```
/*1. Define the object: problem statement. What am I trying to solve? */
/*Questions are listed to create a list of helpful foods for specific health goals. */

/*2. Collecting the data: dataset for foods with nutrient density */
/*DATA FROM: United States Department of Agriculture's Food Composition Database, recommended serving size*/

/*3. Scrub the data: remove major errors, duplicates, outliers, remove unwanted data points
bringing in structure to data, filling in major gaps */

/*4. Analyze the data: univariate/bivariate, time-series, regression
how to apply: descriptive analysis - what has already happened, diagnostic analysis - why
something has already happened, predictive analysis - to identify fiture trends, prescriptive analysis */

/*5. Share the insights - interpreting outcomes, and presenting them in digestible manner (reports, dashboards) */

/*Excel file is easier to import than CSV - so Excel was used to clean up columns before import */



/****************************** DATA MANIPULATION *********************/
/**********************************************************************/

/* Create file and backup file */
Data Amir.foodFile;
set Amir.foodSimp;
run;

Data Amir.foodFilebk;
set Amir.foodFile;
run;

/* Clean the dataset / properly label - though it is already clean */
/*Labels for columns */
Data Amir.foodFile;
set Amir.foodFile;
label   Category = 'General Food Category'
        Description = 'Description or Food preparation'
        DataBNum = 'Nutrient Data Bank Number'
        AlphaCar = 'Alpha Carotene mcg'
        BetaCaro = 'Beta Carotene mcg'
        BCryptoxanthin = 'Beta Cryptoxanthin mcg'
        Carbs = 'Carbohydrates g'
        Cholesterol = 'Cholesterol mg'
        Choline = 'Choline mg'
        Fiber = 'Fiber g'
        LuteinZeaxanthin = 'Lutein and Zeaxanthin mcg'
        Lycopene = 'Lycopene'
        Niacin = 'Niacin mg'
        Protein = 'Protein g'
        Retinol = 'Retinol mcg'
        Riboflavin = 'Riboflavin mg'
        Selenium = 'Selenium mcg'
        SugarTot = 'Sugar total g'
        Thiamin = 'Thiamin mg'
        Water = 'Water g'
        MonosatFat = 'Monosaturated fat g'
        PolysatFat = 'Poly saturated fat g'
        SatFat = 'Saturated fat g'
        TotLipid = 'Total lipid g'
        Calcium = 'Calcium mg'
        Copper = 'Copper mg'
        Iron = 'Iron mg'
        Magnesium = 'Magnesium mg'
        Phosphorus = 'Phosphorus mg'
        Potassium = 'Potassium mg'
        Sodium = 'Sodium mg'
        Zinc = 'Zinc mg'
        A = 'Vitamin A mcg'
        B12 = 'Vitamin B12 mcg'
        B6 = 'Vitamin B6 mg'
        C = 'Vitamin C mg'
        E = 'Vitamin E mg'
        K = 'Vitamin K mcg';
run;
```

```
/*File doesnt have any inconsistensies so we will check for duplicates and sort file*/
/*No duplicate refs found in drfoodfile*/
proc sort data = Amir.FoodFile out=Amir.dfFoodFile dupout=Amir.drFoodFile nodup;
by DataBNum;
run;


/*What foods are the highest in carbs? */
Data Amir.hcarbs;
set Amir.foodfile;
if Carbs > 30; /*30 grams*/
keep Category Description DataBNum Carbs SugarTot Fiber;
run;

/*which method of cooking rice is lowest in carbs, and low in sugar? Sort list in desc by carbs */
Data Amir.ricelcarb;
set Amir.foodfile;
where (Category = 'Rice') and (Carbs < 30) and (SugarTot < 8);
keep Category Description DataBNum Carbs SugarTot;
run;

proc sort data = Amir.ricelcarb;
BY DESCENDING carbs;
run;

/*What foods are the highest in saturated fats? */
Data Amir.sffood;
set Amir.foodfile;
if (satFat > 5);
keep Category Description DataBNum satFat polysatfat monosatfat;
run;

/*High in satfat and is not milk, cheese by itself, or ice cream - sort desc*/
Data Amir.sffoodnd;
set Amir.foodfile;
where (satFat > 5) and (category not in('Milk', 'Buttermilk', 'Cheese', 'Cream')
  and (Description not like '%cheese%')
  and (Description not like '%cream%')
  and (Description not like '%Ice cream%'));
keep Category Description DataBNum satFat polysatfat monosatfat;
run;

proc sort data = Amir.sffoodnd;
BY DESCENDING satfat;
run;

/*What foods are highest in protein? */
Data Amir.hpfood;
set Amir.foodfile;
where protein > 30;
keep Category Description DataBNum protein;
run;

