuts/myfolder/food_data	radishes_dried_oriental.csv	/folders/myshortcuts/myfolder/food_data\radishes_dried_oriental.csv	radishes_dried_oriental
14	/folders/myshortcuts/myfolder/food_data	raspberries_seedless_puree.csv	/folders/myshortcuts/myfolder/food_data\raspberries_seedless_puree.csv	raspberries_seedless_puree
15	/folders/myshortcuts/myfolder/food_data	rhubarb_uncooked_frozen.csv	/folders/myshortcuts/myfolder/food_data\rhubarb_uncooked_frozen.csv	rhubarb_uncooked_frozen
16	/folders/myshortcuts/myfolder/food_data	rice_enriched_raw_mediumgrain_white.csv	/folders/myshortcuts/myfolder/food_data\rice_enriched_raw_mediumgrain_white.csv	rice_enriched_raw_mediumgrain_white
17	/folders/myshortcuts/myfolder/food_data	rice_raw_longgrain_brown.csv	/folders/myshortcuts/myfolder/food_data\rice_raw_longgrain_brown.csv	rice_raw_longgrain_brown
18	/folders/myshortcuts/myfolder/food_data	rye_grain.csv	/folders/myshortcuts/myfolder/food_data\rye_grain.csv	rye_grain
19	/folders/myshortcuts/myfolder/food_data	salt_table.csv	/folders/myshortcuts/myfolder/food_data\salt_table.csv	salt_table
20	/folders/myshortcuts/myfolder/food_data	seal_raw_alaska_native_meat_bearded_oogruk.csv	/folders/myshortcuts/myfolder/food_data\seal_raw_alaska_native_meat_bearded_oogruk.csv	seal_raw_alaska_native_meat_bearded_oogruk
21	/folders/myshortcuts/myfolder/food_data	seaweed_dried_agar.csv	/folders/myshortcuts/myfolder/food_data\seaweed_dried_agar.csv	seaweed_dried_agar
22	/folders/myshortcuts/myfolder/food_data	shallots_freezedried.csv	/folders/myshortcuts/myfolder/food_data\shallots_freezedried.csv	shallots_freezedried
23	/folders/myshortcuts/myfolder/food_data	shortening_industrial_lard_and_vegetable_oil.csv	/folders/myshortcuts/myfolder/food_data\shortening_industrial_lard_and_vegetable_oil.csv	shortening_industrial_lard_and_vegetable_oil
24	/folders/myshortcuts/myfolder/food_data	spices_bay_leaf.csv	/folders/myshortcuts/myfolder/food_data\spices_bay_leaf.csv	spices_bay_leaf
25	/folders/myshortcuts/myfolder/food_data	spices_celery_seed.csv	/folders/myshortcuts/myfolder/food_data\spices_celery_seed.csv	spices_celery_seed

As the macro iterates through each folder, it is creating a variable with the name of each food and is reading the data for the four observations which include information on Calories, Protein, Carbohydrates, and Fat. The next procedure transposes each file from narrow to wide, making the observations Calories, Protein, Carbohydrates, and Fat, into variables and transforming each file to have a single observation with multiple variables. Then, the PROC APPEND procedure is used to create a single dataset with all observations compiled.

```
%macro import_file(path, file_name, dataset_name );

        data &dataset_name REPLACE;
            INFILE "&path./&file_name." firstobs=5 obs=9 dsd truncover end=last;
            LENGTH Nutrient $20 Amount 8. Namevar $50;
            Namevar = "&file_name";
            Namevar = substr(Namevar,1,(length(Namevar)-4));
            INPUT Nutrient Amount;

            if not last then
                output;
        run;

        proc transpose data=&dataset_name out=&dataset_name (DROP=_NAME_);
            VAR Amount;
            ID Nutrient;
            BY Namevar;
        run;

        proc append base = H1
            data = &dataset_name FORCE;
        run;



%mend;

%list_files(/folders/myshortcuts/myfolder/food_data, csv);
```

The data step here serves to trigger the iteration of the macro through each file in the “list” file created earlier.

```
data _null_;
    set list;
    string=catt('%import_file(', dir, ', ', name, ', ', catt('H', _n_, ');'));
    call execute (string);
run;
```

Once run, this final piece of the macro creates the following dataset, which has a few duplicate files for food items. This is a flaw in the selection of this method, but it was then resolved by the NODUPKEY function in the PROC SORT procedure.

Obs	Namevar	Calories	Fat	Protein	Carbohydrate
1	pie_peach	224	10.00	1.90	32.90
2	pie_peach	224	10,00	1,90	32,90
3	pie_prepared_from_recipe_apple	265	12,50	2,40	37,10
4	pie_prepared_from_recipe_pecan	412	22,20	4,90	52,20
5	plums_raw	46	0.28	0.70	11.42
6	pork_raw_ground_fresh	263	21,19	16,88	0,00
7	pork_unprepared_bacon_cured	393	37,13	13,66	0,00
8	potatoes_baked_flesh_and_skin_russet	95	0,13	2,63	21,44
9	potatoes_raw_flesh_and_skin_russet	79	0.08	2.14	18.07
10	potatoes_skin_raw	58	0,10	2,57	12,44
11	prune_puree	257	0,20	2,10	65,10
12	puddings_prepared_with_2_milk_dry_mix_rice	111	1.63	3.29	20.81
13	pumpkin_raw	26	0,10	1,00	6,50
14	radishes_dried_oriental	271	0,72	7,90	63,37
15	raspberries_seedless_puree	41	0.87	1.02	7.99
16	rhubarb_uncooked_frozen	21	0.11	0.55	5.10
17	rice_enriched_raw_mediumgrain_white	360	0,58	6,61	79,34
18	rice_raw_longgrain_brown	367	3,20	7,54	76,25
19	rye_grain	338	1.63	10.34	75.86
20	salt_table	0	0.00	0.00	0.00
21	seal_raw_alaska_native_meat_bearded_oogruk	110	0,40	26,70	0,00
22	seaweed_dried_agar	306	0,30	6,21	80,88
23	shallots_freezedried	348	0.50	12.30	80.70
24	shortening_industrial_lard_and_vegetable_oil	900	100,00	0,00	0,00
25	spices_bay_leaf	313	8,36	7,61	74,97

```
PROC SORT DATA=H1
     OUT=capstone.combined_food_data
     NODUPKEY;
     BY Namevar;
RUN ;
```

The food dataset is cleaned. Now, A category variable must be added to each food item in order to determine the median calorie, fat, protein, and carbohydrate levels for each category to facilitate category analysis. To do so, the function INDEX() is used to search through the name of each food item for a key word and categorize the food item based on that. The categories are strategically ordered in the IF statements so that items are not misclassified. The key words for each category were determined through the Global Consumption Database, Description of Sectors through the World Bank website. The main disadvantage with this technique is that ordering the food items for category classification was time intensive and resulted in a long procedure. However, this method avoided the creation of web scraping the food types from The World Bank website.

```
data capstone.food_data_cats;
    set capstone.combined_food_data;
    if INDEX(Namevar, 'rice')>0 then cat=1;
    else if INDEX(Namevar, 'pasta')>0 then cat=5;
    else if INDEX(Namevar, 'crackers')>0 then cat=4;
    else if INDEX(Namevar, 'rusk')>0 then cat=4;
    else if INDEX(Namevar, 'toast')>0 then cat=4;
    else if INDEX(Namevar, 'biscuit')>0 then cat=4;
    else if INDEX(Namevar, 'wafer')>0 then cat=4;
    else if INDEX(Namevar, 'waffle')>0 then cat=4;
    else if INDEX(Namevar, 'muffin')>0 then cat=4;
    else if INDEX(Namevar, 'croissant')>0 then cat=4;
    else if INDEX(Namevar, 'cake')>0 then cat=4;
    else if INDEX(Namevar, 'tart')>0 then cat=4;
    else if INDEX(Namevar, 'pie')>0 then cat=4;
    else if INDEX(Namevar, 'bread')>0 then cat=3;
    else if INDEX(Namevar, 'english')>0 then cat=3;
    else if INDEX(Namevar, 'bagel')>0 then cat=3;
    else if INDEX(Namevar, 'butterscotch')>0 then cat=26;
    else if INDEX(Namevar, 'goat')>0 then cat=8;
    else if INDEX(Namevar, 'maiz')>0 then cat=2;
    else if INDEX(Namevar, 'wheat')>0 then cat=2;
    else if INDEX(Namevar, 'barley')>0 then cat=2;
    else if INDEX(Namevar, 'oat')>0 then cat=2;
    else if INDEX(Namevar, 'grain')>0 then cat=2;
    else if INDEX(Namevar, 'beef')>0 then cat=6;
    else if INDEX(Namevar, 'veal')>0 then cat=6;
    else if INDEX(Namevar, 'pork')>0 then cat=7;
    else if INDEX(Namevar, 'egg_custard')>0 then cat=16;
    else if INDEX(Namevar, 'egg_raw')>0 then cat=16;
    else if INDEX(Namevar, 'egg_poached')>0 then cat=16;
    else if INDEX(Namevar, 'egg_pasteurized')>0 then cat=16;
    else if INDEX(Namevar, 'egg_dried')>0 then cat=16;
    else if INDEX(Namevar, 'egg_fresh')>0 then cat=16;
    else if INDEX(Namevar, 'lamb')>0 then cat=8;
    else if INDEX(Namevar, 'mutton')>0 then cat=8;
    else if INDEX(Namevar, 'chicken')>0 then cat=9;
    else if INDEX(Namevar, 'duck')>0 then cat=9;
    else if INDEX(Namevar, 'goose')>0 then cat=9;
    else if INDEX(Namevar, 'turkey')>0 then cat=9;
    else if INDEX(Namevar, 'guinea')>0 then cat=9;
    else if INDEX(Namevar, 'game')>0  then cat=10;
    else if INDEX(Namevar, 'tea')>0 then cat=28;
    else if INDEX(Namevar, 'seal')>0 then cat=10;
    else if INDEX(Namevar, 'ham')>0 then cat=10;
    else if INDEX(Namevar, 'bacon')>0 then cat=10;
    else if INDEX(Namevar, 'spam')>0 then cat=10;
    else if INDEX(Namevar, 'fish_raw')>0 then cat=11;
    else if INDEX(Namevar, 'fish_dried')>0 then cat=12;
    else if INDEX(Namevar, 'fish_regular_lox')>0 then cat=12;
    else if INDEX(Namevar, 'fish_pickled')>0 then cat=12;
    else if INDEX(Namevar, 'fish_prepared')>0 then cat=12;
    else if INDEX(Namevar, 'fish_smoked')>0 then cat=12;
    else if INDEX(Namevar, 'curd')>0 then cat=15;
```

Above: Item 1

```
    else if INDEX(Namevar, 'cheese')>0 then cat=15;
    else if INDEX(Namevar, 'dessert')>0 then cat=14;
    else if INDEX(Namevar, 'chocolate')>0 then cat=26;
    else if INDEX(Namevar, 'ice_cream')>0 then cat=26;
    else if INDEX(Namevar, 'toffee')>0 then cat=26;
    else if INDEX(Namevar, 'gum')>0 then cat=26;
    else if INDEX(Namevar, 'yogurt')>0 then cat=14;
    else if INDEX(Namevar, 'cream')>0 then cat=14;
    else if INDEX(Namevar, 'evaporated') then cat=14;
    else if INDEX(Namevar, 'milk')>0 then cat=13;
    else if INDEX(Namevar, 'butter_salted')>0 then cat=17;
    else if INDEX(Namevar, 'butter_without_salt')>0 then cat=17;
    else if INDEX(Namevar, 'margarine')>0  then cat=17;
    else if INDEX(Namevar, 'oil')>0 then cat=18;
    else if INDEX(Namevar, 'lard')>0 then cat=18;
    else if INDEX(Namevar, 'melons_raw')>0 then cat=19;
    else if INDEX(Namevar, 'melon_raw')>0 then cat=19;
    else if INDEX(Namevar, 'strawberries_raw')>0 then cat=19;
    else if INDEX(Namevar, 'blueberries_raw')>0 then cat=19;
    else if INDEX(Namevar, 'blackberries_raw')>0 then cat=19;
    else if INDEX(Namevar, 'figs_raw')>0 then cat=19;
    else if INDEX(Namevar, 'plums_raw')>0 then cat=19;
    else if INDEX(Namevar, 'bananas_raw')>0 then cat=19;
    else if INDEX(Namevar, 'pears_raw')>0 then cat=19;
    else if INDEX(Namevar, 'limes_raw')>0 then cat=19;
    else if INDEX(Namevar, 'durian_raw')>0 then cat=19;
    else if INDEX(Namevar, 'loganberries_frozen')>0 then cat=20;
    else if INDEX(Namevar, 'melon_balls_frozen')>0 then cat=20;
    else if INDEX(Namevar, 'blueberries_frozen')>0 then cat=20;
    else if INDEX(Namevar, 'nance')>0 then cat=20;
    else if INDEX(Namevar, 'persimmons')>0 then cat=20;
    else if INDEX(Namevar, 'rhubarb_uncooked_frozen')>0 then cat=20;
    else if INDEX(Namevar, 'beets_raw')>0 then cat=21;
    else if INDEX(Namevar, 'onions_raw')>0 then cat=21;
    else if INDEX(Namevar, 'broccoli_raw')>0 then cat=21;
    else if INDEX(Namevar, 'kale_raw')>0 then cat=21;
    else if INDEX(Namevar, 'celery_raw')>0 then cat=21;
    else if INDEX(Namevar, 'spinach_raw')>0 then cat=21;
    else if INDEX(Namevar, 'pumpkin_raw')>0 then cat=21;
    else if INDEX(Namevar, 'cabbage_raw')>0 then cat=21;
    else if INDEX(Namevar, 'eggplant_raw')>0 then cat=21;
    else if INDEX(Namevar, 'asparagus_raw')>0 then cat=21;
    else if INDEX(Namevar, 'peas_raw')>0 then cat=21;
    else if INDEX(Namevar, 'bamboo_shoots_raw')>0 then cat=21;
    else if INDEX(Namevar, 'potato')>0 then cat=22;
    else if INDEX(Namevar, 'arrowroot')>0 then cat=22;
    else if INDEX(Namevar, 'cassava')>0 then cat=22;
    else if INDEX(Namevar, 'carrot_dehydrated')>0 then cat=23;
    else if INDEX(Namevar, 'tomatoes_sundried')>0 then cat=23;
    else if INDEX(Namevar, 'seaweed_dried')>0 then cat=23;
    else if INDEX(Namevar, 'peppers_dried')>0 then cat=23;
    else if INDEX(Namevar, 'hearts_of_palm_canned')>0 then cat=23;
    else if INDEX(Namevar, 'shallots_freezedried')>0 then cat=23;
    else if INDEX(Namevar, 'pickle')>0 then cat=23;
    else if INDEX(Namevar, 'kale_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'okra_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'radishes_dried')>0 then cat=23;
    else if INDEX(Namevar, 'edamame_prepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'edamame_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'carrots_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'turnips_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'peppers_canned')>0 then cat=23;
    else if INDEX(Namevar, 'asparagus_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'cauliflower_unprepared_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'onions_unprepared_whole_frozen')>0 then cat=23;
    else if INDEX(Namevar, 'jam')>0 then cat=25;
    else if INDEX(Namevar, 'sugar')>0 then cat=24;
    else if INDEX(Namevar, 'honey')>0 then cat=25;
    else if INDEX(Namevar, 'nectar')>0 then cat=25;
    else if INDEX(Namevar, 'marmalade')>0 then cat=25;
    else if INDEX(Namevar, 'puree')>0 then cat=25;
    else if INDEX(Namevar, 'salt')>0 then cat=27;
    else if INDEX(Namevar, 'spice')>0 then cat=27;
    else if INDEX(Namevar, 'parsley')>0 then cat=27;
    else if INDEX(Namevar, 'chives')>0 then cat=27;
    else if INDEX(Namevar, 'relish')>0 then cat=27;
    else if INDEX(Namevar, 'ginger')>0 then cat=27;
    else if INDEX(Namevar, 'garlic')>0 then cat=27;
    else if INDEX(Namevar, 'coffee')>0 then cat=28;
    else if INDEX(Namevar, 'decaffeinated')>0 then cat=28;
    else if INDEX(Namevar, 'cocoa')>0 then cat=28;
    else if INDEX(Namevar, 'soda')>0 then cat=29;
    else if INDEX(Namevar, 'soft_dri')>0 then cat=29;
    else if INDEX(Namevar, 'water')>0 then cat=29;
    else if INDEX(Namevar, 'beer')>0 then cat=32;
    else if INDEX(Namevar, 'wine')>0 then cat=31;
    else if INDEX(Namevar, 'red_table')>0 then cat=31;
    else if INDEX(Namevar, 'white_table')>0 then cat=31;
    else if INDEX(Namevar, 'cabernet')>0 then cat=31;
    else cat=30;
run;
```

Above: Item 2

Each item has been categorized and is then sorted and PROC MEANS is used to find the median of each category and place these results into an output file named “food_avgs.” The median was selected as a method to avoid outliers in the nutrition information, due to the fact that the food categories were diverse and the nutrition information varied greatly within a category. While this method avoided issues caused by the skewed distribution, the mean would consider the entirety of the data and perhaps be equally representative of the data.

```
proc sort data=capstone.food_data_cats out=food_data_cats2;
    by cat;
run;

proc means data=food_data_cats2 median;
    output out=capstone.food_avgs(DROP=_TYPE_ _FREQ_)
        median = /autoname;
    by cat;
run;
```

The MEANS Procedure

cat=1

Variable	Median
Calories	235.5000000
Fat	1.1050000
Protein	5.3000000
Carbohydrate	48.7950000

cat=2

Variable	Median
Calories	338.0000000
Fat	2.3000000
Protein	10.3400000
Carbohydrate	73.4800000

cat=3

Variable	Median
Calories	266.0000000
Fat	3.3000000
Protein	10.2000000
Carbohydrate	47.8000000

cat=4

Variable	Median
Calories	334.0000000
Fat	8.4000000
Protein	6.6000000
Carbohydrate	53.1000000

cat=5

Variable	Median
Calories	371.0000000
Fat	1.5100000
Protein	13.0400000
Carbohydrate	74.6700000

Now that the medians for each category have been identified, the file is given an additional variable named Sector and then given the proper name of the category instead of a number. This step was to facilitate a merge of this data onto the food consumption data in a later step.

```
data capstone.food_data_mergeable;
    set capstone.food_avgs;
    length sector $100;
    if cat=1 then sector='Rice';
    if cat=2 then sector='Other Cereals, Flour and Other Products';
    if cat=3 then sector='Bread';
    if cat=4 then sector='Other Bakery Products';
    if cat=5 then sector='Pasta Products';
    if cat=6 then sector='Beef and Veal';
    if cat=7 then sector='Pork';
    if cat=8 then sector='Lamb, Mutton and Goat';
    if cat=9 then sector='Poultry';
    if cat=10 then sector='Other Meats and Meat Preparations';
    if cat=11 then sector='Fresh, Chilled or Frozen Fish and Seafood';
    if cat=12 then sector='Preserved or Processed Fish and Seafood';
    if cat=13 then sector='Fresh Milk';
    if cat=14 then sector='Preserved Milk and Other Milk Products';
    if cat=15 then sector='Cheese';
    if cat=16 then sector='Eggs and Egg-Based Products';
    if cat=17 then sector='Butter and Margarine';
    if cat=18 then sector='Other Edible Oil and Fats';
    if cat=19 then sector='Fresh or Chilled Fruit';
    if cat=20 then sector='Frozen, Preserved or Processed Fruit and Fruit-bas';
    if cat=21 then sector='Fresh or Chilled Vegetables Other than Potatoes';
    if cat=22 then sector='Fresh or Chilled Potatoes';
    if cat=23 then sector='Frozen, Preserved or Processed Vegetables and Vegetable-Based Products';
    if cat=24 then sector='Sugar';
    if cat=25 then sector='Jams, Marmalades and Honey';
    if cat=26 then sector='Confectionery, Chocolate and Ice Cream';
    if cat=27 then sector='Food Products n.e.c.';
    if cat=28 then sector='Coffee, Tea and Cocoa';
    if cat=29 then sector='Mineral Waters, Soft Drinks, Fruit and Vegetable Juices';
    if cat=30 then sector='Spirits';
    if cat=31 then sector='Wine';
    if cat=32 then sector='Beer';
run;
```

Obs	cat	Calories_Median	Fat_Median	Protein_Median	Carbohydrate_Median	sector
1	1	236,5	1,105	5,300	48,795	Rice
2	2	338,0	2,300	10,340	73,480	Other Cereals, Flour and Other Products
3	3	266,0	3,300	10,200	47,800	Bread
4	4	334,0	8,400	6,600	53,100	Other Bakery Products
5	5	371,0	1,510	13,040	74,670	Pasta Products
6	6	388,0	36,675	13,440	0,000	Beef and Veal
7	7	328,0	29,160	15,270	0,000	Pork
8	8	258,0	15,370	22,675	0,000	Lamb, Mutton and Goat
9	9	146,0	7,680	19,200	0,000	Poultry
10	10	117,0	2,370	22,015	0,000	Other Meats and Meat Preparations

Moving on to the preparation of the Food Consumption Rate Dataset. The Food Consumption rate csv files were available on The World Bank website with each file separated by category. For this reason, the same methodology was used to process the data. First a macro was created, each file name was read into the file "list," and then an iterative process transferred the csv files into SAS files, keeping only the US currency variable (out of the various choices for currency) and the National variable (out of the Rural, Suburban, and Urban additional designators). The

data was then sorted by country and transposed, creating a wide observation for each country. The macro finishes out by merging all of the datasets into one.

```
%macro list_files(dir, ext);
    %local filrf rc did memcnt name i;
    %let rc=%sysfunc(filename(filrf, &dir));
    %let did=%sysfunc(dopen(&filrf));

    %if &did eq 0 %then
        %do;
            %put Directory &dir cannot be open or does not exist;
            %return;
        %end;

    %do i=1 %to %sysfunc(dnum(&did));
        %let name=%qsysfunc(dread(&did, &i));

        %if %qupcase(%qscan(&name, -1, .))=%upcase(&ext) %then
            %do;
                %put &dir\&name;
                %let file_name =  %qscan(&name, 1, .);
                %put &file_name;

                data _tmp;
                    length dir $512 name $100;
                    dir=symget("dir");
                    name=symget("name");
                    path=catx('\', dir, name);
                    the_name=substr(name, 1, find(name, '.')-1);
                run;

                proc append base=list data=_tmp out=work.list force;
                run;

                quit;

                proc sql;
                    drop table _tmp;
                quit;

            %end;
        %else %if %qscan(&name, 2, .)=%then
            %do;
                %list_files(&dir\&name, &ext)
            %end;
    %end;
    %let rc=%sysfunc(dclose(&did));
    %let rc=%sysfunc(filename(filrf));

%mend list_files;
```

```
%macro import_file(path, file_name, dataset_name );

        data &dataset_name REPLACE;
            INFILE "&path./&file_name." firstobs=2 dsd truncover end=last;
            LENGTH Area Consumption_Segment Country Measure_Names $15
                Sector $50;
            INPUT Area Consumption_Segment Country Measure_Names
                Sector Measure_Values;

            if Measure_Names = "US$" and Area = "National" then
                output;
        run;

        PROC SORT DATA=&dataset_name OUT=&dataset_name;
            BY Country;
        RUN;

        proc transpose data= &dataset_name out= &dataset_name (DROP=_Name_);
            VAR Measure_Values;
            ID Consumption_Segment;
            BY Country Sector;
        run;

        proc append base = C1
            data = &dataset_name FORCE;
        run;

%mend;

%list_files(/folders/myshortcuts/myfolder/consumption_data, csv);

/*This code will iterate through the files and import each of them with a new title*/
```

```
    data _null_;
        set list;
        string=catt('%import_file(', dir, ', ', name, ', ', catt('C', _n_, ');'));
        call execute (string);
    run;
```

Obs	dir	name	path	the_name
1	/folders/myshortcuts/myfolder/consumption_data	ST_02_data_bakery.csv	/folders/myshortcuts/myfolder/consumption_data/ST_02_data_bakery.csv	ST_02_data_bakery
2	/folders/myshortcuts/myfolder/consumption_data	ST_02_data_beef_veal.csv	/folders/myshortcuts/myfolder/consumption_data/ST_02_data_beef_veal.csv	ST_02_data_beef_veal
3	/folders/myshortcuts/myfolder/consumption_data	ST_02_data_Beer.csv	/folders/myshortcuts/myfolder/consumption_data/ST_02_data_Beer.csv	ST_02_data_Beer
4	/folders/myshortcuts/myfolder/consumption_data	ST_02_data_bread.csv	/folders/myshortcuts/myfolder/consumption_data/ST_02_data_bread.csv	ST_02_data_bread
5	/folders/myshortcuts/myfolder/consumption_data	ST_02_data_butter.csv	/folders/myshortcuts/myfolder/consumption_data/ST_02_data_butter.csv	ST_02_data_butter

Above: List Dataset created to iterate through the Food Consumption categories

The next code turns the temporary file into a permanent one saved in the Capstone folder for use later on in the analysis, then sorts the file to remove duplicates.

```
data capstone.combined_consump_data;
    set work.c1;
run;

PROC SORT DATA=capstone.combined_consump_data
     OUT=capstone.combined_consump_data nodupkey;
     BY Country Sector;
RUN;
```

Obs	Country	Sector	Higher	Middle	Low	Loweet	All
1	Afghanistan	Fresh or Chilled Potatoes	0.04	6.74	124.11	104.43	235.32
2	Afghanistan	Poultry	0.09	7.37	104.31	63.11	174.88
3	Afghanistan	Pork	0.00	0.00	0.00	0.00	0.00
4	Afghanistan	Lamb, Mutton and Goat	0.36	34.73	470.61	144.16	649.87
5	Afghanistan	Spirits	0.00	0.00	0.00	0.00	0.00
6	Afghanistan	Fresh or Chilled Vegetables Other than Potatoes	0.59	41.80	439.25	212.58	694.23
7	Afghanistan	Coffee, Tea and Cocoa	0.07	7.23	142.05	128.54	277.88
8	Afghanistan	Frozen, Preserved or Processed Fruit and Fruit-bas	1.08	33.29	176.58	39.69	250.64
9	Afghanistan	Rice	0.14	38.90	590.86	420.46	1050.36
10	Afghanistan	Cheese	0.04	3.50	13.47	7.03	24.03

The final dataset, health outcomes, also needs to be transferred from a csv file to a SAS file. The input variables are reduced to four variables relevant to the research. Then, the file is reduced to only the observations for 2010 health outcomes. This is to align the data with the World Bank data, 2010 Food Consumption Rates. The dataset is sorted by country and then transposed to create a wide dataset with variables for each health outcome.

```
data capstone.health_outcomes REPLACE;
    INFILE "/folders/myshortcuts/myfolder/health_outcome/Health Outcomes Dataset.csv"
        firstobs=3 dsd truncover;
    LENGTH Country $40 Year 5. Causes $30 Deaths_both 8.;
    INPUT Country Year Causes Deaths_both;
run;

data capstone.health_outcomes(Drop=Year) replace;
    set capstone.health_outcomes;
    where Year=2010;
    if Country="Viet Nam" then Country="Vietnam";
    if Country="Kyrgyzstan" then Country="Kyrgyz Republic";
    if Country="Lao People's Democratic Republic" then Country="Lao PDR";
    if Country="Côte d'Ivoire" then Country="Cote d'Ivoire";
    if Country="Congo" then Country="Congo, Rep.";
    if Country="Democratic Republic of the Congo" then Country="Congo, Dem. Rep.";
    if Country="United Republic of Tanzania" then Country="Tanzania";
    if Country="Bolivia (Plurinational State of)" then Country="Bolivia";
    if Country="Egypt" then Country="Egypt, Arab Rep.";
    if Country="Gambia" then Country="Gambia, The";
    if Country="Republic of North Macedonia" then Country="Macedonia, FYR";
    if Country="Republic of Moldova" then Country="Moldova";
    if Country="Eswatini" then Country="Swaziland";
    if Country="Timor-Leste" then Country="Timor Leste";
    if Country="Yemen" then Country="Yemen, Rep.";
run;

proc sort data=capstone.health_outcomes out=capstone.health_outcomes;
    by Country;
run;

proc transpose data=capstone.health_outcomes out=capstone.health_outcomes (DROP=_NAME_);
    ID Causes;
    BY Country;
run;
```

Obs	Country	Malignant_neoplasms	Diabetes_mellitus	Cardiovascular_diseases	Chronic_obstructive_pulmonary
1	Afghanistan	16199	4852	43842	5706
2	Albania	4700	141	12796	737
3	Algeria	20548	6754	58367	5149
4	Angola	7041	1907	19915	3429
5	Antigua and Barbuda	102	52	184	21
6	Argentina	63885	8884	98551	38637
7	Armenia	7769	1229	12335	1144
8	Australia	42508	3963	44796	10218
9	Austria	20308	3091	32951	3870
10	Azerbaijan	8904	1347	30321	1877

Above, several of the naming conventions for the Health Outcomes data set had to be adjusted to match the Food Consumption data set in order to prepare the set for a successful merge. This was executed with several if statements to rename countries. While this was a tedious and lengthy approach, it avoided the removal of the entire columns removal of additional characters and spaces, which would've resulted in lengthier procedures and muddled observations in that column.

The datasets were then ready to begin merging. The DATA step was selected for this procedure due to the fact that it was a simple merge. Contrastingly, SQL could have been used which would have reduced some code, but would have reduced processing time. The Food Consumption dataset and Food dataset are then sorted by Sector and merged. After merging, the new dataset is sorted by country alongside the Health Outcomes dataset in order to prepare for a second merge.

```
proc sort data=capstone.combined_consump_data out=combined_consump_data2;
    by sector;
run;

proc sort data=capstone.food_data_mergeable out=food_data_mergeable2;
    by sector;
run;

data capstone.consumpandfood_merged;
    merge combined_consump_data2(in=a)
          food_data_mergeable2(in=b);
    by sector;
    if a;
run;

proc sort data=capstone.health_outcomes out=capstone.health_outcomes;
    by Country;
run;

proc sort data=capstone.consumpandfood_merged out=capstone.consumpandfood_merged2;
    by Country;
run;

data capstone.consumpfoodhealth;
    merge capstone.consumpandfood_merged2(in=a)
          capstone.health_outcomes(in=b);
    by Country;
    if a;
run;
```

Obs	Country	Sector	Higher	Middle	Low	Lowest	All	cat	Calories_Median	Fat_Median	Protein_Median	Carbohydrate_Median	Malignant_neoplasms	Diabetes_mellitus	Cardiovascular_diseases	Chronic_obstructive_pulmonary
1	Afghanistan	Beef and Veal	0.27	42.02	376.93	151.53	570.75	6	388.0	36.875	13.440	0.000	16199	4852	43842	5706
2	Afghanistan	Beer	0.00	0.00	0.00	0.00	0.00	32	41.0	0.000	0.240	0.870	16199	4852	43842	5706
3	Afghanistan	Bread	0.33	43.71	170.58	33.85	248.47	3	266.0	3.300	10.200	47.800	16199	4852	43842	5706
4	Afghanistan	Butter and Margarine	0.00	2.96	92.78	49.06	144.81	17	559.0	62.785	0.560	0.060	16199	4852	43842	5706
5	Afghanistan	Cheese	0.04	3.50	13.47	7.03	24.03	15	350.0	27.440	23.410	2.570	16199	4852	43842	5706
6	Afghanistan	Coffee, Tea and Cocoa	0.07	7.23	142.05	128.54	277.88	28	2.0	0.000	0.110	0.185	16199	4852	43842	5706
7	Afghanistan	Confectionery, Chocolate and Ice Cream	0.08	9.61	130.98	81.21	221.88	26	489.5	27.815	3.700	56.410	16199	4852	43842	5706
8	Afghanistan	Eggs and Egg-Based Products	0.11	9.71	110.75	49.17	169.74	16	159.0	10.915	12.685	1.140	16199	4852	43842	5706
9	Afghanistan	Food Products n.e.c.	0.06	9.07	102.91	66.48	178.52	27	282.0	5.200	11.430	47.750	16199	4852	43842	5706
10	Afghanistan	Fresh Milk	0.06	8.79	156.21	158.03	323.09	13	51.5	1.485	3.385	4.920	16199	4852	43842	5706

In the merge statements, the IN= paired with the IF clause was used to manufacture a "left join," where only the observations that matched with the Food Consumption dataset were kept. This practice reduced missing values in the end product.

Then the population dataset needed to be read into SAS, sorted and merged with this dataset. This dataset was already in a csv file and was imported into SAS with a data step similar to the one used with the health outcomes data set.

```
data capstone.pop_by_country REPLACE;
    INFILE "/folders/myshortcuts/myfolder/Capstone/pop_by_country.csv"
        firstobs=2 dsd truncover;
        LENGTH Country $20;
    INPUT Country Country_Code Year_1960-Year_2016;
run;

data capstone.pop_by_country(rename=(Year_2010=Population)) replace;
    set capstone.pop_by_country;
    Keep Country Year_2010;
    if Country="Timor-Leste" then Country="Timor Leste";
    if Country="St. Lucia" then Country="Saint Lucia";
run;

proc sort data=capstone.pop_by_country out=capstone.pop_by_country;
    by Country;
run;

proc sort data=capstone.consumpfoodhealth out=capstone.consumpfoodhealth;
    by Country;
run;

data capstone.consumpfoodhealth;
    merge capstone.consumpfoodhealth(in=a)
          capstone.pop_by_country(in=b);
    by Country;
    if a;
run;

data capstone.consumpfoodhealth replace;
    set capstone.consumpfoodhealth;
    percapita_all=(all*1000000)/population;
    percapita_neoplasms=malignant_neoplasms/population;
    percapita_diabetes=Diabetes_mellitus/population;
    percapita_cardio=Cardiovascular_diseases/population;
    percapita_pulmonary=Chronic_obstructive_pulmonary/population;
run;
```

In the above procedures, the data set was paired down to essential variables and several of the Country observations were changed to prepare for the subsequent merge. After the merge, several variables were created. The “percapita_all” variable showed the per capita consumption rates for each nation, adjusted for the “all” variable, which was in millions. Each health outcome was then adjusted to a per capita rate. This would facilitate more accurate results that were not inflated by countries with larger populations.

The final data set is clean, prepared, merged, and ready for exploratory data analysis.