proc sort data = Amir.hpfood;
BY DESCENDING protein;
run;

/*What foods have the best electrolyte content? Calcium. Magnesium. Phosphorus. Potassium. Sodium. - but based on Sodium, Pot
/*above 10% - 230 for sod, 150 - pot,  */
Data Amir.sodFood Amir.potFood Amir.calFood Amir.oeFood;
set Amir.foodfile;
if Sodium < 1500 and Sodium > 300 then output Amir.sodFood; /*good range of sodium*/
else if Potassium > 300 and Potassium < 1500 then output Amir.potFood; /*good range of K*/
else if Calcium >  150 and Calcium < 1000 then output Amir.calFood; /*good range of calcium*/
else output Amir.oeFood;
keep Category Description DataBNum calcium magnesium potassium phosphorus sodium;
run;

/*take the top 25 of each and place in seperate files*/
proc sql outobs=25;
   create table sodfood25
     as select * from Amir.sodFood
     order by sodium desc;
quit;

proc sql outobs=25;
   create table potfood25
```

```
     as select * from Amir.potfood
     order by potassium desc;
quit;

proc sql outobs=25;
   create table calfood25
     as select * from Amir.calfood
     order by calcium desc; /*sort highest to lowest calcium value then take top 25*/
quit;


/*Which foods have the most b vitamins and protein content, sort by most protein? (beneficial for energy production) */
Data Amir.BPfoods;
set Amir.foodfile;
where b12 > 1.2 and niacin > 7 and thiamin > 0.3 and b6 > 0.2 and riboflavin > 0.2 and protein > 15;
keep Category Description DataBNum thiamin riboflavin niacin b12 b6 protein;
run;

proc sort data=Amir.bpfoods;
by descending protein;
run;

/*Low carb and low fat foods?*/
Data Amir.lclffood;
set Amir.foodfile;
where carbs < 10 and totalFat < 20 and (category not in ('Infant formula'));
keep Category Description DataBNum carbs totalFat;
run;

/*What are the top 100 low carb options with high fiber? */
Data Amir.lchffood;
set Amir.foodfile;
where carbs < 20 and fiber > 3;
keep Category Description DataBNum carbs fiber;
run;

/*sort by asc to put lowest at the top*/
proc sql outobs=100;
   create table Amir.lchffood100
     as select * from Amir.lchffood
     order by carbs asc;
quit;


/*Which foods are best to increase testosterone? D, Cholesterol, selenium, B vitamins, protein */
Data Amir.htfood;
set Amir.foodfile;
where b12 > 1.2 and protein > 15 and selenium > 25 and zinc > 5 and cholesterol < 100 ;
keep Category Description DataBNum Cholesterol selenium B12 Zinc protein;
run;

/*Separate macros: carbs, fats, proteins to top 200 sources of each. */
/*THEN merge all 3 tables - this will be the healthy foods table */

/*add total fat column */
Data Amir.foodfile;
set Amir.foodfile;
totalfat = polysatfat + monosatfat + satfat;
label totalfat = 'Total fats g';
run;

/*separate the files with highest protein fat and a healthy range of carbs */
Data Amir.hcfood Amir.hpfood Amir.hffood Amir.ofood;
set Amir.foodfile;
if carbs > 30 and carbs < 60 then output Amir.hcfood; /*range of carbs for total intake*/
else if protein > 20 then output Amir.hpfood;
else if totalfat > 20 then output Amir.hffood;
else output Amir.ofood;
keep Category Description DataBNum totalfat protein carbs;
run;

/*sort each protein fat carb table from greatest to least with 200 obs*/
proc sql outobs=200;
   create table Amir.hcfood200
     as select * from Amir.hcfood
     order by carbs desc;
quit;
```

```
proc sql outobs=200;
   create table Amir.hffood200
     as select * from Amir.hffood
     order by totalfat desc;
quit;

proc sql outobs=200;
   create table Amir.hpfood200
     as select * from Amir.hpfood
     order by protein desc;
quit;

/*Merge the tables: hffood200, hpfood200, hcfood200 */
/*sort the data first */
proc sort data = Amir.hffood200;
by DataBNum;
run;

proc sort data = Amir.hpfood200;
by DataBNum;
run;

proc sort data = Amir.hcfood200;
by DataBNum;
run;

Data Amir.mergehfcp;
merge Amir.hffood200 Amir.hcfood200 Amir.hpfood200;
by DataBNum;
run;

/* What are good low carb and high protein foods? */
/*Demonstrate the process of joining */
Data Amir.lcfood;
set Amir.foodfile;
where carbs < 10;
keep Category Description DataBNum carbs;
run;

Data Amir.hpfood30;
set Amir.foodfile;
where protein > 30;
keep Category Description DataBNum protein;
run;