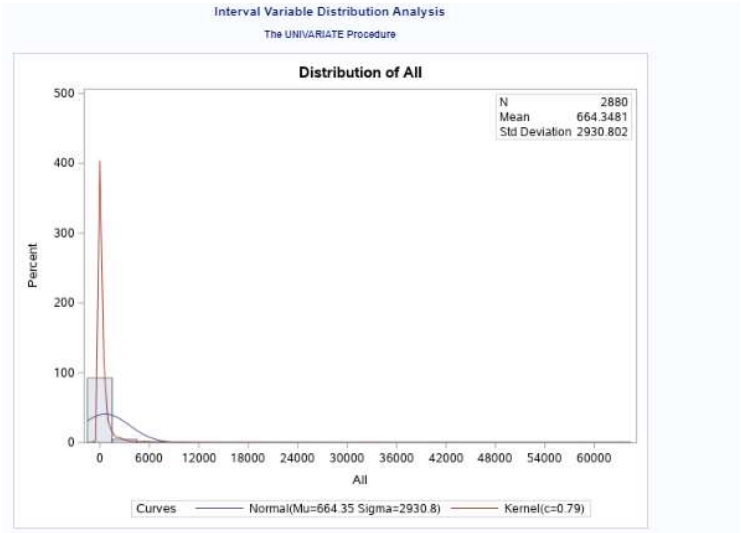
D. Exploratory Data Analysis

I. Distribution Analysis

Upon reviewing the data, there are no categorical variables, only continuous variables. To analyze the distribution of these variables, the research uses the UNIVARIATE procedure with the histogram function. This procedure was selected in order to provide both graphs and statistics on the distribution of the numeric data. While the MEANS procedure might offer several of the same statistics, the UNIVARIATE procedure has more features for analyzing distribution and additionally produces graphs. Analyzing the distribution was key in determining whether this data would provide accurate models, or whether the distribution needed to be modified in order to improve the normalcy. From this analysis, it is shown that none of the variables have a strictly normal distribution. The only variables with a semi-normal distribution are protein median, fat median, carbohydrate median, and calories median. Though removing outliers would potentially

resolve the issue, the data points were both verified and legitimate and the truth could be misrepresented without these outliers.

```
proc univariate data=capstone.consumpfoodhealth noprint;
    var &interval;
    histogram &interval / normal kernel;
    inset n mean std /position=ne;
run;
```



Interval Variable Distribution Analysis

The UNIVARIATE Procedure
Fitted Normal Distribution for All

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	664.3481
Std Dev	Sigma	2930.802

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.410337	Pr > D	<0.010
Cramer-von Mises	W-Sq	160.109984	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	764.821319	Pr > A-Sq	<0.005

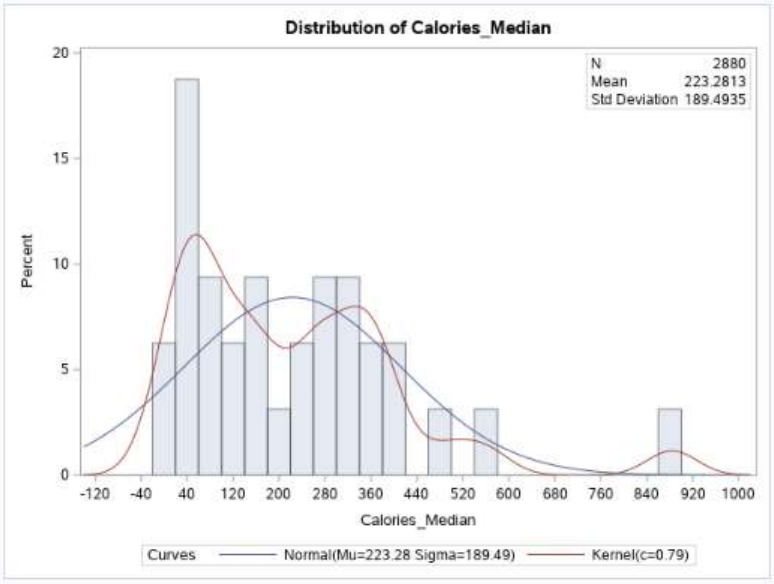
Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.0	-6153.716
5.0	0.0	-4156.391
10.0	1.3	-3091.625
25.0	10.9	-1312.447
50.0	83.3	664.348
75.0	299.3	2641.144
90.0	1151.5	4420.321
95.0	2472.1	5485.088
99.0	11595.0	7482.412

The above distribution analysis is representative of the analysis results for consumption rates amongst the lowest-income, low-income, middle-income, high-income, and all-income areas of developing countries. The standard deviation for each consumption rate indicated there is a large spread of the data which is shown in the graph. The Kolmogorov-Smirnov test for all five analyses is above .05, which indicated that the difference between the data and the normal curve was significant. The more powerful Anderson-Darling and Cramer-von Mises test show similar results to the KS test. This is also seen in the differences between observed and estimated quantiles for the distribution. Due to the tails of the data carrying much of the abnormalities, the Anderson-Darling Test is the most reliable for these variables. The Kolmogorov-Smirnov and

Cramer von Mises tests are preferable when the center of the data contains the majority of the deviation (Yap and Sim, 2011).

Interval Variable Distribution Analysis

The UNIVARIATE Procedure



Interval Variable Distribution Analysis

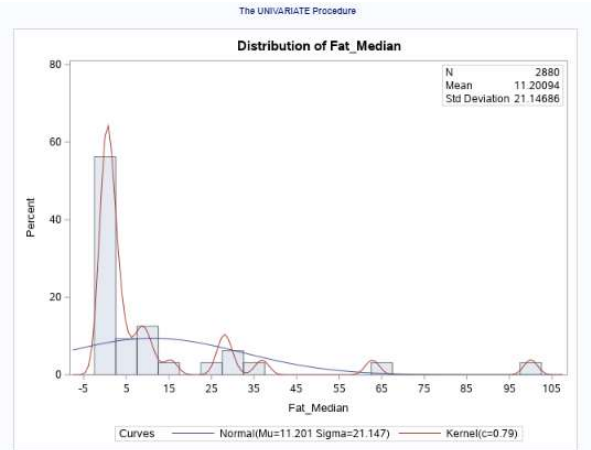
The UNIVARIATE Procedure
Fitted Normal Distribution for Calories_Median

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	223.2813
Std Dev	Sigma	189.4935

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.1193374	Pr > D	<0.010
Cramer-von Mises	W-Sq	10.0647054	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	76.2894993	Pr > A-Sq	<0.005

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.000	-217.5465
5.0	2.000	-88.4078
10.0	36.000	-19.5644
25.0	65.000	95.4698
50.0	178.250	223.2813
75.0	336.000	351.0927
90.0	388.000	466.1269
95.0	559.000	534.9703
99.0	884.000	664.1090

Above is the distribution analysis for median calorie rates for each food group. The standard deviation indicated that the spread of the data was relatively small, which is also shown in the graph. The Kolmogorov-Smirnov test is above .05, which, though the distribution was much closer to normal than the consumption variables, indicated that the difference between the data and the normal curve was significant. The more powerful Anderson-Darling and Cramer-von Mises test show similar results to the KS test. In this case, the more centered skew in the data makes the KS and Cramer-von Mises statistics more reliable. This is also seen in the differences between observed and estimated quantiles for the distribution.



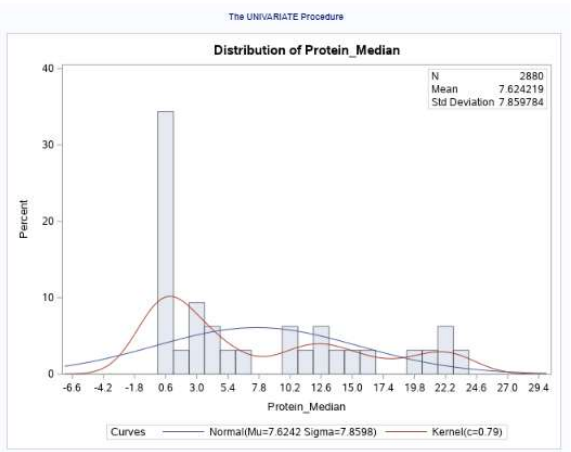
The UNIVARIATE Procedure
Fitted Normal Distribution for Fat_Median

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	11.20094
Std Dev	Sigma	21.14686

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.298169	Pr > D	<0.010
Cramer-von Mises	W-Sq	84.412248	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	437.836520	Pr > A-Sq	<0.005

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.0000	-37.99402
5.0	0.0000	-23.58256
10.0	0.0000	-15.89986
25.0	0.1525	-3.06241
50.0	1.9050	11.20094
75.0	10.1325	25.46428
90.0	29.1600	38.30173
95.0	62.7850	45.98443
99.0	100.0000	60.39590

Image one: Fat_Median Distribution



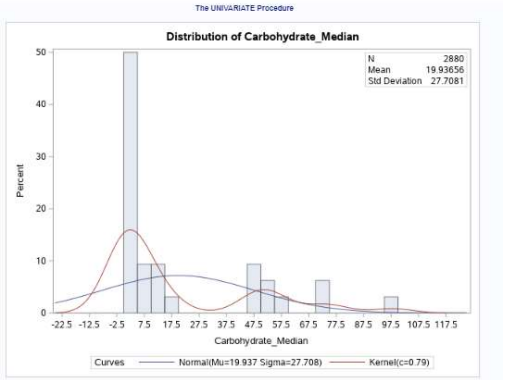
The UNIVARIATE Procedure
Fitted Normal Distribution for Protein_Median

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	7.624219
Std Dev	Sigma	7.859784

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.195081	Pr > D	<0.010
Cramer-von Mises	W-Sq	28.914863	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	163.172684	Pr > A-Sq	<0.005

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.0000	-10.66037
5.0	0.0000	-5.30398
10.0	0.0000	-2.44850
25.0	0.4300	2.32287
50.0	4.0000	7.62422
75.0	13.2400	12.92556
90.0	20.8400	17.69694
95.0	22.6750	20.55241
99.0	23.4100	25.90881

Image two: Protein_Median Distribution



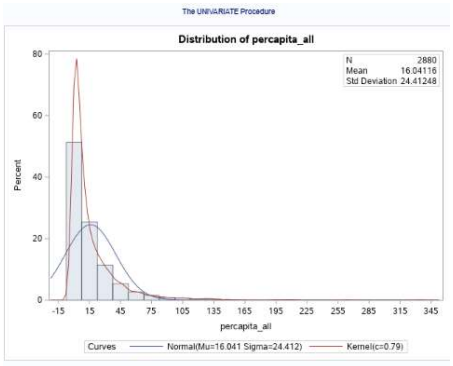
The UNIVARIATE Procedure
Fitted Normal Distribution for Carbohydrate_Median

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	19.93656
Std Dev	Sigma	27.7081

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.280258	Pr > D	<0.010
Cramer-von Mises	W-Sq	65.583883	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	338.884585	Pr > A-Sq	<0.005

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.0000	-44.52213
5.0	0.0000	-25.63921
10.0	0.0000	-15.57280
25.0	0.0000	1.24773
50.0	5.2850	19.93656
75.0	47.7750	38.62539
90.0	56.4100	55.44593
95.0	74.6700	65.51234
99.0	98.0900	84.39525

Image three: Carbohydrate Median Distribution



The UNIVARIATE Procedure
Fitted Normal Distribution for percapita_all

Parameters for Normal Distribution		
Parameter	Symbol	Estimate
Mean	Mu	16.04116
Std Dev	Sigma	24.41248

Goodness-of-Fit Tests for Normal Distribution				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.255562	Pr > D	<0.010
Cramer-von Mises	W-Sq	51.242552	Pr > W-Sq	<0.005
Anderson-Darling	A-Sq	274.981936	Pr > A-Sq	<0.005

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	0.000	-40.75077
5.0	0.001	-24.11380
10.0	0.159	-15.24469
25.0	1.543	-0.42481
50.0	7.135	16.04116
75.0	20.897	32.50713
90.0	42.568	47.32702
95.0	61.328	56.19613
99.0	115.386	72.83310

Image four: Percapita_All Distribution

Above is the distribution analysis for median fat, protein, and carbohydrate levels for each food group, and also the per capita adjustment for the consumption rate. Interestingly, this per capita adjustment has drastically improved the normalcy of the distribution. The standard deviation indicated that the spread of the data was relatively small, which is also shown in each of the graphs. The Kolmogorov-Smirnov test is between .19 and .3 for each of the graphs, which again indicated that the difference between the data and the normal curve was significant. The more powerful Anderson-Darling and Cramer-von Mises test show similar results to the KS test. The above data shows irregularities around the center of the distribution, indicating that the KS and Cramer-von Mises tests are better representations of the distribution. These irregularities were also seen in the differences between observed and estimated quantiles for the distribution. Overall, each of the nutrient variables demonstrate a much closer-to-normal distribution than the consumption variables. Adjusting the nutrient variables with a logarithmic transformation proved unnecessary.

II. Correlation Analysis

PROC CORR was used to produce correlation statistics and scatter plots in order to determine which variables are linearly associated with diabetes mellitus, chronic obstructive pulmonary disease, cardiovascular disease, and malignant neoplasms respectively. This analysis was designed to analyze relationships for further analysis and model building. The CORR procedure was utilized to calculate the Pearson Correlation Coefficients for the continuous variables. This method was selected to measure the strength of any linear relationship between the variables. Linear associations are a critical component of building a predictive model with the LOGISTIC procedure, which is why the Pearson Correlation Coefficient was selected as the methodology of choice. While many assumptions of this method were met, including that both variables were continuous, there was independence of cases, and the cases were paired, but one violation was the normality of the data (Laerd, 2020). Pearson's Correlation was selected over Spearman's correlation because the variables weren't ordinal or ranked and was thus better represented by Pearson's.

Pearson Correlation Coefficients, N = 2880
Prob > |r| under H0: Rho=0

	percapita_all	Middle	All	Low	Lowest	Higher	Carbohydrate_Median	Protein_Median	Fat_Median	Calories_Median
percapita_diabetes	0.06067 0.0011	-0.05077 0.0064	-0.04872 0.0089	-0.03829 0.0399	-0.03768 0.0432	-0.03375 0.0701	0.00000 1.0000	0.00000 1.0000	0.00000 1.0000	0.00000 1.0000

```
ods graphics / reset=all imagemap;

proc corr data=capstone.consumpfoodhealth rank
    plots(only)=scatter(nvar=all ellipse=none);
    var &interval;
    with percapita_diabetes;
    id cat;
    title "Correlation and Scatter Plots with Diabetes";
run;

ods graphics off;
```

Pearson Correlation Coefficients, N = 2880 Prob > |r| under H0: Rho=0										
Diabetes_mellitus	Lowest 0.53314 <.0001	All 0.49691 <.0001	Low 0.45763 <.0001	Middle 0.33254 <.0001	Higher 0.22453 <.0001	percapita_all -0.06963 0.0002	Carbohydrate_Median 0.00000 1.0000	Protein_Median 0.00000 1.0000	Fat_Median 0.00000 1.0000	Calories_Median 0.00000 1.0000

Above are the results of the correlation analysis for diabetes mellitus. These results show that, especially once there is an adjustment for population, the consumption rate is positively correlated to the rate of diabetes. When compared to the consumption rates that have been adjusted for income level, but not for population levels, there is a negative relationship. More than anything, this depicts the need for population adjustment levels in these areas. When diabetes is analyzed without a per capita adjustment, higher consumption is correlated with lower rates of diabetes mellitus. This is true of all of the consumption rates, but less so amongst the middle and higher-income classes. These results are representative of the correlation analysis for chronic obstructive pulmonary disease, cardiovascular disease, and malignant neoplasms. This analysis aided the model that would be built based on this research. The correlation analysis was concluded with a correlation table to investigate relationships amongst the predictors.

```
proc corr data=capstone.consumpfoodhealth nosimple best=3;
    var &interval;
    title "Correlations of Predictors";
run;
```

Correlations and Scatter Plot Matrix of Predictors

The CORR Procedure

9 Variables:	Higher Middle Low Lowest All Calories_Median Fat_Median Protein_Median Carbohydrate_Median

Pearson Correlation Coefficients, N = 2880 Prob > |r| under H0: Rho=0			
Higher	Higher 1.00000	Middle 0.83681 <.0001	All 0.68756 <.0001
Middle	Middle 1.00000	Low 0.90604 <.0001	All 0.90120 <.0001
Low	Low 1.00000	All 0.96569 <.0001	Middle 0.90604 <.0001
Lowest	Lowest 1.00000	All 0.72389 <.0001	Low 0.60927 <.0001
All	All 1.00000	Low 0.96569 <.0001	Middle 0.90120 <.0001
Calories_Median	Calories_Median 1.00000	Fat_Median 0.83927 <.0001	Carbohydrate_Median 0.28469 <.0001
Fat_Median	Fat_Median 1.00000	Calories_Median 0.83927 <.0001	Carbohydrate_Median -0.22629 <.0001
Protein_Median	Protein_Median 1.00000	Carbohydrate_Median -0.14827 <.0001	Calories_Median 0.10699 <.0001
Carbohydrate_Median	Carbohydrate_Median 1.00000	Calories_Median 0.28469 <.0001	Fat_Median -0.22629 <.0001

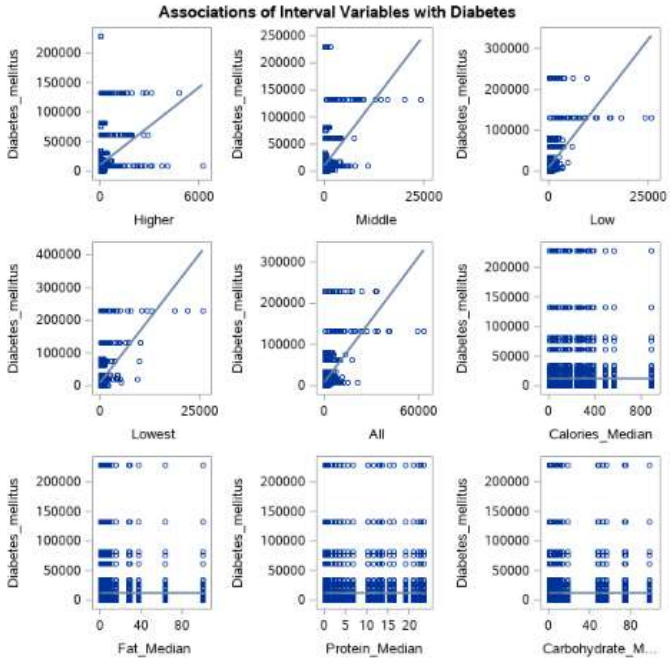
None of the variables had an unusual or surprising relationship with one another. Consumption rates had a significant correlation amongst themselves, as would be expected. Additionally, fat and calories were highly correlated, protein and carbohydrates had a negative relationship, and carbohydrates and fat had a negative relationship. These relationships provided a sound basis upon which to develop a model.

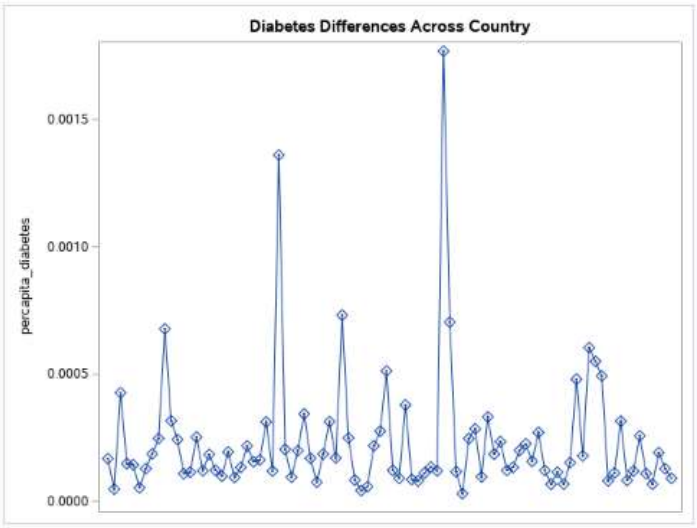
III. Plotting the Data to Explore Associations

After analyzing the correlations, it was necessary to investigate associations between the health outcome variables and the predictor variables. PROC SGSCATTER was used to develop scatter plots that would demonstrate the shape of the associations. The CORR procedure could have been selected to produce scatter plot matrices, but the matrix would not have trained the variables against diabetes specifically. For this reason, SGSCATTER was chosen. Then PROC SGPLOT was used to produce vertical bar charts. Diabetes mellitus was examined for its distribution by country. SGPLOT was the prime candidate for this graphical display because of its unique customization features and versatility.

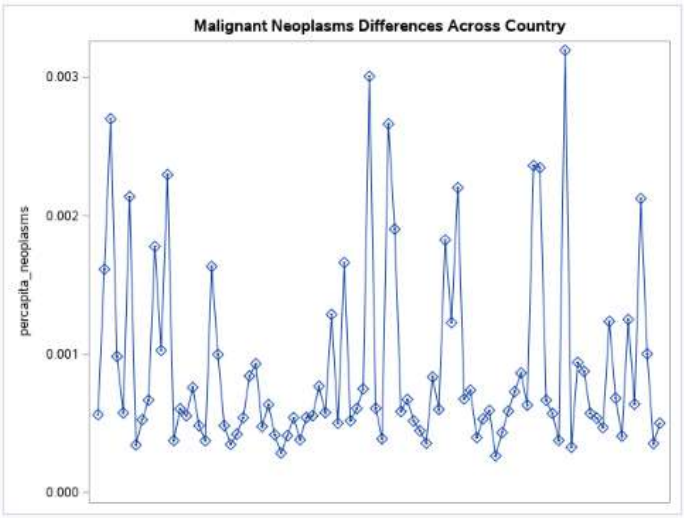
```
ods html path='/folders/myfolders' GPATH='/folders/myfolders/';
ods graphics on / imagemap;

proc sgscatter data=capstone.consumpfoodhealth;
    plot Diabetes_mellitus*(&interval) / reg;
    title "Associations of Interval Variables with Diabetes";
run;

proc sgplot data=capstone.consumpfoodhealth;
    vbox Diabetes_mellitus / category=Country connect=mean;
    xaxis display=none;
    title "Diabetes Differences Across Country";
run;
```

Above, the ODS options deliver x and y coordinate information in a data box as the user scrolled over a grid point. For this reason, the x axis is not displayed in the SGPLOT procedure.

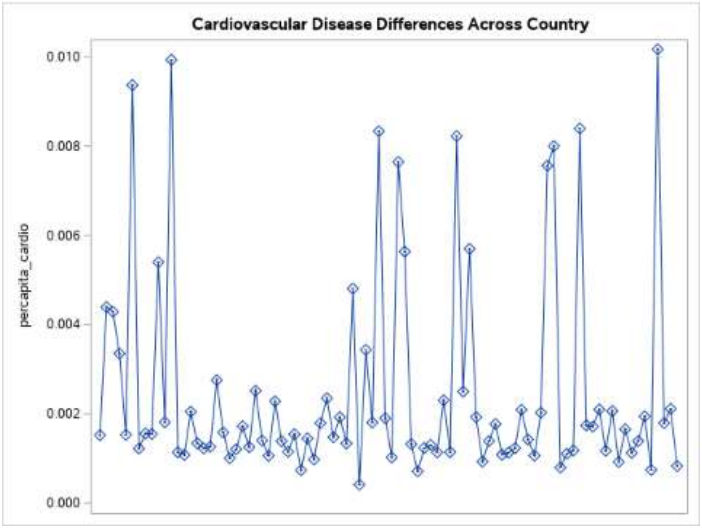




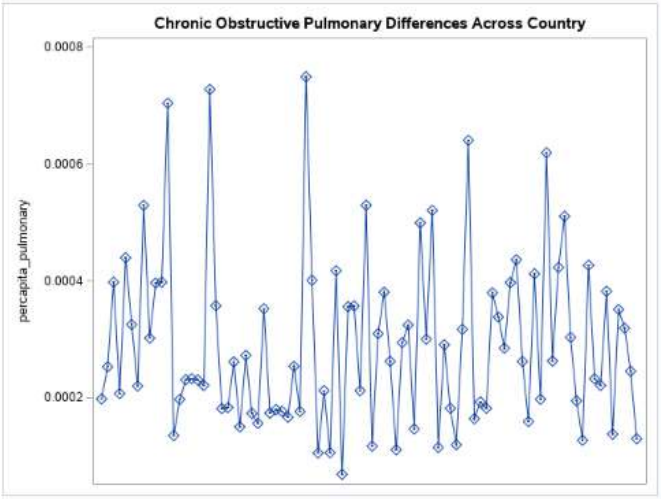
The associations in the scatter plots demonstrate the same results found in the correlation analysis, confirming the associations between consumption rates and diabetes, as well as the decreased strength in the associations amongst high-income and middle-income areas. Interestingly, the diabetes rates vs. country plot demonstrates that there is a cause-effect relationship worth investigating. There are countries whose per capita diabetes rate far exceeds that of other nations. In the above graph, Mauritius has the highest rate, followed in descending order by Fiji, Jamaica, Mexico, and Bosnia and Herzegovina. The model will attempt to explain possible predictors for these higher rates.



Shown above: per capita Malignant Neoplasm rates are led, in order, by Serbia, Latvia, Armenia. Lithuania, and Bulgaria



Shown above: per capita cardiovascular disease rates are led, in order, by Ukraine, Bulgaria, Belarus, Serbia, Latvia, Moldova



Shown above: per capita chronic obstructive pulmonary disease rates are led, in order, by India, China, Bulgaria, Nepal, Serbia

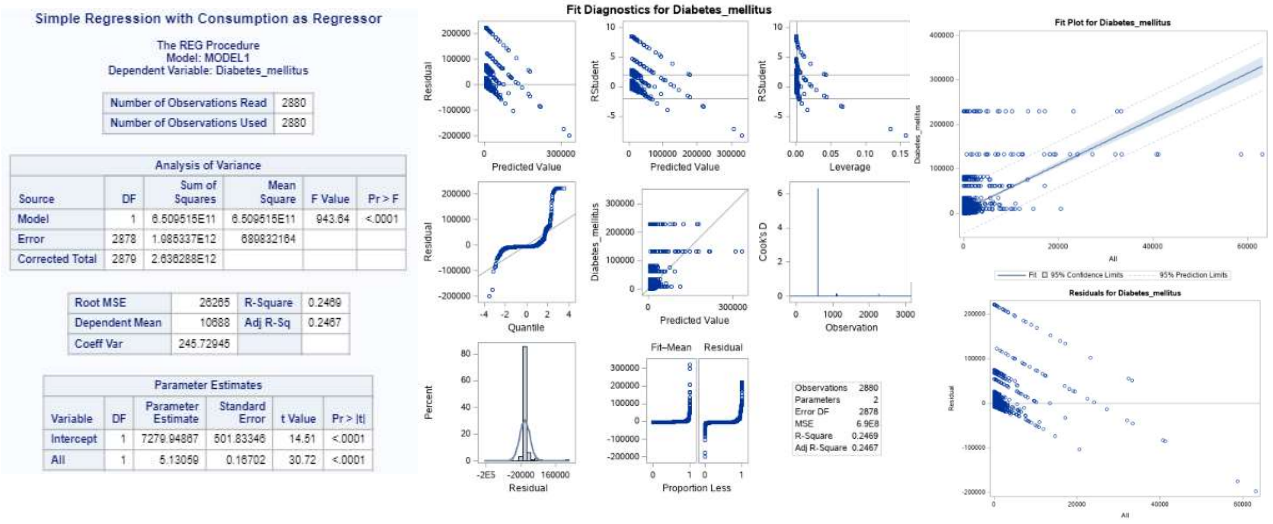
The differences in leading nations for each noncommunicable disease category opened a new possibility that different food category consumption rates could lead to higher rates of some noncommunicable diseases and not others. This was explored in the model building process.

IV. Simple Linear Regression

In order to determine whether the rate of diabetes differs across various consumption rates, a linear regression was performed. The assumptions of the model were examined with Levene's Test of Homogeneity, Q-Q plots, and residual plots. Levene's Test was chosen over the Bartlett Test because it is less sensitive "to departures from normality" (Croarkin and Tobias, 2013). This is a critical aspect as the data does not follow the normal curve, though the downfall is that Levene's Test is less sensitive. The Q-Q plot was selected to demonstrate whether the

distribution of the residuals is normal and a residual plot to examine the dispersion of the data. These metrics are to determine whether the relationship between the variables meets basic assumptions prior to building a model. The REG procedure was selected because it maximizes the code to statistical results ratio. While the UNIVARIATE procedure will also generate the Q-Q Plot, the REG procedure offers more plots and statistics for in-depth analysis with a small amount of code.

```
proc reg data=capstone.consumpfoodhealth;
    model diabetes_mellitus=all;
    title "Simple Regression with Consumption as Regressor";
run;
```



Simple Regression with Consumption as Regressor

The REG Procedure
Model: MODEL1
Dependent Variable: Diabetes_mellitus

Number of Observations Read	2880
Number of Observations Used	2880

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	6.509515E11	6.509515E11	943.64	<.0001
Error	2878	1.986337E12	690832164		
Corrected Total	2879	2.636288E12			

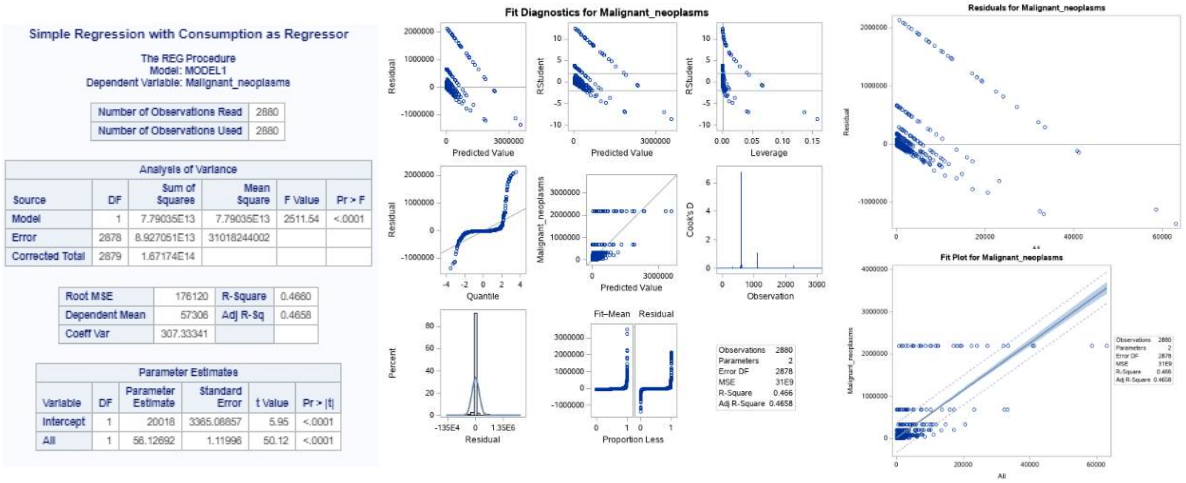
Root MSE	26285	R-Square	0.2469
Dependent Mean	10688	Adj R-Sq	0.2467
Coeff Var	245.72945		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	7279.94887	501.83346	14.51	<.0001
All	1	5.13059	0.16702	30.72	<.0001

The p-value above is less than .05, so the regression fits the model better than a baseline. From the analysis, there is a significant linear relationship between diabetes and consumption rates. The coefficient of determination indicates that consumption rates explain 24.6% of the variation in diabetes. The Q-Q Plot and histogram of residuals fall relatively close to a straight line and bell-shaped curve, meaning that the deviations deviate from normality insignificantly.

```
proc reg data=capstone.consumpfoodhealth;
    model Malignant_neoplasms=all;
    title "Simple Regression with Consumption as Regressor";
run;
```



Simple Regression with Consumption as Regressor

The REG Procedure
Model: MODEL1
Dependent Variable: Malignant_neoplasms

Number of Observations Read	2880
Number of Observations Used	2880

Analysis of Variance

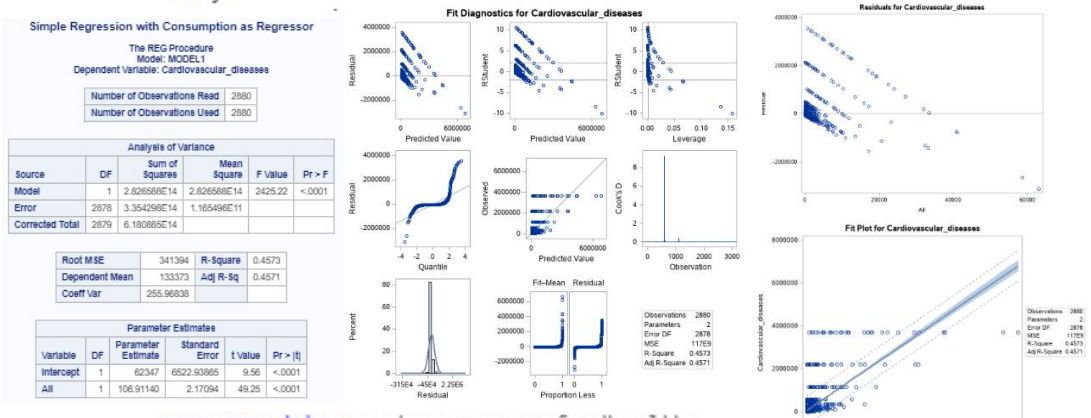
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7.79035E13	7.79035E13	2511.54	<.0001
Error	2878	8.927051E13	31018244002		
Corrected Total	2879	1.67174E14			

Root MSE	176120	R-Square	0.4660
Dependent Mean	57306	Adj R-Sq	0.4658
Coeff Var	307.33341		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	20018	3365.08857	5.95	<.0001
All	1	56.12692	1.11996	50.12	<.0001

```
proc reg data=capstone.consumpfoodhealth;
    model Cardiovascular_diseases=all;
    title "Simple Regression with Consumption as Regressor";
run;
```



Simple Regression with Consumption as Regressor

The REG Procedure
Model: MODEL1
Dependent Variable: Cardiovascular_diseases

Number of Observations Read	2880
Number of Observations Used	2880

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	2.826588E14	2.826588E14	2425.22	<.0001
Error	2878	3.354298E14	1.165496E11		
Corrected Total	2879	6.180886E14			

Root MSE	341394	R-Square	0.4573
Dependent Mean	133373	Adj R-Sq	0.4571
Coeff Var	255.96838		

Parameter Estimates

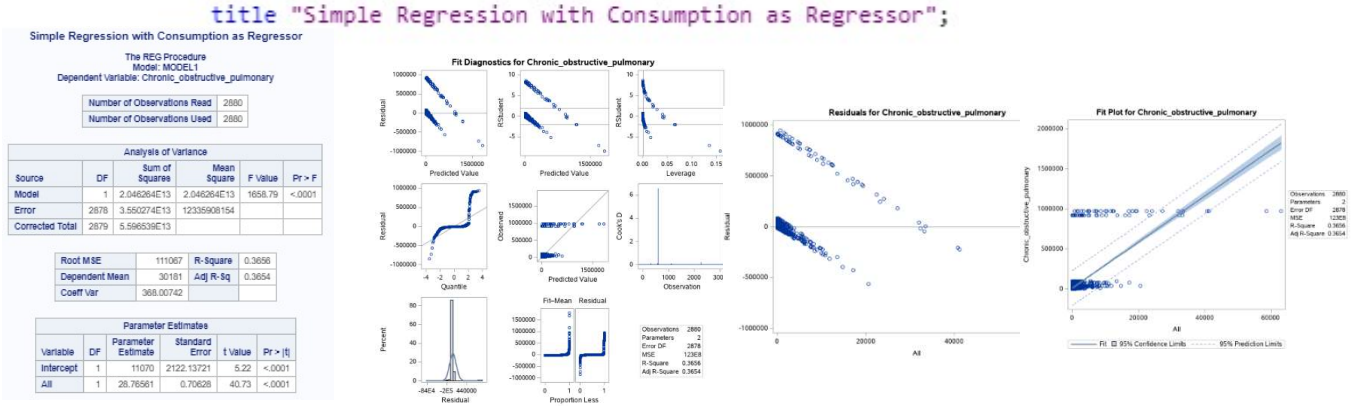
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	62347	6522.93865	9.56	<.0001
All	1	106.91140	2.17094	49.25	<.0001

Fit Diagnostics for Cardiovascular_diseases

Residuals for Cardiovascular_diseases

Fit Plot for Cardiovascular_diseases

```
proc reg data=capstone.consumpfoodhealth;
    model Chronic_obstructive_pulmonary=all;
    title "Simple Regression with Consumption as Regressor";
```



Simple Regression with Consumption as Regressor

The REG Procedure
Model: MODEL1
Dependent Variable: Chronic_obstructive_pulmonary

Number of Observations Read	2880
Number of Observations Used	2880

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	2.046264E13	2.046264E13	1658.79	<.0001
Error	2878	3.550274E13	12335908154		
Corrected Total	2879	5.596539E13			

Root MSE	111067	R-Square	0.3656
Dependent Mean	30181	Adj R-Sq	0.3654
Coeff Var	368.00742		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	11070	2122.13721	5.22	<.0001
All	1	28.76561	0.70628	40.73	<.0001

Fit Diagnostics for Chronic_obstructive_pulmonary

Residuals for Chronic_obstructive_pulmonary

Fit Plot for Chronic_obstructive_pulmonary

The above plots are very similar to the diabetes vs consumption rates. They are all related linearly and the models are better than the baseline. Conversely, the normality of each is not exact. These results indicated this relationship between consumption and health outcomes was strong.

E. Predictive Modeling Diabetes, Malignant Neoplasms, Cardiovascular Disease, Chronic Obstructive Pulmonary Disease

I. Splitting the Data

The model aims to identify predictors for high rates of NCDs. This was defined by the upper quantile or 25% of the data, in order to identify the upper quantile of the data as the target, a binary variable was created for each NCD. The data was then sorted by the diabetes mellitus binary variable and followed with the SURVEYSELECT procedure in order to split the data into training and validation data sets. SURVEYSELECT was chosen over the DATA step, which would have required more code and ended with a messier result. SURVEYSELECT was the

more efficient method. Below are the data splitting procedures for diabetes, malignant neoplasms, cardiovascular disease and chronic obstructive pulmonary disease respectively.