/*based on procedure (good practice) ensure no duplicates then join*/
proc sort data=Amir.lcfood out=Amir.dflcfood nodup;
by dataBNum;
run;

proc sort data=Amir.hpfood30 out=Amir.dfhpfood30 nodup;
by dataBNum;
run;

/*inner join high protein 30g low carb < 15g*/
proc sql;
create table Amir.lchp_ij as
select dfhpfood30.DataBNum, dfhpfood30.category, dfhpfood30.description, *
from Amir.dflcfood inner join Amir.dfhpfood30
on dflcfood.dataBNum = dfhpfood30.dataBNum; quit;

/*right join high protein 30g low carb < 15g*/
proc sql;
create table Amir.lchp_rj as
select dfhpfood30.DataBNum, dfhpfood30.category, dfhpfood30.description, *
from Amir.dflcfood right join Amir.dfhpfood30
on dflcfood.dataBNum = dfhpfood30.dataBNum; quit;

/*left join*/
proc sql;
create table Amir.lchp_lj as
select dflcfood.DataBNum, dflcfood.category, dflcfood.description, *
from Amir.dflcfood left join Amir.dfhpfood30
on dflcfood.dataBNum = dfhpfood30.dataBNum; quit;

/*full join of low fat and high protein*/
```

```
proc sql;
create table Amir.lfhp_fj as
select coalesce (dflcfood.dataBNum, dfhpfood30.dataBNum) as DataBNum,
        coalesce (dflcfood.description, dfhpfood30.description) as Description,
        coalesce (dflcfood.category, dfhpfood30.category) as category, *
from Amir.dflcfood full join Amir.dfhpfood30
on dflcfood.dataBNum = dfhpfood30.dataBNum; quit;



/************************DATA VISULIZATION*********************/
/**************************************************************/

/*Create category macro*/
%let vcat = category;
%let top10 = 10;
%let top15 = 15;
%let top20 = 20;
%let top25 = 25;

/*Check frequency to see the distribution of categories: What category of foods are the highest in carbs? */
/*check the frequency of categories*/
/** proc freq data=Amir.hcfood ORDER=FREQ noprint;
tables &vcat / out=Amir.hcfoodfreq;
run; *//
/*create other category*/
/** data Amir.hcother;
set Amir.hcfoodfreq;
label topCat = 'Top 10 High Carb Categories & Other';
topCat = &vcat;
if _n_ > &top10 then
topCat='Other';
run; */
/* proc freq data= Amir.hcother ORDER=data; /* order by data and use WEIGHT statement for count */
/* tables TopCat / plots=FreqPlot(scale=percent);
 weight Count;
run; */

/*GRAPH 1: What category of foods are the highest in carbs? */

/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.hcfood noprint nway ;
var carbs;
class &vcat;
output out=Amir.hcmean mean=carbmean;
run;

/* Sort the data for the highest carb content first */
proc sort data=Amir.hcmean;
by descending carbmean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.tophcmean;
set Amir.hcmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
Title'Highest Carbohydrate categories';
proc gchart data=Amir.tophcmean;
hbar &vcat / discrete type=sum sumvar=carbmean nostats;
run;
quit;

/*GRAPH 2: What foods are highest in protein? */
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.hpfood noprint nway ;
var protein;
class &vcat;
output out=Amir.hpmean mean=proteinmean;
run;

/* Sort the data for the highest carb content first */
proc sort data=Amir.hpmean;
by descending proteinmean;
run;
```

```
/* Seperate table for the top 25 data to be displayed*/
data Amir.tophpmean;
set Amir.hpmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'Highest Protein categories';
proc gchart data=Amir.tophpmean;
hbar &vcat / discrete type=sum sumvar=proteinmean nostats;
run;
quit;



/*GRAPH 3: Category of foods highest in saturated fats? */
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.sffood noprint nway ;
var satFat;
class &vcat;
output out=Amir.sfmean mean=satfatmean;
run;

/* Sort the data for the highest carb content first */
proc sort data=Amir.sfmean;
by descending satfatmean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.topsfmean;
set Amir.sfmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'Highest Protein categories';
proc gchart data=Amir.topsfmean;
hbar &vcat / discrete type=sum sumvar=satfatmean nostats;
run;
quit;


/*GRAPH 4: What are the top 25 low carb categories with high fiber? */
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.lchffood noprint nway;
var carbs fiber;
class &vcat;
output out=Amir.lchfmean Mean(carbs)= Mean(fiber)= / autoname; /*TWO means in one output file*/
run;