Diabetes:

```
data capstone.consumpfoodhealth;
    set capstone.consumpfoodhealth;
    if percapita_diabetes >=    2.52795E-04 then diabetes_bin = 1;
    else diabetes_bin =0;
    if percapita_neoplasms >= 0.000997668 then neoplasms_bin = 1;
    else neoplasms_bin =  0;
    if percapita_cardio >= 2.28153E-03 then cardio_bin = 1;
    else cardio_bin = 0;
    if percapita_pulmonary >=   3.80540E-04 then pul_bin = 1;
    else pul_bin = 0;
run;

proc sort data=capstone.consumpfoodhealth out=health_sort;
    by diabetes_bin;
run;

proc surveyselect noprint data=work.health_sort
        samprate=0.5 out=health_sample seed=27755
        outall stratumseed=restore;
    strata diabetes_bin;
run;

data capstone.diabetes_train (drop=selected)
     capstone.diabetes_valid (drop=selected);
     set health_sample;
     if selected then output capstone.diabetes_train;
     else output capstone.diabetes_valid;
run;
```

Malignant Neoplasms:

```
proc sort data=capstone.consumpfoodhealth out=health_sort;
    by neoplasms_bin;
run;

proc surveyselect noprint data=work.health_sort
        samprate=0.5 out=health_sample seed=27755
        outall stratumseed=restore;
    strata neoplasms_bin;
run;

data capstone.neoplasms_train (drop=selected)
     capstone.neoplasms_valid (drop=selected);
     set health_sample;
     if selected then output capstone.neoplasms_train;
     else output capstone.neoplasms_valid;
run;
```

Cardiovascular Disease:

```
proc sort data=capstone.consumpfoodhealth out=health_sort;
    by cardio_bin;
run;

proc surveyselect noprint data=work.health_sort
        samprate=0.5 out=health_sample seed=27755
        outall stratumseed=restore;
    strata cardio_bin;
run;

data capstone.cardio_train (drop=selected)
     capstone.cardio_valid (drop=selected);
     set health_sample;
     if selected then output capstone.cardio_train;
     else output capstone.cardio_valid;
run;
```

Chronic Obstructive Pulmonary Disease:

```
proc sort data=capstone.consumpfoodhealth out=health_sort;
    by pul_bin;
run;

proc surveyselect noprint data=work.health_sort
        samprate=0.5 out=health_sample seed=27755
        outall stratumseed=restore;
    strata pul_bin;
run;

data capstone.pul_train (drop=selected)
     capstone.pul_valid (drop=selected);
     set health_sample;
     if selected then output capstone.pul_train;
     else output capstone.pul_valid;
run;
```

In the above code, the data is split 50/50 into training and validation datasets. Additionally, per capita rates of the NCDs are used in order to highlight the differences in health outcomes as opposed to differences in population rates.

II. Handling Missing Values

Prior to creating a model, it is necessary to remove missing values. The following step uses the MEANS procedure to identify any missing values in the training dataset. This method proved the simplest and shortest method for calculating the number of missing values across the dataset. By using the original dataset and not the training and validation datasets to calculate missing values, this eliminated the need to repeat this step on the validation dataset.

```
proc means data=capstone.consumpfoodhealth
    nmiss n;
run;
```

This procedure demonstrated that there are no missing values in the dataset, and thus, no further adjustments are necessary.

III. Computing Smoothed Weight of Evidence

When dealing with a categorical predictor, computing a smoothed weight of evidence assists with creating a predictive model. Categorical variables cause problems if left alone due to high dimensionality and quasi-complete separation. In this case, the food category variable necessitated smoothed weight of evidence to turn it into a continuous variable. Smooth Weight of Evidence reduced the risk of dimensionality that the alternative method, dummy coding, would have caused. Using this statistical method insured that sampling variability was accounted for and overfitting was avoided. Below is the code for computing smoothed weight of evidence for diabetes, malignant neoplasms, cardiovascular disease and chronic obstructive pulmonary disease.

Diabetes:

```
%global rho1;
proc sql noprint;
    select mean(diabetes_bin) into :rho1
    from capstone.diabetes_train;
run;

proc means data=capstone.diabetes_train sum nway noprint;
    class cat;
    var diabetes_bin;
    output out=work.diab_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/diab_brswoe.sas";

data _null_;
    file brswoe;
    set work.diab_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.train_imputed_swoe_diab;
    set capstone.diabetes_train;
    %include brswoe /source2;
run;
```

Malignant Neoplasms:

```
%global rho1;
proc sql noprint;
    select mean(neoplasms_bin) into :rho1
    from capstone.neoplasms_train;
run;

proc means data=capstone.neoplasms_train sum nway noprint;
    class cat;
    var neoplasms_bin;
    output out=work.neoplasm_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/neoplasms_brswoe.sas";

data _null_;
    file brswoe;
    set work.neoplasm_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.train_imputed_swoe_neo;
    set capstone.neoplasms_train;
    %include brswoe /source2;
run;
```

Cardiovascular Disease:

```
%global rho1;
proc sql noprint;
    select mean(cardio_bin) into :rho1
    from capstone.cardio_train;
run;

proc means data=capstone.cardio_train sum nway noprint;
    class cat;
    var cardio_bin;
    output out=work.cardio_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/cardio_brswoe.sas";

data _null_;
    file brswoe;
    set work.cardio_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.train_imputed_swoe_cardio;
    set capstone.cardio_train;
    %include brswoe /source2;
run;
```

Chronic Obstructive Pulmonary:

```
%global rho1;
proc sql noprint;
    select mean(pul_bin) into :rho1
    from capstone.pul_train;
run;

proc means data=capstone.pul_train sum nway noprint;
    class cat;
    var pul_bin;
    output out=work.pul_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/pul_brswoe.sas";

data _null_;
    file brswoe;
    set work.pul_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.train_imputed_swoe_pul;
    set capstone.pul_train;
    %include brswoe /source2;
run;
```

The above data takes the frequency of the target variable in each category and takes the logit in order to create a continuous variable that will be utilized in the model building phase.

IV. Detecting Nonlinear Relationships

Due to the fact that logistic models are designed to best represent linear relationships, if nonlinear relationships exist in the data, adjustments must occur in order to proceed to model building. This study used Spearman's Correlation and Hoeffding's D statistics to evaluate whether the variable relationships were nonmonotonic with a high association, or nonlinear. Spearman's Correlation is used here as opposed to Pearson's for its lower sensitivity to outliers. Alternatively, TRANSREG can be used to model nonlinear relationships visually, but it does not illustrate the strength of these nonlinear relationships numerically nor is it possible to compare them easily across variables. Thus, Spearman's and Hoeffding's were selected. Below is the code for the detection of nonlinear relationships for diabetes:

```
ods select none;
ods output spearmancorr=work.spearman
           hoeffdingcorr=work.hoeffding;

proc corr data=work.train_imputed_swoe_diab spearman hoeffding;
    var diabetes_bin;
    with percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
run;

ods select all;

proc sort data=work.spearman;
    by variable;
run;

proc sort data=work.hoeffding;
    by variable;
run;

data work.correlations;
    merge work.spearman(rename=(diabetes_bin=scorr pdiabetes_bin=spvalue))
          work.hoeffding(rename=(diabetes_bin=hcorr pdiabetes_bin=hpvalue));
    by variable;
    scorr_abs=abs(scorr);
    hcorr_abs=abs(hcorr);
run;

proc rank data=work.correlations out=work.correlations1 descending;
    var scorr_abs hcorr_abs;
    ranks ranksp rankho;
run;

proc sort data=work.correlations1;
    by ranksp;
run;
```

```
title1 "Rank of Spearman Correlations and Hoeffding Correlations";
proc print data=work.correlations1 label split='*';
    var variable ranksp rankho scorr spvalue hcorr hpvalue;
    label ranksp = 'Spearman rank*of variables'
          scorr = 'Spearman Correlation'
          spvalue = 'Spearman p-value'
          rankho = 'Hoeffding rank*of variables'
          hcorr = 'Hoeffding Correlation'
          hpvalue = 'Hoeffding p-value';
run;

%global vref href;

proc sql noprint;
    select min(ranksp) into :vref
    from (select ranksp
    from work.correlations1
    having spvalue > .5);

    select min(rankho) into :href
    from (select rankho
    from work.correlations1
    having hpvalue> .5);
quit;

title1 "Scatter Plot of the Ranks of Spearman vs. Hoeffding";
proc sgplot data=work.correlations1;
    refine &vref / axis = y;
    refine &href /axis = x;
    scatter y=ranksp x=rankho / datalabel=variable;
    yaxis label="Rank of Spearman";
    xaxis label="Rank of Hoeffding";
run;
title1 ;

%global screened;
%let screened= percapita_all calories_median fat_median protein_median
    carbohydrate_median cat_swoe;
```

Malignant neoplasms:

```
ods select none;
ods output spearmancorr=work.spearman
           hoeffdingcorr=work.hoeffding;

proc corr data=work.train_imputed_swoe_neo spearman hoeffding;
    var neoplasms_bin;
    with percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
run;

ods select all;

proc sort data=work.spearman;
    by variable;
run;

proc sort data=work.hoeffding;
    by variable;
run;

data work.correlations;
    merge work.spearman(rename=(neoplasms_bin=scorr pneoplasms_bin=spvalue))
          work.hoeffding(rename=(neoplasms_bin=hcorr pneoplasms_bin=hpvalue));
    by variable;
    scorr_abs=abs(scorr);
    hcorr_abs=abs(hcorr);
run;

proc rank data=work.correlations out=work.correlations1 descending;
    var scorr_abs hcorr_abs;
    ranks ranksp rankho;
run;

proc sort data=work.correlations1;
    by ranksp;
run;

title1 "Rank of Spearman Correlations and Hoeffding Correlations";
proc print data=work.correlations1 label split='*';
    var variable ranksp rankho scorr spvalue hcorr hpvalue;
    label ranksp = 'Spearman rank*of variables'
          scorr = 'Spearman Correlation'
          spvalue = 'Spearman p-value'
          rankho = 'Hoeffding rank*of variables'
          hcorr = 'Hoeffding Correlation'
          hpvalue = 'Hoeffding p-value';
run;

%global vref href;
```

```
proc sql noprint;
    select min(ranksp) into :vref
    from (select ranksp
    from work.correlations1
    having spvalue > .5);

    select min(rankho) into :href
    from (select rankho
    from work.correlations1
    having hpvalue> .5);
quit;

title1 "Scatter Plot of the Ranks of Spearman vs. Hoeffding";
proc sgplot data=work.correlations1;
    refine &vref / axis = y;
    refine &href /axis = x;
    scatter y=ranksp x=rankho / datalabel=variable;
    yaxis label="Rank of Spearman";
    xaxis label="Rank of Hoeffding";
run;
title1 ;

%global screened;
%let screened= percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
```

Cardiovascular disease:

```
ods select none;
ods output spearmancorr=work.spearman
           hoeffdingcorr=work.hoeffding;

proc corr data=work.train_imputed_swoe_cardio spearman hoeffding;
    var cardio_bin;
    with percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
run;

ods select all;

proc sort data=work.spearman;
    by variable;
run;

proc sort data=work.hoeffding;
    by variable;
run;

data work.correlations;
    merge work.spearman(rename=(cardio_bin=scorr pneoplasms_bin=spvalue))
          work.hoeffding(rename=(cardio_bin=hcorr pneoplasms_bin=hpvalue));
    by variable;
    scorr_abs=abs(scorr);
    hcorr_abs=abs(hcorr);
run;

proc rank data=work.correlations out=work.correlations1 descending;
    var scorr_abs hcorr_abs;
    ranks ranksp rankho;
run;

proc sort data=work.correlations1;
    by ranksp;
run;

title1 "Rank of Spearman Correlations and Hoeffding Correlations";
proc print data=work.correlations1 label split='*';
    var variable ranksp rankho scorr spvalue hcorr hpvalue;
    label ranksp = 'Spearman rank*of variables'
          scorr = 'Spearman Correlation'
          spvalue = 'Spearman p-value'
          rankho = 'Hoeffding rank*of variables'
          hcorr = 'Hoeffding Correlation'
          hpvalue = 'Hoeffding p-value';
run;

%global vref href;
```

```
proc sql noprint;
    select min(ranksp) into :vref
    from (select ranksp
    from work.correlations1
    having spvalue > .5);

    select min(rankho) into :href
    from (select rankho
    from work.correlations1
    having hpvalue> .5);
quit;

title1 "Scatter Plot of the Ranks of Spearman vs. Hoeffding";
proc sgplot data=work.correlations1;
    refine &vref / axis = y;
    refine &href /axis = x;
    scatter y=ranksp x=rankho / datalabel=variable;
    yaxis label="Rank of Spearman";
    xaxis label="Rank of Hoeffding";
run;
title1 ;

%global screened;
%let screened= percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
```

Chronic obstructive pulmonary disease:

```
ods select none;
ods output spearmancorr=work.spearman
           hoeffdingcorr=work.hoeffding;

proc corr data=work.train_imputed_swoe_pul spearman hoeffding;
    var pul_bin;
    with percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
run;

ods select all;

proc sort data=work.spearman;
    by variable;
run;

proc sort data=work.hoeffding;
    by variable;
run;

data work.correlations;
    merge work.spearman(rename=(pul_bin=scorr pneoplasms_bin=spvalue))
          work.hoeffding(rename=(pul_bin=hcorr pneoplasms_bin=hpvalue));
    by variable;
    scorr_abs=abs(scorr);
    hcorr_abs=abs(hcorr);
run;

proc rank data=work.correlations out=work.correlations1 descending;
    var scorr_abs hcorr_abs;
    ranks ranksp rankho;
run;

proc sort data=work.correlations1;
    by ranksp;
run;

title1 "Rank of Spearman Correlations and Hoeffding Correlations";
proc print data=work.correlations1 label split='*';
    var variable ranksp rankho scorr spvalue hcorr hpvalue;
    label ranksp = 'Spearman rank*of variables'
          scorr = 'Spearman Correlation'
          spvalue = 'Spearman p-value'
          rankho = 'Hoeffding rank*of variables'
          hcorr = 'Hoeffding Correlation'
          hpvalue = 'Hoeffding p-value';
run;

%global vref href;
```

```
proc sql noprint;
    select min(ranksp) into :vref
    from (select ranksp
    from work.correlations1
    having spvalue > .5);

    select min(rankho) into :href
    from (select rankho
    from work.correlations1
    having hpvalue> .5);
quit;

title1 "Scatter Plot of the Ranks of Spearman vs. Hoeffding";
proc sgplot data=work.correlations1;
    refine &vref / axis = y;
    refine &href /axis = x;
    scatter y=ranksp x=rankho / datalabel=variable;
    yaxis label="Rank of Spearman";
    xaxis label="Rank of Hoeffding";
run;
title1 ;

%global screened;
%let screened= percapita_all calories_median fat_median protein_median carbohydrate_median cat_swoe;
```

In the above code, the CORR procedure was used to create two files with the Spearman and Hoeffding statistics respectively. The two files were sorted by the variables and then merged. Two new variables were added that take the absolute value of the Spearman and Hoeffding statistics in order to sort the relationships between variables. The RANK procedure was then used to rank the statistics prioritizing the smallest statistics with the DESENDING option. This procedure output the following results for diabetes:

Rank of Spearman Correlations and Hoeffding Correlations

Obs	Variable	Spearman rank of variables	Hoeffding rank of variables	Spearman Correlation	Spearman p-value	Hoeffding Correlation	Hoeffding p-value
1	cat_swoe	1	2	0.11109	<.0001	0.00095	0.0212
2	percapita_all	2	1	0.10712	<.0001	0.00101	0.0179
3	Carbohydrate_Median	3	4	-0.01739	0.5096	-0.00066	1.0000
4	Calories_Median	4	6	-0.01155	0.6615	-0.00063	1.0000
5	Fat_Median	5	5	-0.00618	0.8148	-0.00066	1.0000
6	Protein_Median	6	3	0.00108	0.9674	-0.00068	1.0000

Malignant neoplasms:

Rank of Spearman Correlations and Hoeffding Correlations

Obs	Variable	Spearman rank of variables	Hoeffding rank of variables	Spearman Correlation	Spearman p-value	Hoeffding Correlation	Hoeffding p-value
1	percapita_all	1	1	0.33906	<.0001	0.01585	<.0001
2	cat_swoe	2	2	0.11109	<.0001	0.00095	0.0212
3	Carbohydrate_Median	3	4	-0.01739	0.5096	-0.00066	1.0000
4	Calories_Median	4	6	-0.01155	0.6615	-0.00063	1.0000
5	Fat_Median	5	5	-0.00618	0.8148	-0.00066	1.0000
6	Protein_Median	6	3	0.00108	0.9674	-0.00068	1.0000

Cardiovascular disease:

Rank of Spearman Correlations and Hoeffding Correlations

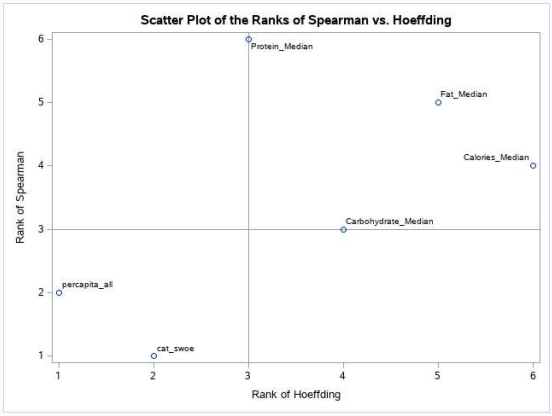
Obs	Variable	Spearman rank of variables	Hoeffding rank of variables	Spearman Correlation	Spearman p-value	Hoeffding Correlation	Hoeffding p-value
1	percapita_all	1	1	0.33906	<.0001	0.01585	<.0001
2	cat_swoe	2	2	0.11109	<.0001	0.00095	0.0212
3	Carbohydrate_Median	3	4	-0.01739	0.5096	-0.00066	1.0000
4	Calories_Median	4	6	-0.01155	0.6615	-0.00063	1.0000
5	Fat_Median	5	5	-0.00618	0.8148	-0.00066	1.0000
6	Protein_Median	6	3	0.00108	0.9674	-0.00068	1.0000

Chronic obstructive pulmonary:

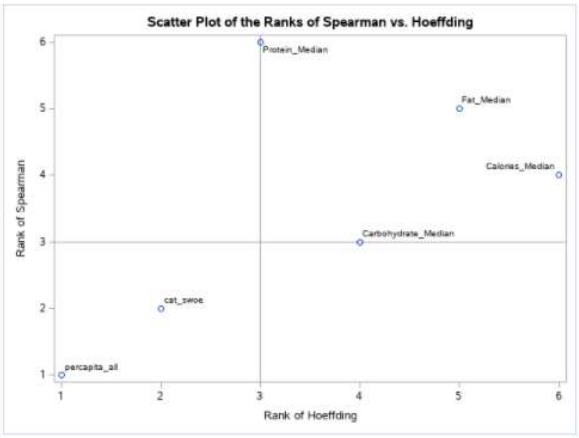
Rank of Spearman Correlations and Hoeffding Correlations

Obs	Variable	Spearman rank of variables	Hoeffding rank of variables	Spearman Correlation	Spearman p-value	Hoeffding Correlation	Hoeffding p-value
1	percapita_all	1	1	0.33906	<.0001	0.01585	<.0001
2	cat_swoe	2	2	0.11109	<.0001	0.00095	0.0212
3	Carbohydrate_Median	3	4	-0.01739	0.5096	-0.00066	1.0000
4	Calories_Median	4	6	-0.01155	0.6615	-0.00063	1.0000
5	Fat_Median	5	5	-0.00618	0.8148	-0.00066	1.0000
6	Protein_Median	6	3	0.00108	0.9674	-0.00068	1.0000

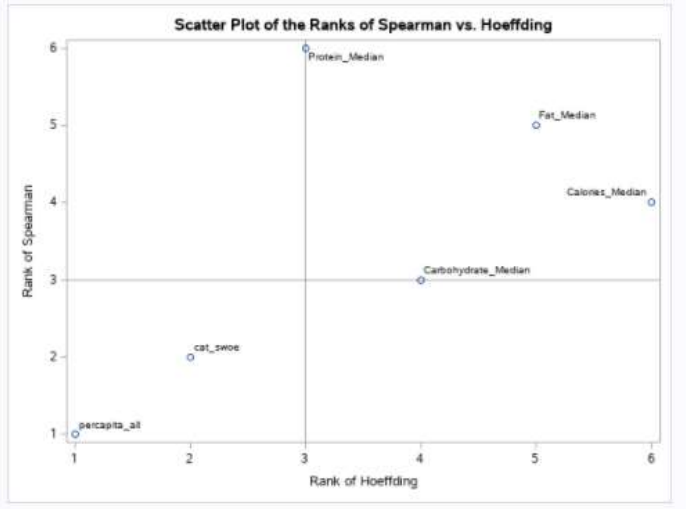
The ranks show that a nonlinear relationship likely did not exist. In order to highlight this further, a SQL procedure was used that highlighted the smallest Spearman and Hoeffding rank with a p-value greater than .5. Then SGPLOT was used to visually illustrate the statistics. The following graph was the output for diabetes:



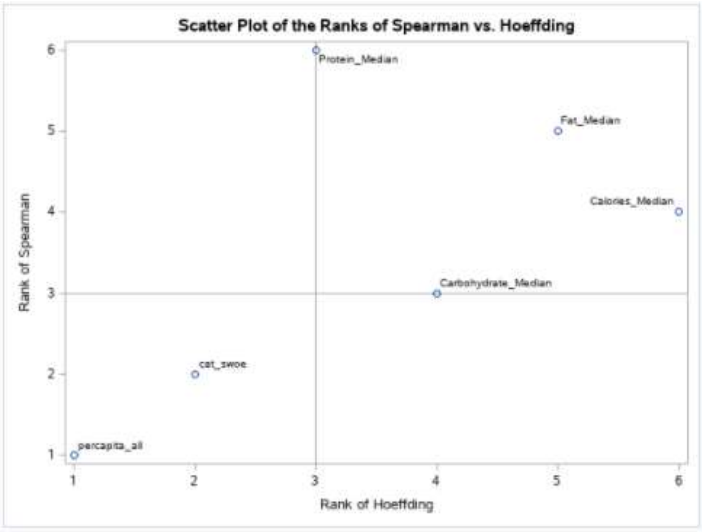
Malignant neoplasms:



Cardiovascular disease:



Chronic obstructive pulmonary:



Due to the fact that there are no variables in the upper left quadrant, indicating a high Hoeffding and low Spearman statistic. This meant there were no linear relationships that necessitated adjustment prior to model building.

V. Interaction Detection

The data preparation then necessitated the discovery of interactions between model building. Detecting these interactions would assist in the most comprehensive model possible. The following code was used to determine interactions across the data for diabetes:

```
title1 "P-Value for Entry and Retention";

%global sl;
proc sql;
    select 1-probchi(log(sum(diabetes_bin ge 0)),1) into :sl
    from work.train_imputed_swoe_diab;
quit;

title "Interaction Detection Using Forward Selection";
proc logistic data=work.train_imputed_swoe_diab;
    model diabetes_bin(event='1')= &screened
        percapita_all|calories_median|fat_median|protein_median|carbohydrate_median|cat_swoe @2
            /include=28 clodds=pl
            selection=forward slentry=&sl;
run;
```

Malignant neoplasms:

```
title1 "P-Value for Entry and Retention";

%global sl;
proc sql;
    select 1-probchi(log(sum(neoplasms_bin ge 0)),1) into :sl
    from work.train_imputed_swoe_neo;
quit;

title "Interaction Detection Using Forward Selection";
proc logistic data=work.train_imputed_swoe_neo;
    model neoplasms_bin(event='1')= &screened
        percapita_all|calories_median|fat_median|protein_median|carbohydrate_median|cat_swoe @2 /include=28 clodds=pl
            selection=forward slentry=&sl;
run;
```

Cardiovascular disease:

```
title1 "P-Value for Entry and Retention";

%global sl;
proc sql;
    select 1-probchi(log(sum(cardio_bin ge 0)),1) into :sl
    from work.train_imputed_swoe_cardio;
quit;

title "Interaction Detection Using Forward Selection";
proc logistic data=work.train_imputed_swoe_cardio;
    model cardio_bin(event='1')= &screened
        percapita_all|calories_median|fat_median|protein_median|carbohydrate_median|cat_swoe @2 /include=28 clodds=pl
            selection=forward slentry=&sl;
run;
```

Chronic obstructive pulmonary:

```
title1 "P-Value for Entry and Retention";

%global sl;
proc sql;
    select 1-probchi(log(sum(pul_bin ge 0)),1) into :sl
    from work.train_imputed_swoe_pul;
quit;

title "Interaction Detection Using Forward Selection";
proc logistic data=work.train_imputed_swoe_pul;
    model pul_bin(event='1')= &screened
        percapita_all|calories_median|fat_median|protein_median|carbohydrate_median|cat_swoe @2 /include=28 clodds=pl
            selection=forward slentry=&sl;
run;
```

The above code first uses SQL to determine the log of the number of observations. This figure would be the p-value utilized for entry and retention. The next procedure used forward selection in order to detect important two-factor interactions. This method was selected over the EFFECTPLOT procedure, which would have required a macro to iterate through the procedure

for as many interaction plots necessitating presentation. Forward selection was chosen above backward selection because it is more efficient and does not have to consider the full model. Backward selection would have been selected if collinearity were present, in order to keep those variables in the model, but it was not. Finally, Stepwise selection was also considered, but due to the biased coefficients and p-values it produces, was not chosen in the end (Choueiry, 2020). The purpose of this method was to identify key interactions in the dataset that could be verified and selected with further analysis. The following tables show the results of interaction detection for diabetes, malignant neoplasms, cardiovascular disease, and chronic obstructive pulmonary.

Diabetes:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.8117	0.8955	0.8216	0.3647
percapita_all	1	-0.0106	0.0131	0.6546	0.4185
Calories_Median	1	-0.00211	0.0226	0.0087	0.9256
Fat_Median	1	0.0238	0.1984	0.0143	0.9047
Protein_Median	1	-0.00157	0.1041	0.0002	0.9880
Carbohydrate_Median	1	0.0148	0.0913	0.0262	0.8714
cat_swoe	1	1.6889	0.8064	4.3862	0.0362
percapita*Calories_M	1	0.000084	0.000151	0.3137	0.5754
percapita*Fat_Median	1	-0.00066	0.00133	0.2427	0.6223
Calories_*Fat_Median	1	1.959E-7	0.000023	0.0001	0.9932
percapita*Protein_Me	1	-0.00013	0.000705	0.0363	0.8490
Calories_*Protein_Me	1	0.000076	0.000250	0.0913	0.7625
Fat_Media*Protein_Me	1	-0.00057	0.00240	0.0571	0.8112
percapita*Carbohydra	1	-0.00021	0.000577	0.1306	0.7178
Calories_*Carbohydra	1	0.000014	0.000034	0.1683	0.6816
Fat_Media*Carbohydra	1	5.507E-6	0.000360	0.0002	0.9878
Protein_M*Carbohydra	1	-0.00040	0.000909	0.1921	0.6612
percapita_a*cat_swoe	1	-0.00990	0.0129	0.5929	0.4413
Calories_Me*cat_swoe	1	-0.00078	0.0184	0.0018	0.9664
Fat_Median*cat_swoe	1	0.0139	0.1627	0.0073	0.9319
Protein_Med*cat_swoe	1	0.00660	0.0960	0.0047	0.9452
Carbohydrat*cat_swoe	1	0.0137	0.0760	0.0325	0.8569

Malignant neoplasms:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.2785	0.9871	0.0796	0.7778
percapita_all	1	-0.0186	0.0159	1.3717	0.2415
Calories_Median	1	0.0318	0.0240	1.7596	0.1847
Fat_Median	1	-0.2023	0.2117	0.9135	0.3392
Protein_Median	1	-0.0976	0.1079	0.8177	0.3659
Carbohydrate_Median	1	-0.1201	0.0967	1.5450	0.2139
cat_swoe	1	1.9548	0.8929	4.7930	0.0286
percapita*Calories_M	1	0.000042	0.000138	0.0938	0.7594
percapita*Fat_Median	1	0.000065	0.00124	0.0047	0.9453
Calories_*Fat_Median	1	-0.00001	0.000026	0.3097	0.5778
percapita*Protein_Me	1	-0.00140	0.000735	3.6327	0.0567
Calories_*Protein_Me	1	-0.00020	0.000267	0.5557	0.4560
Fat_Media*Protein_Me	1	0.00165	0.00257	0.4123	0.5208
percapita*Carbohydra	1	-0.00071	0.000534	1.7671	0.1837
Calories_*Carbohydra	1	-1.4E-6	0.000035	0.0016	0.9679
Fat_Media*Carbohydra	1	0.000300	0.000377	0.6308	0.4270
Protein_M*Carbohydra	1	0.000629	0.000968	0.4228	0.5155
percapita_a*cat_swoe	1	-0.0578	0.0158	13.4173	0.0002
Calories_Me*cat_swoe	1	0.0234	0.0195	1.4396	0.2302
Fat_Median*cat_swoe	1	-0.1333	0.1731	0.5934	0.4411
Protein_Med*cat_swoe	1	-0.0922	0.1001	0.8494	0.3567
Carbohydrat*cat_swoe	1	-0.0954	0.0801	1.4169	0.2339

Cardiovascular disease:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-0.0646	1.0104	0.0041	0.9490
percapita_all	1	0.0122	0.0191	0.4095	0.5222
Calories_Median	1	0.0304	0.0237	1.6434	0.1999
Fat_Median	1	-0.2160	0.2083	1.0750	0.2998
Protein_Median	1	-0.1348	0.1075	1.5738	0.2097
Carbohydrate_Median	1	-0.1151	0.0958	1.4449	0.2293
cat_swoe	1	1.4246	0.9123	2.4385	0.1184
percapita*Calories_M	1	0.000334	0.000179	3.4892	0.0625
percapita*Fat_Median	1	-0.00291	0.00159	3.3404	0.0676
Calories_*Fat_Median	1	-1.37E-6	0.000024	0.0031	0.9553
percapita*Protein_Me	1	-0.00114	0.000884	1.6642	0.1970
Calories_*Protein_Me	1	-0.00016	0.000267	0.3667	0.5448
Fat_Media*Protein_Me	1	0.00164	0.00256	0.4129	0.5205
percapita*Carbohydra	1	-0.00185	0.000703	6.9614	0.0083
Calories_*Carbohydra	1	0.000012	0.000034	0.1160	0.7334
Fat_Media*Carbohydra	1	0.000112	0.000367	0.0940	0.7591
Protein_M*Carbohydra	1	0.000695	0.000969	0.5154	0.4728
percapita_a*cat_swoe	1	-0.0170	0.0180	0.8849	0.3469
Calories_Me*cat_swoe	1	0.0244	0.0193	1.5925	0.2070
Fat_Median*cat_swoe	1	-0.1654	0.1710	0.9352	0.3335
Protein_Med*cat_swoe	1	-0.1110	0.0992	1.2541	0.2628
Carbohydrat*cat_swoe	1	-0.0953	0.0795	1.4342	0.2311

Chronic obstructive pulmonary:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-0.3533	0.7413	0.2271	0.6337
percapita_all	1	0.0856	0.0254	11.3868	0.0007
Calories_Median	1	0.0215	0.0171	1.5859	0.2079
Fat_Median	1	-0.1774	0.1454	1.4898	0.2222
Protein_Median	1	-0.1085	0.1441	0.5665	0.4517
Carbohydrate_Median	1	-0.0856	0.0584	2.1498	0.1426
cat_swoe	1	1.0358	0.6204	2.7875	0.0950
percapita*Calories_M	1	-0.00008	0.000137	0.3313	0.5649
percapita*Fat_Median	1	0.000724	0.00123	0.3437	0.5577
Calories_*Fat_Median	1	-0.00007	0.000097	0.4895	0.4842
percapita*Protein_Me	1	-0.00045	0.000653	0.4849	0.4954
Calories_*Protein_Me	1	0.000484	0.000346	1.9570	0.1618
Fat_Media*Protein_Me	1	-0.00507	0.00405	1.5651	0.2109
percapita*Carbohydra	1	0.000264	0.000534	0.2448	0.6208
Calories_*Carbohydra	1	4.334E-6	0.000040	0.0118	0.9135
Fat_Media*Carbohydra	1	0.000034	0.000320	0.0115	0.9145
Protein_M*Carbohydra	1	-0.00147	0.00116	1.6188	0.2033
percapita_a*cat_swoe	1	0.0558	0.0213	6.8524	0.0089
Calories_Me*cat_swoe	1	0.0225	0.0189	1.4151	0.2342
Fat_Median*cat_swoe	1	-0.2291	0.1750	1.7137	0.1905
Protein_Med*cat_swoe	1	-0.0685	0.1599	0.1835	0.6684
Carbohydrat*cat_swoe	1	-0.0879	0.0642	1.8766	0.1707

The above are the results of the interaction detection procedures. This illustrated that 21 interactions were detected. Once interactions were identified, this necessitated further exploration.

VI. Best Subsets Selection Method

In order to determine the best model for each size, the best subset selection was used. Best subsets selection was chosen over stepwise because of the more in-depth information it presents. Stepwise selection would have arrived at a single best model, whereas best subset selection presents the best models for each number of variables, allowing the analyst to balance knowledge of the data, complexity, and best fit. This method selected the best model for each variable size based on the chi-square statistic. The chi-square statistic compares the actual data to the model predictions, causing it to be a useful tool in model selection. The following depicts the code for diabetes:

```
title1 "Models Selected by Best Subsets Selection";
proc logistic data=work.train_imputed_swoe_diab;
    model diabetes_bin(event='1')=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe / selection=score best=1;
run;
```

Models Selected by Best Subsets Selection

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_DIAB
Response Variable	diabetes_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	diabetes_bin	Total Frequency
1	0	1088
2	1	352

Probability modeled is diabetes_bin=1.

Regression Models Selected by Score Criterion

Number of Variables	Score Chi-Square	Variables Included in Model
1	18.9630	cat_swoe
2	25.2330	cat_swoe percapita_a*cat_swoe
3	26.0588	percapita_all cat_swoe percapita_a*cat_swoe
4	26.6359	Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median
5	26.9416	percapita_all Calories_Median cat_swoe percapita*Calories_M percapita_a*cat_swoe
6	27.2940	percapita_all Calories_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita_a*cat_swoe
7	27.5056	percapita_all Calories_Median Fat_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita_a*cat_swoe
8	27.6758	percapita_all Calories_Median Fat_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Carbohydra percapita_a*cat_swoe
9	27.7929	percapita_all Calories_Median Fat_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
10	27.8431	percapita_all Calories_Median Fat_Median Protein_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
11	27.8439	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe

Malignant neoplasms:

```
title1 "Models Selected by Best Subsets Selection";
proc logistic data=work.train_imputed_swoe_neo;
    model neoplasms_bin(event='1')=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe / selection=score best=1;
run;
```

Models Selected by Best Subsets Selection

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_NEO
Response Variable	neoplasms_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	neoplasms_bin	Total Frequency
1	0	1088
2	1	352

Probability modeled is neoplasms_bin=1.

Regression Models Selected by Score Criterion		
Number of Variables	Score Chi-Square	Variables Included in Model
1	92.7161	percapita_all
2	110.1637	cat_swoe percapita_a*cat_swoe
3	125.2476	cat_swoe percapita*Carbohydra percapita_a*cat_swoe
4	131.5975	cat_swoe percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
5	137.0656	Carbohydrate_Median cat_swoe percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
6	140.1060	Calories_Median Fat_Median cat_swoe percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
7	145.7655	Calories_Median Fat_Median cat_swoe percapita*Calories_M percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
8	146.7708	percapita_all Calories_Median Fat_Median cat_swoe percapita*Calories_M percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
9	147.0733	percapita_all Calories_Median Fat_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
10	147.0652	percapita_all Calories_Median Fat_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
11	147.1362	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe

Cardiovascular disease:

```
title1 "Models Selected by Best Subsets Selection";
proc logistic data=work.train_imputed_swoe_cardio;
    model cardio_bin(event='1')=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe / selection=score best=1;
run;
```

Models Selected by Best Subsets Selection

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_CARDIO
Response Variable	cardio_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	cardio_bin	Total Frequency
1	0	1088
2	1	352

Probability modeled is cardio_bin=1.

Regression Models Selected by Score Criterion		
Number of Variables	Score Chi-Square	Variables Included in Model
1	73.5073	percapita_all
2	99.5051	percapita_all percapita*Carbohydra
3	110.0535	cat_swoe percapita*Carbohydra percapita_a*cat_swoe
4	117.7624	Carbohydrate_Median cat_swoe percapita*Carbohydra percapita_a*cat_swoe
5	118.4791	Carbohydrate_Median cat_swoe percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
6	121.5290	percapita_all Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Carbohydra
7	124.9750	percapita_all Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra
8	125.8475	percapita_all Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra
9	126.2777	percapita_all Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
10	126.3510	percapita_all Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe
11	126.5416	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe

Chronic obstructive pulmonary:

```
data work.train_imputed_swoe_pul;
    set work.train_imputed_swoe_pul;
run;

title1 "Models Selected by Best Subsets Selection";
proc logistic data=work.train_imputed_swoe_pul;
    model pul_bin(event='1')=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe / selection=score best=1;
run;
```

Models Selected by Best Subsets Selection

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_PUL
Response Variable	pul_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	pul_bin	Total Frequency
1	0	1072
2	1	368

Probability modeled is pul_bin=1.