/* Sort the data for the lowest carb content first */
proc sort data=Amir.lchfmean;
by descending carbs_mean fiber_mean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.toplchfmean;
set Amir.lchfmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'Low Carbohydrate & High fiber categories';
proc gchart data=Amir.toplchfmean;
hbar &vcat / discrete type=sum sumvar=carbs_mean nostats;
run;
quit;

/*What category of foods are highest in electrolytes? */

/* PIE CHARTS */
/*GRAPH 5: Sodium*/
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.sodfood noprint nway ;
var sodium;
class &vcat;
```

```
output out=Amir.sodmean mean=sodmean;
run;

/* Sort the data for the highest sodium content first */
proc sort data=Amir.sodmean;
by descending sodmean;
run;

/* Seperate table for the top 15 data to be displayed*/
data Amir.topsodmean;
set Amir.sodmean;
if _n_ <= &top15;
run;

/* pie chart for sodium */
title'High Sodium categories';
proc gchart data = Amir.topsodmean;
pie &vcat / discrete percent = inside sumvar = sodmean;
format sodmean 7.0;
run; quit;

/*GRAPH 6: Calcium */
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.calfood noprint nway ;
var calcium;
class &vcat;
output out=Amir.calmean mean=calmean;
run;

/* Sort the data for the highest calcium content first */
proc sort data=Amir.calmean;
by descending calmean;
run;

/* Separate table for the top 10 data to be displayed*/
data Amir.topcalmean;
set Amir.calmean;
if _n_ <= &top10;
run;

/* pie chart for calcium */
title'High Calcium categories';
proc gchart data = Amir.topcalmean;
pie &vcat / discrete percent = inside sumvar = calmean;
format calmean 7.0;
run; quit;

/*GRAPH 7: Potassium*/
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.potfood noprint nway ;
var potassium;
class &vcat;
output out=Amir.potmean mean=potmean;
run;

/* Sort the data for the highest calcium content first */
proc sort data=Amir.potmean;
by descending potmean;
run;

/* Separate table for the top 20 data to be displayed*/
data Amir.toppotmean;
set Amir.potmean;
if _n_ <= &top20;
run;

/* pie chart for potassium */
title'High Potassium categories';
proc gchart data = Amir.toppotmean;
pie &vcat / discrete percent = inside sumvar = potmean;
format potmean 7.0;
run; quit;

/*GRAPH 8: What category of foods is best to increase testosterone? */
/*b12 > 1.2 and protein > 15 and selenium > 25 and zinc > 5 and cholesterol < 100 */
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.htfood noprint nway ;
```

```
var b12 protein selenium zinc cholesterol;
class &vcat;
output out=Amir.htmean Mean(b12)= Mean(protein)= Mean(selenium)= Mean(zinc)= Mean(cholesterol)= / autoname;
run;

/* Sort the data for the highest content first */
proc sort data=Amir.htmean;
by descending selenium_mean;
run;

/* Seperate table for the top 10 data to be displayed*/
data Amir.tophtmean;
set Amir.htmean;
if _n_ <= &top10;
run;

/* horizontal chart for high test boosting foods */
title'High Testosterone boosting categories (by Selenium)';
proc gchart data = Amir.tophtmean;
hbar &vcat / discrete inside = percent sumvar = selenium_mean;
format selenium_mean 7.0;
run; quit;

/*GRAPH 9: What category of protein rich foods has the most b-vitamins? */
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.bpfoods noprint nway ;
var protein niacin riboflavin thiamin b6 b12 ;
class &vcat;
output out=Amir.bpmean mean(protein)= mean(niacin)= mean(riboflavin)= mean(thiamin)= mean(b6)= mean(b12)=  / autoname;
run;

/* Sort the data for the b vitamin and protein content first */
proc sort data=Amir.bpmean;
by descending protein_mean;
run;

/* Seperate table for the top 15 data to be displayed*/
data Amir.topbpmean;
set Amir.bpmean;
if _n_ <= &top15;
run;

/*Create graph to display the results*/
title'High Protein & B-vitamin rich categories';
proc gchart data=Amir.topbpmean;
hbar &vcat / discrete type=sum sumvar=protein_mean nostats;
run;
quit;

/*GRAPH 10: What category of foods are high protein and low carbs? */
/*Can use the lchp_ij inner join table*/

/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.LCHP_IJ noprint nway ;
var protein carbs;
class &vcat;
output out=Amir.lchpmean mean(carbs)= mean(protein) = / autoname;
run;