Regression Models Selected by Score Criterion		
Number of Variables	Score Chi-Square	Variables Included in Model
1	27.0328	percapita_all
2	59.8582	percapita_all cat_swoe
3	62.9560	percapita_all cat_swoe percapita_a*cat_swoe
4	67.9741	percapita_all cat_swoe percapita*Protein_Me percapita_a*cat_swoe
5	68.5021	percapita_all Fat_Median cat_swoe percapita*Protein_Me percapita_a*cat_swoe
6	68.9011	percapita_all Calories_Median Carbohydrate_Median cat_swoe percapita*Protein_Me percapita_a*cat_swoe
7	69.4462	percapita_all Calories_Median Fat_Median Carbohydrate_Median cat_swoe percapita*Protein_Me percapita_a*cat_swoe
8	69.6465	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Protein_Me percapita_a*cat_swoe
9	69.7265	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Fat_Median percapita*Protein_Me percapita_a*cat_swoe
10	69.7639	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita_a*cat_swoe
11	69.8589	percapita_all Calories_Median Fat_Median Protein_Median Carbohydrate_Median cat_swoe percapita*Calories_M percapita*Fat_Median percapita*Protein_Me percapita*Carbohydra percapita_a*cat_swoe

The above code uses the LOGISTIC procedure to conduct best subsets selection. Interestingly, cat_swoe and percapita_all, as well as their interaction are front runners in the model selection. This confirms the initial hypothesis that food category consumption rates may be a predictor for noncommunicable disease. The scoring technique used is not the best method for determining final model as it is skewed by model size. Therefore, the study continues with fit statistics to determine the best model.

VII. Fit Statistics for Model Selection

Fit Statistics were selected as the most comprehensive method in model selection. The following code develops a FITSTAT macro to generate the fit statistics. FITSTAT in the LOGISTIC procedure was selected above the PHREG procedure, which would have depicted the Akaike Information Criterion, Schwarz Bayesian Criterion, and -2 log likelihood statistic. This occurred because the PHREG procedure did not offer enough statistics. An iterative loop is used to score

each of the models generated and gather the statistics in a single file. Below is the loop code for diabetes:

```
%macro fitstat(data=, target=, event=, inputs=, best=, priorevent=);

ods select none;
ods output bestsubsets=work.score;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&inputs / selection=sore best=&best;
run;

proc sql noprint;
    select variablesinmodel into :inputs1 -
    from work.score;

    select NumberOfVariables into :ic1 -
    from work.score;
quit;

%let lastindx=&SQLOBS;

%do model_indx=1 %to &lastindx;

%let im=&&inputs&model_indx;
%let ic=&&ic&model_indx;

ods output scorefitstat=work.stat&ic;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&im;
    score data=&data out=work.scored fitstat
        priorevent=&priorevent;
run;
ods output scorefitstat=work.stat&ic;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&im;
    score data=&data out=work.scored fitstat
        priorevent=&priorevent;
run;

proc datasets
    library=work
    nodetails
    nolist;
    delete scored;
run;
quit;

%end;

data work.modelfit;
    set work.stat1 - work.stat&lastindx;
    model=_n_;
run;

%mend fitstat;

%fitstat(data=work.train_imputed_swoe_diab, target=diabetes_bin, event=1, inputs=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe, best=1, priorevent=0.02);

proc sort data=work.modelfit;
    by bic;
run;

title1 "Fit Statistics from Models Selected from Best Subseets";
ods select all;
proc print data=work.modelfit;
    var model auc aic bic misclass adjrsquare brierscore;
run;
```

Malignant neoplasms:

```
%macro fitstat(data=, target=, event=, inputs=, best=, priorevent=);

ods select none;
ods output bestsubsets=work.score;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&inputs / selection=sore best=&best;
run;

proc sql noprint;
    select variablesinmodel into :inputs1 -
    from work.score;

    select NumberOfVariables into :ic1 -
    from work.score;
quit;

%let lastindx=&SQLOBS;

%do model_indx=1 %to &lastindx;

%let im=&&inputs&model_indx;
%let ic=&&ic&model_indx;

ods output scorefitstat=work.stat&ic;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&im;
    score data=&data out=work.scored fitstat
        priorevent=&priorevent;
run;

proc datasets
    library=work
    nodetails
    nolist;
    delete scored;
run;
quit;

%end;

data work.modelfit;
    set work.stat1 - work.stat&lastindx;
    model=_n_;
run;

%mend fitstat;

%fitstat(data=work.train_imputed_swoe_neo, target=neoplasms_bin, event=1, inputs=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe, best=1, priorevent=0.02);

proc sort data=work.modelfit;
     by bic;
run;

title1 "Fit Statistics from Models Selected from Best Subseets";
ods select all;
proc print data=work.modelfit;
     var model auc aic bic misclass adjrsquare brierscore;
run;

%global selected;
proc sql;
     select VariablesInModel into :selected
     from work.score;
     where numberofvariables=35;
quit;
```

Cardiovascular disease:

```
%macro fitstat(data=, target=, event=, inputs=, best=, priorevent=);

ods select none;
ods output bestsubsets=work.score;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&inputs / selection=sore best=&best;
run;

proc sql noprint;
    select variablesinmodel into :inputs1 -
    from work.score;

    select NumberOfVariables into :ic1 -
    from work.score;
quit;

%let lastindx=&SQLOBS;

%do model_indx=1 %to &lastindx;

%let im=&&inputs&model_indx;
%let ic=&&ic&model_indx;

ods output scorefitstat=work.stat&ic;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&im;
    score data=&data out=work.scored fitstat
        priorevent=&priorevent;
run;

proc datasets
    library=work
    nodetails
    nolist;
    delete scored;
run;
quit;

%end;

data work.modelfit;
    set work.stat1 - work.stat&lastindx;
    model=_n_;
run;

%mend fitstat;

%fitstat(data=work.train_imputed_swoe_cardio, target=cardio_bin, event=1, inputs=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe, best=1, priorevent=0.02);

proc sort data=work.modelfit;
    by bic;
run;

title1 "Fit Statistics from Models Selected from Best Subseets";
ods select all;
proc print data=work.modelfit;
    var model auc aic bic misclass adjrsquare brierscore;
run;

%global selected;
proc sql;
    select VariablesInModel into :selected
    from work.score;
    where numberofvariables=35;
quit;
```

Chronic obstructive pulmonary:

```
%macro fitstat(data=, target=, event=, inputs=, best=, priorevent=);

ods select none;
ods output bestsubsets=work.score;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&inputs / selection=sore best=&best;
run;

proc sql noprint;
    select variablesinmodel into :inputs1 -
    from work.score;

    select NumberOfVariables into :ic1 -
    from work.score;
quit;

%let lastindx=&SQLOBS;

%do model_indx=1 %to &lastindx;

%let im=&&inputs&model_indx;
%let ic=&&ic&model_indx;

ods output scorefitstat=work.stat&ic;

proc logistic data=&data namelen=50;
    model &target(event="&event")=&im;
    score data=&data out=work.scored fitstat
        priorevent=&priorevent;
run;

proc datasets
    library=work
    nodetails
    nolist;
    delete scored;
run;
quit;

%end;

data work.modelfit;
    set work.stat1 - work.stat&lastindx;
    model=_n_;
run;

%mend fitstat;

%fitstat(data=work.train_imputed_swoe_pul, target=neoplasms_bin, event=1, inputs=&screened
        percapita_all*calories_median percapita_all*fat_median
        percapita_all*protein_median percapita_all*carbohydrate_median
        percapita_all*cat_swoe, best=1, priorevent=0.02);

proc sort data=work.modelfit;
    by bic;
run;

title1 "Fit Statistics from Models Selected from Best Subseets";
ods select all;
proc print data=work.modelfit;
    var model auc aic bic misclass adjrsquare brierscore;
run;

%global selected;
proc sql;
    select VariablesInModel into :selected
    from work.score;
    where numberofvariables=35;
quit;
```

The above code generated all possible models with fit statistics for each and then saved the results into a file called modelfit. The file was then sorted by the Bayesian Information Criterion and the results were printed.

Fit statistics for diabetes:

Fit Statistics from Models Selected from Best Subseets

Obs	model	AUC	AIC	BIC	MisClass	AdjRSquare	BrierScore
1	2	0.59672	2771.732	2787.549	0.2444	0.025884	0.234232
2	1	0.574566	2777.742	2788.287	0.2444	0.019518	0.234408
3	3	0.59531	2772.559	2793.649	0.2444	0.026814	0.234205
4	4	0.593309	2773.979	2800.341	0.2444	0.027273	0.234156
5	5	0.595358	2775.538	2807.173	0.2444	0.027622	0.234163
6	6	0.594738	2777.072	2813.979	0.2444	0.027991	0.234141
7	7	0.594921	2778.843	2821.022	0.2444	0.028172	0.234133
8	8	0.595781	2780.64	2828.092	0.2444	0.028333	0.234127
9	9	0.595621	2782.492	2835.216	0.2444	0.02845	0.23412
10	10	0.595118	2784.419	2842.415	0.2444	0.028508	0.234122
11	11	0.595126	2786.418	2849.687	0.2444	0.028508	0.234122

As seen from the above diabetes table, the model with a low Bayesian Information Criterion combined with a sufficiently high adjusted r-square value is the model with three variables: percapita_all, cat_swoe, percapita_all*cat_swoe. When analyzing the fit statistics, the adjusted r-square values for diabetes decrease at the second model, then increase only marginally after the third. For this reason, the third model was selected.

Malignant neoplasms:

Fit Statistics from Models Selected from Best Subseets

Obs	model	AUC	AIC	BIC	MisClass	AdjRSquare	BrierScore
1	5	0.722301	2616.336	2647.971	0.2444	0.146898	0.227815
2	6	0.72673	2612.756	2649.663	0.2444	0.150844	0.227618
3	7	0.729576	2609.485	2651.665	0.2451	0.154557	0.227561
4	4	0.714034	2625.7	2652.062	0.2451	0.138814	0.228454
5	3	0.708629	2631.073	2652.163	0.2438	0.133535	0.228672
6	8	0.725173	2611.008	2658.46	0.2444	0.154892	0.227486
7	9	0.725885	2613.005	2665.729	0.2444	0.154895	0.227565
8	2	0.710909	2653.866	2669.684	0.2465	0.115583	0.231016
9	10	0.725925	2614.918	2672.914	0.2444	0.154956	0.227557
10	11	0.725656	2616.677	2679.946	0.2444	0.155125	0.227511
11	1	0.727738	2672.915	2683.46	0.2465	0.100098	0.231724

In the above malignant neoplasms table, the model with a low Bayesian Information Criterion and high adjusted r-square value is the model with three variables: cat_swoe, percapita_all*Carbohydrate_Median, and percapita_all*cat_swoe. For this NCD's fit statistics, the adjusted r-square values begin to decrease after the third model, then rise again only slightly after the fifth model. The third model was selected here again.

Cardiovascular disease:

Fit Statistics from Models Selected from Best Subseets

Obs	model	AUC	AIC	BIC	MisClass	AdjRSquare	BrierScore
1	4	0.697823	2645.14	2671.502	0.2444	0.124836	0.228934
2	5	0.696737	2646.624	2678.258	0.2444	0.12521	0.228751
3	3	0.684873	2657.265	2678.354	0.2458	0.114562	0.229621
4	6	0.702594	2644.999	2681.906	0.2444	0.127829	0.228086
5	2	0.696156	2668.452	2684.269	0.2458	0.104877	0.229823
6	7	0.707362	2642.273	2684.453	0.2465	0.131235	0.228697
7	8	0.706239	2642.857	2690.309	0.2451	0.132253	0.228465
8	9	0.704105	2644.808	2697.532	0.2451	0.132289	0.228626
9	10	0.704284	2646.792	2704.789	0.2451	0.1323	0.228626
10	11	0.703536	2648.189	2711.458	0.2451	0.132733	0.228571
11	1	0.715764	2704.223	2714.768	0.2451	0.076643	0.232564

For cardiovascular disease, the model with a low Bayesian Information Criterion combined and high adjusted r-square value is the model with four variables: Carbohydrate_Median, cat_swoe, percapita_all*Carbohydrate_Median, and percapita_all*cat_swoe. Looking at the fit statistics, the adjusted r-square values begin to decrease at the third model, rise for the fourth, then decrease again before a marginal improvement with more complex models. For this case, the fourth model was selected.

Chronic obstructive pulmonary:

Fit Statistics from Models Selected from Best Subseets

Obs	model	AUC	AIC	BIC	MisClass	AdjRSquare	BrierScore
1	5	0.754086	2558.822	2590.456	0.2417	0.171288	0.223326
2	3	0.732438	2570.228	2591.318	0.2410	0.160541	0.223274
3	4	0.748456	2569.126	2595.488	0.2438	0.162714	0.22443
4	6	0.747242	2559.49	2596.397	0.2417	0.172211	0.223241
5	2	0.734119	2584.498	2600.315	0.2424	0.149066	0.225432
6	7	0.74919	2559.345	2601.524	0.2417	0.173697	0.22339
7	1	0.746089	2593.387	2603.931	0.2424	0.141314	0.225993
8	8	0.747692	2558.063	2605.514	0.2417	0.175967	0.223072
9	9	0.747638	2560.038	2612.762	0.2417	0.175984	0.223074
10	10	0.746887	2561.604	2619.6	0.2410	0.176283	0.222985
11	11	0.746267	2562.761	2626.029	0.2410	0.176865	0.222857

In the case of chronic obstructive pulmonary disease, the best model based on the outlined criteria was the model with four variables: percapita_all, percapita_all*Fat_Median, percapita_all*Protein_Median, and percapita_all*cat_swoe. When considering the fit statistics, the adjusted r-square value rises until the fourth variable, then decreases, with more complex models maintaining only a marginal improvement to the four variable model r-square value. In the end, these models were chosen because they balanced lower complexity, representation of the data, and the Bayesian Information Criterion.

VIII. Model Selection

Each selected model was then trained with the LOGISTIC procedure to examine the model further.

Diabetes:

```
proc logistic data=work.train_imputed_swoe_diab;
    model diabetes_bin(event='1') = percapita_all cat_swoe percapita_all*cat_swoe;
run;
```

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_DIAB
Response Variable	diabetes_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	diabetes_bin	Total Frequency
1	0	1088
2	1	352

Probability modeled is diabetes_bin=1.

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	1603.709	1583.690
SC	1608.982	1604.780
-2 Log L	1601.709	1575.690

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > Chi Sq
Likelihood Ratio	26.0189	3	<.0001
Score	26.0588	3	<.0001
Wald	25.5140	3	<.0001

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > Chi Sq
Intercept	1	0.9233	0.4913	3.5319	0.0602
percapita_all	1	-0.0134	0.0121	1.2386	0.2657
cat_swoe	1	1.9069	0.4382	18.9408	<.0001
percapita_a*cat_swoe	1	-0.0171	0.0110	2.4189	0.1199

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	59.5	Somers' D	0.191
Percent Discordant	40.5	Gamma	0.191
Percent Tied	0.0	Tau-a	0.070
Pairs	382976	c	0.595

Malignant neoplasms:

```
proc logistic data=work.train_imputed_swoe_neo;
    model neoplasms_bin(event='1') = cat_swoe percapita_all*Carbohydrate_Median
    percapita_all*cat_swoe;
run;
```

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_NEO
Response Variable	neoplasms_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	neoplasms_bin	Total Frequency
1	0	1088
2	1	352

Probability modeled is neoplasms_bin=1.

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	1603.709	1491.897
SC	1608.982	1512.986
-2 Log L	1601.709	1483.897

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > Chi Sq
Likelihood Ratio	117.8123	3	<.0001
Score	125.2476	3	<.0001
Wald	97.8685	3	<.0001

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > Chi Sq
Intercept	1	0.4280	0.4209	1.0339	0.3092
cat_swoe	1	1.7748	0.3753	22.3630	<.0001
percapita*Carbohydra	1	-0.00031	0.000080	14.7299	0.0001
cat_swoe*percapita_a	1	-0.0276	0.00314	77.3419	<.0001

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	70.9	Somers' D	0.417
Percent Discordant	29.1	Gamma	0.417
Percent Tied	0.0	Tau-a	0.154
Pairs	382976	c	0.709

Cardiovascular disease

```
proc logistic data=work.train_imputed_swoe_cardio;
    model cardio_bin(event='1') = Carbohydrate_Median
    cat_swoe percapita_all*Carbohydrate_Median percapita_all*cat_swoe;
run;
```

:

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_CARDIO
Response Variable	cardio_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	cardio_bin	Total Frequency
1	0	1088
2	1	352

Probability modeled is cardio_bin=1.

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	1603.709	1501.978
SC	1608.982	1526.340
-2 Log L	1601.709	1491.978

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > Chi Sq
Likelihood Ratio	109.7313	4	<.0001
Score	117.7624	4	<.0001
Wald	89.9035	4	<.0001

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > Chi Sq
Intercept	1	0.5718	0.4253	1.8081	0.1787
Carbohydrate_Median	1	0.00849	0.00290	8.5781	0.0034
cat_swoe	1	2.0089	0.3906	26.4543	<.0001
Carbohydr*percapita_	1	-0.00054	0.000108	26.2880	<.0001
cat_swoe*percapita_a	1	-0.0298	0.00351	71.7877	<.0001

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	69.8	Somers' D	0.396
Percent Discordant	30.2	Gamma	0.396
Percent Tied	0.0	Tau-a	0.146
Pairs	382976	c	0.698

Chronic obstructive pulmonary:

```
proc logistic data=work.train_imputed_swoe_pul;
    model pul_bin(event='1') = percapita_all
    percapita_all*Fat_Median percapita_all*Protein_Median percapita_all*cat_swoe;
run;
```

The LOGISTIC Procedure

Model Information	
Data Set	WORK.TRAIN_IMPUTED_SWOE_PUL
Response Variable	pul_bin
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	1440
Number of Observations Used	1440

Response Profile		
Ordered Value	pul_bin	Total Frequency
1	0	1072
2	1	368

Probability modeled is pul_bin=1.

Model Convergence Status
Convergence criterion (GCONV=1E-6) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	1638.867	1591.817
SC	1644.140	1618.179
-2 Log L	1636.867	1581.817

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > Chi Sq
Likelihood Ratio	55.0504	4	<.0001
Score	56.6080	4	<.0001
Wald	50.2602	4	<.0001

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > Chi Sq
Intercept	1	-1.3088	0.0776	284.7573	<.0001
percapita_all	1	0.1095	0.0192	32.5870	<.0001
percapita*Fat_Median	1	0.000060	0.000140	0.1807	0.6708
percapita*Protein_Me	1	-0.00067	0.000289	5.3567	0.0206
percapita_a*cat_swoe	1	0.0821	0.0163	25.2141	<.0001

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	63.9	Somers' D	0.281
Percent Discordant	35.9	Gamma	0.281
Percent Tied	0.2	Tau-a	0.107
Pairs	394496	c	0.640

As is clear from the results of each procedure, the concordant pairs greatly outnumber the tied and discordant pairs, indicating the models are representative of the data. Of note, the training data c statistic is .595 for diabetes, .709 for malignant neoplasms, .696 for cardiovascular disease, and .640 for chronic obstructive pulmonary disease. This would prove useful for measuring the model performance later.

F. Measuring Model Performance

I. Preparing the Validation Data

Prior to scoring the validation data, the validation data set must be adjusted for the new variables that were added to the training dataset. This is done to most accurately measure the model's ability to generalize to new data. In this case, only a smooth weight of evidence variable necessitates addition. The same procedure was conducted for each NCD.

Diabetes:

```
%global rho1;
proc sql noprint;
    select mean(diabetes_bin) into :rho1
    from capstone.diabetes_valid;
run;

proc means data=capstone.diabetes_valid sum nway noprint;
    class cat;
    var diabetes_bin;
    output out=work.diab_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/diab_brswoe.sas";

data _null_;
    file brswoe;
    set work.diab_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.train_imputed_swoe_diab;
    set capstone.diabetes_valid;
    %include brswoe /source2;
run;
```

Malignant neoplasms:

```
%global rho1;
proc sql noprint;
    select mean(neoplasms_bin) into :rho1
    from capstone.neoplasms_valid;
run;

proc means data=capstone.neoplasms_valid sum nway noprint;
    class cat;
    var neoplasms_bin;
    output out=work.neoplasm_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/neoplasms_brswoe.sas";

data _null_;
    file brswoe;
    set work.neoplasm_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;
```

Cardiovascular disease:

```
%global rho1;
proc sql noprint;
    select mean(cardio_bin) into :rho1
    from capstone.cardio_valid;
run;

proc means data=capstone.cardio_valid sum nway noprint;
    class cat;
    var cardio_bin;
    output out=work.neoplasm_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/cardio_brswoe.sas";

data _null_;
    file brswoe;
    set work.cardio_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.valid_imputed_swoe_cardio;
    set capstone.cardio_valid;
    %include brswoe /source2;
run;
```

Chronic obstructive pulmonary:

```
%global rho1;
proc sql noprint;
    select mean(pul_bin) into :rho1
    from capstone.pul_valid;
run;

proc means data=capstone.pul_valid sum nway noprint;
    class cat;
    var pul_bin;
    output out=work.pul_counts sum=events;
run;

filename brswoe "/folders/myshortcuts/myfolder/Capstone/brswoe/pul_brswoe.sas";

data _null_;
    file brswoe;
    set work.pul_counts end=last;
    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));
    if _n_=1 then put "select (cat);" ;
    put " when ('" cat +(-1)"') cat_swoe = " logit ";";
    if last then do;
    logit = log(&rho1/(1-&rho1));
    put " otherwise cat_swoe = " logit ";" / "end;";
    end;
run;

data work.valid_imputed_swoe_pul;
    set capstone.pul_valid;
    %include brswoe /source2;
run;
```

The cat_swoe variable was added to the validation dataset using the same methodology and the data was then prepared for subsequent scoring and analysis.

B. Measuring Model Performance

A receiver-operator curve (ROC) was utilized to illustrate the predictive power of the model. The receiver-operator curve is a "performance measurement for [the] classification problem at various threshold settings." The area under the curve/c statistic is the ability the model has to predict the class to which the target will belong (Narkhede, 2018). This method was selected because it is widely used and easily interpreted. The gains and lift charts could also depict the same information, but may necessitate a more extensive explanation to the audience. Below is the ROC curve code for diabetes:

```
%global selected;
%let selected= percapita_all cat_swoe percapita_all*cat_swoe;

%let pi1= 0.25;

ods select roccurve scorefitstat;
proc logistic data=work.train_imputed_swoe_diab;
    model diabetes_bin(event='1')=&selected;
    score data=work.train_imputed_swoe_diab out=work.scova
        priorevent=&pi1 outroc=work.roc fitstat;
run;

title1 "Statistics in the ROC Data Set";
proc print data=work.roc(obs=10);
    var _prob_ _sensit_ _1mspec_;
run;

data work.roc;
    set work.roc;
    cutoff=_PROB_;
    specif=1-_1MSPEC_;
    tp=&pi1*_SENSIT_;
    fn=&pi1*(1-_SENSIT_);
    tn=(1-&pi1)*specif;
    fp=(1-&pi1)*_1MSPEC_;
    depth=tp+fp;
    pospv=tp/depth;
    negpv=tn/(1-depth);
    acc=tp+tn;
    lift=pospv/&pi1;
    keep cutoff tn fp fn tp
        _SENSIT_ _1MSPEC_ specif depth
        pospv negpv acc lift;
run;
```

Malignant neoplasms:

```
%global selected;
%let selected= cat_swoe percapita_all*Carbohydrate_Median percapita_all*cat_swoe;

%let pi1= 0.25;

ods select roccurve scorefitstat;
proc logistic data=work.train_imputed_swoe_neo;
    model neoplasms_bin(event='1')= &selected;
    score data=work.train_imputed_swoe_neo out=work.scova
        priorevent=&pi1 outroc=work.roc fitstat;
run;

title1 "Statistics in the ROC Data Set";
proc print data=work.roc(obs=10);
    var _prob_ _sensit_ _1mspec_;
run;

data work.roc;
    set work.roc;
    cutoff=_PROB_;
    specif=1-_1MSPEC_;
    tp=&pi1*_SENSIT_;
    fn=&pi1*(1-_SENSIT_);
    tn=(1-&pi1)*specif;
    fp=(1-&pi1)*_1MSPEC_;
    depth=tp+fp;
    pospv=tp/depth;
    negpv=tn/(1-depth);
    acc=tp+tn;
    lift=pospv/&pi1;
    keep cutoff tn fp fn tp
        _SENSIT_ _1MSPEC_ specif depth
        pospv negpv acc lift;
run;
```

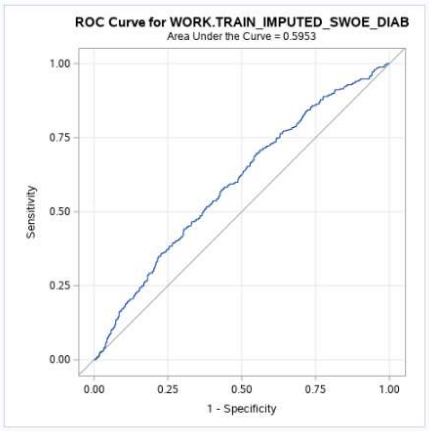
Cardiovascular disease:

```
%global selected;
%let selected= Carbohydrate_Median
    cat_swoe percapita_all*Carbohydrate_Median percapita_all*cat_swoe;

%let pi1= 0.25;

ods select roccurve scorefitstat;
proc logistic data=work.train_imputed_swoe_cardio;
    model cardio_bin(event='1')= &selected;
    score data=work.train_imputed_swoe_cardio out=work.scova
        priorevent=&pi1 outroc=work.roc fitstat;
run;

title1 "Statistics in the ROC Data Set";
proc print data=work.roc(obs=10);
    var _prob_ _sensit_ _1mspec_;
run;

data work.roc;
    set work.roc;
    cutoff=_PROB_;
    specif=1-_1MSPEC_;
    tp=&pi1*_SENSIT_;
    fn=&pi1*(1-_SENSIT_);
    tn=(1-&pi1)*specif;
    fp=(1-&pi1)*_1MSPEC_;
    depth=tp+fp;
    pospv=tp/depth;
    negpv=tn/(1-depth);
    acc=tp+tn;
    lift=pospv/&pi1;
    keep cutoff tn fp fn tp
        _SENSIT_ _1MSPEC_ specif depth
        pospv negpv acc lift;
run;
```

Chronic obstructive pulmonary:

```
%global selected;
%let selected= percapita_all
    percapita_all*Fat_Median percapita_all*Protein_Median percapita_all*cat_swoe;

%let pi1= 0.25;

ods select roccurve scorefitstat;
proc logistic data=work.train_imputed_swoe_pul;
    model pul_bin(event='1')= &selected;
    score data=work.train_imputed_swoe_pul out=work.scova
        priorevent=&pi1 outroc=work.roc fitstat;
run;

title1 "Statistics in the ROC Data Set";
proc print data=work.roc(obs=10);
    var _prob_ _sensit_ _1mspec_;
run;

data work.roc;
    set work.roc;
    cutoff=_PROB_;
    specif=1-_1MSPEC_;
    tp=&pi1*_SENSIT_;
    fn=&pi1*(1-_SENSIT_);
    tn=(1-&pi1)*specif;
    fp=(1-&pi1)*_1MSPEC_;
    depth=tp+fp;
    pospv=tp/depth;
    negpv=tn/(1-depth);
    acc=tp+tn;
    lift=pospv/&pi1;
    keep cutoff tn fp fn tp
        _SENSIT_ _1MSPEC_ specif depth
        pospv negpv acc lift;
run;
```

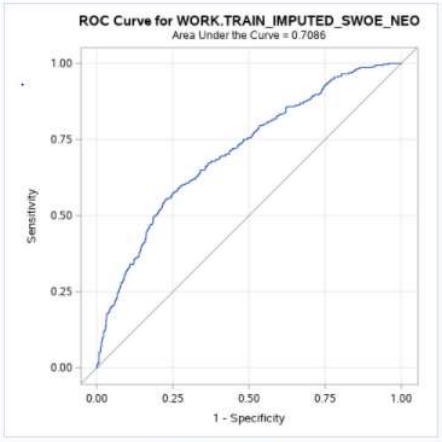
The above code depicts the fit statistics of the model on the new dataset and a receiver-operator curve to illustrates the sensitivity against one minus specificity. The curve is shown below for diabetes:



Fit Statistics for SCORE Data

Data Set	Total Frequency	Log Likelihood	Error Rate	AIC	AICC	BIC	SC	R-Square	Max-Rescaled R-Square	AUC	Brier Score
WORK.TRAIN_IMPUTED_SWOE_DIAB	1440	-788.0	0.2465	1583.924	1583.952	1605.014	1605.014	0.017909	0.026581	0.59531	0.181533

Malignant neoplasms:



Fit Statistics for SCORE Data

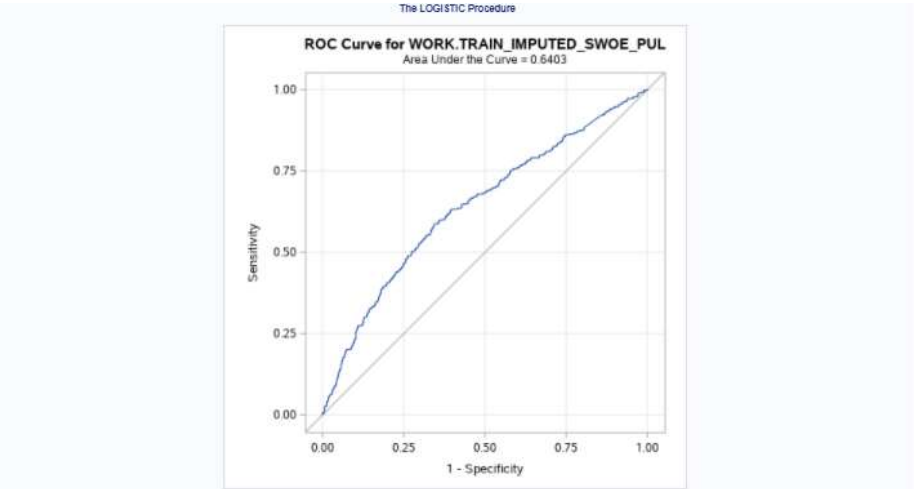
Data Set	Total Frequency	Log Likelihood	Error Rate	AIC	AICC	BIC	SC	R-Square	Max-Rescaled R-Square	AUC	Brier Score
WORK.TRAIN_IMPUTED_SWOE_NEO	1440	-742.1	0.2333	1492.114	1492.141	1513.203	1513.203	0.078571	0.117051	0.708829	0.167299

Cardiovascular disease:

Fit Statistics for SCORE Data

Data Set	Total Frequency	Log Likelihood	Error Rate	AIC	AICC	BIC	SC	R-Square	Max-Rescaled R-Square	AUC	Brier Score
WORK.TRAIN_IMPUTED_SWOE_CARDIO	1440	-746.1	0.2354	1502.196	1502.238	1528.556	1528.556	0.073384	0.109324	0.697823	0.169352

Chronic obstructive pulmonary:



Fit Statistics for SCORE Data

Data Set	Total Frequency	Log Likelihood	Error Rate	AIC	AICC	BIC	SC	R-Square	Max-Rescaled R-Square	AUC	Brier Score
WORK.TRAIN_IMPUTED_SWOE_PUL	1440	-791.0	0.2576	1592.043	1592.085	1618.405	1618.405	0.037515	0.055235	0.640265	0.182803

As seen, the area under the diabetes curve was .5953, which was extremely close to the training data set c statistic of .595. The area under the malignant neoplasms curve was .7086 which was also close to the training data set c statistic of .709. The cardiovascular disease area under the curve was .6978, close to the training c statistic of .696, and finally the area under the curve for chronic obstructive pulmonary disease was .6403, which was again similar to the training c statistic of .640. The similarities between the areas under the curve and the training data set c statics indicate the model generalized well to new data.

G. Data Summary and Implications

In the end the model that best fit the data was indeed the consumption rate, food category, and food category * consumption rate interaction for diabetes. The best fit for malignant neoplasms was the food category, consumption rate * carbohydrate interaction, and food category * consumption rate interaction. The best model for cardiovascular disease was carbohydrate, food category, the food consumption * carbohydrate interaction, and the food consumption * food category interaction. Finally the best model for chronic obstructive pulmonary disease involved food consumption rate, the food consumption * fat interaction, the food consumption * protein interaction, and the food consumption * food category interaction. Ultimately nutrition facts, food categories, and consumption rates are excellent predictors for NCDs. Each of the models provides an accurate method for predicting rates of NCDs based on certain criteria.

However, the data is limited by the confusion caused by using a smoothed weight of evidence to represent the food category variable. As a result of this fact, the data is difficult to interpret. Though, if the data was interpreted with this in mind, the model results can serve as an indicator for emerging nations to shift the focus of their markets to certain food groups that are not correlated with higher rates of NCDs.

A main limitation of this analysis is the cumulative effect of limitations within each dataset. The food consumption rates are based on the consumption as best estimated by The World Bank. Accurate consumption rates were not available due to government entities failing to disclose, as well as the unavailability of rates due to informal markets. This combined with the fact that the median nutrition rates for each food category may not be the most accurate representation of the food category is a major constraint.

In summary, there is clearly a significant relationship between food consumption rates, the categories of food being consumed, the nutrition facts of those foods, and the subsequent health outcomes. This relationship should be further analyzed with more data that analyzes specific food groups combined with finely tuned nutrition information. This would provide a better model that would assist the resourcing of developing nations. A more in-depth analysis of the specific areas within these countries that are consuming each food would be more useful for these purposes as well.

H. Resources

Alwan, A., Dr. (Ed.). (2011). *Global status report on noncommunicable diseases 2010* (p. 33, Rep.). Italy: World Health Organization. Retrieved May 8, 2020, from https://apps.who.int/iris/bitstream/handle/10665/44579/9789240686458_eng.pdf;jsessionid=B563EF98D7EC35CD759F3FC2408B9F85?sequence=1.

Choueiry, G. (2020). Understand Forward and Backward Stepwise Regression. Quantifying Health. https://quantifyinghealth.com/stepwise-selection/.

Croarkin, C., & Tobias, P. (Eds.). (2013, October 10). 1.3.5.10. Levene Test for Equality of Variances. Engineering Statistics Handbook. https://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm.

Data Hub. (2018). In *Population Figures by Country*. John Snow Labs. Retrieved July 10, 2020, from https://datahub.io/JohnSnowLabs/population-figures-by-country#readme.

John Snow Labs. (2018). Data Hub. In Population Figures by Country (pp. 1–1). https://datahub.io/JohnSnowLabs/population-figures-by-country#readme.

Laerd Statistics. (2020). Pearson Product-Moment Correlation. Statistical tutorials and software guides. https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php.

Narkhede, S. (2018, June 26). Understanding AUC - ROC Curve. Medium. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

SAS Studio: Help Center. (2016, June 21). Retrieved May 08, 2020, from https://support.sas.com/software/products/sas-studio/faq/SAS_whatis.htm.

SAS Institute. (2018). SAS University EditionVersion (3.8). S.l.

SAS Institute. (2020). *Predictive Modeling Using Logistic Regression (15.1)*. https://vle.sas.com/course/view.php?id=3472.

SAS Institute. (2020). *Statistics 1: Introduction to ANOVA, Regression, and Logistic Regression*. https://vle.sas.com/course/view.php?id=2113.

U.S. Department of Agriculture. (2020, April). Download FoodData Central Data. Retrieved May 08, 2020, from https://fdc.nal.usda.gov/download-datasets.html.

World Bank Group. (2010). Global Consumption Database. Retrieved May 08, 2020, from http://datatopics.worldbank.org/consumption/product/Rice.

World Health Organization. (2018, April 5). NCD Deaths by Cause and Sex - Data by Country. Retrieved May 08, 2020, from https://apps.who.int/gho/data/view.main.NCDDEATHCAUSESNUMBERv?lang=en.

Yap, B. W., & Sim, C. H. (2011). Comparisons of various types of normality tests. Journal of Statistical Computation and Simulation, 81(12), 2143–2155. https://doi.org/10.1080/00949655.2010.520163.