/* Sort the data for the highest carb content first */
proc sort data=Amir.lchpmean;
by descending protein_mean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.toplchpmean;
set Amir.lchpmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'High Protein and Low Carb categories';
proc gchart data=Amir.toplchpmean;
hbar &vcat / discrete type=sum sumvar=protein_mean nostats;
run;
quit;
```

```
/*GRAPH 11: Low carb and low fat foods */
/*check the frequency of categories*/
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.lclffood noprint nway ;
var carbs totalFat;
class &vcat;
output out=Amir.lclfmean mean(carbs)= mean(totalFat)= / autoname;
run;

/* Sort the data for the highest carb content first */
proc sort data=Amir.lclfmean;
by carbs_mean totalfat_mean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.toplclfmean;
set Amir.lclfmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'Low Carb & Low Fat categories';
proc gchart data=Amir.toplclfmean;
hbar &vcat / discrete type=sum sumvar=totalFat_mean nostats;
run;
quit;

/*Graph 12: Highest protein, fat and a healthy range of carbs?*/
/*Healthy: fat carb protein*/
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.mergehfcp noprint nway ;
var totalfat carbs protein;
class &vcat;
output out=Amir.fcpmean mean(totalfat)= mean(carbs)= mean(protein)= / autoname;
run;

/* Sort the data for the highest protein content first */
proc sort data=Amir.fcpmean;
by descending protein_mean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.topfcpmean;
set Amir.fcpmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'Healthy Macronutrient range categories (Protein based)';
proc gchart data=Amir.topfcpmean;
hbar &vcat / discrete type=sum sumvar=protein_mean nostats;
run;
quit;


/*Graph 13: Low carbs High Fiber*/
/* Calculate the mean for the categories, high frequency foods will be averaged */
proc means data=Amir.lchffood noprint nway ;
var carbs fiber;
class &vcat;
output out=Amir.lchfmean mean(carbs)= mean(fiber)= / autoname;
run;

/* Sort the data for the lowest carbs content first */
proc sort data=Amir.lchfmean;
by carbs_mean;
run;

/* Seperate table for the top 25 data to be displayed*/
data Amir.toplchfmean;
set Amir.lchfmean;
if _n_ <= &top25;
run;

/*Create graph to display the results*/
title'Low carbs, High Fiber categories';
proc gchart data=Amir.toplchfmean;
```

```
hbar &vcat / discrete type=sum sumvar=carbs_mean nostats;
run;
quit;

/****************** STATISTICAL ANALYSIS *************************/
/*****************************************************************/

/*Mean Median Std Dev of the top 200 high protein, fat and healthy carbs*/
proc means data = Amir.hcfood200 order=freq mean median stddev;
var carbs;
class category;
run;

proc means data = Amir.hpfood200 order=freq mean median stddev;
var protein;
class category;
run;

proc means data = Amir.hffood200 order=freq mean median stddev;
var totalfat;
class category;
run;

/*Univariate of top 600 foods based on protein fat and carbs*/
proc univariate data=Amir.mergehfcp;
var carbs protein totalFat;
run;

/*Simple random sampling*/
proc surveyselect data = Amir.foodfile method=srs n=1000 out=Amir.foodSample;
run;

/*Paired sample test */
/*H0: There is no significant mean difference between 2 variables from 0 */
/*H1: There is a significant mean difference between 2 variables from 0 */
proc ttest data = Amir.foodsample;
paired carbs*sugartot;
run;
/*PR: <.0001 (less than 0.05)- therefore this is contributing highly and the null hypothesis is rejected */

/*Correlation between sugar and carbs*/
proc corr data=Amir.foodsample;
var carbs sugartot;
run;
/*Correlation value: 0.71841 is a strong positive correlation between sugar and carbs */

/*Correlation values for macros which correlate with electrolytes */
proc corr data=Amir.foodsample;
var protein carbs totalFat sodium potassium calcium;
run;

/*Sodium-Protein: ftest (0.41090) - weak pos correlation*/
/*Linear regression Model */
proc reg data = Amir.foodsample;
    model protein = sodium;
run; quit;
/*ftest: <.0001 - contributing highly */

/*Calcium-carbs: ftest (0.17877) - weak pos correlation*/
/*Linear regression Model */
proc reg data = Amir.foodsample;
    model carbs = calcium:
```