FOOD EXCHANGE ANALYSIS

SALAKI REYNALDO JOSHUA

INTRODUCTION

The purpose of this study is to analyze Food Data (Ingredients) to get Food Exchange Values by using Korean Food Data as a case study.

RELATED WORK

IS THERE ANY RELATED WORK?

- 2003 : A Basic Study of Food Exchange Database Construction and Search System (ENECC/E-Food Exchange) Based on Internet (Sun Myung Hong)
- 2004 : Improvements in e-Food Exchange of commonly used Foods and Search System (ENECC/E-Food Exchange) based on Internet (Sun Myung Hong)
- 2005 : Web Expert System for Nutrition Counseling and Menu Management (Sun Myung Hong)
- 2010 : Korean Food Exchange Lists for Diabetes: Revised 2010 (Ju, Dal Lae, et.al)

2003

대한영양사협회 학술지 9(2) : 159~171, 2003
Journal of the Korean Dietetic Association

인터넷 기반의 식품 교환량 데이터베이스 구축과 검색 시스템 (ENECC/E-Food Exchange)에 관한 기초 연구

홍순명[†] · 조희선 · 김곤*
울산대학교 식품영양전공 · 울산대학교 컴퓨터정보통신공학전공*

A Basic Study of Food Exchange Database Construction and Search System (ENECC/E-Food Exchange) Based on Internet

Hong Soon-Myung[†] · Cho Hee-Sun · Kim Gon*
Dept. of Food & Nurition, University of Ulsan
*Dept. of Computer & Information Communication, University of Ulsan**

ABSTRACT

The food composition tables are frequently used to health and nutrition practices. But it is difficult to find out food exchange lists with food exchange groups in the food composition table. Over 2500 items and many kinds of nutrients are in the food composition table. But now food exchange lists are used a few foods. The internet demands the users needs for obtaining more food exchange lists and nutrient information from food composition. This basic study is to solve the users need and the supply more efficient and effective manipulation system for e-food exchange database construction and search system:ENECC/e-food exchnage(E-Nutrition Education and Couseling Center/e-food exchange). This paper introduces the food exchange database construction and search system(ENECC/e-food exchange) using the formula which calculates the food exchange quantity of 6 food exchange groups and added one extra groups(alcohol) based on the internet. The ENECC/e-food exchange database is basically based on the 6th food composition table(2001) of the National Rural Living Science Institution in Rural Development Administration, Korea.

The e-food exchange database are consisted of 2,261 foods in 6 basic food groups and one extra groups by using ENECC calculating formula. Also, the e-food exchange database has the proximate composition, mineral

접수일 : 2003년 3월 19일, 채택일 : 2003년 4월 19일
[†]Corresponding author : Hong, Soonmyung, Department of food and Nutrition, College of Human Ecology, University of Ulsan, San 29, Moogeo 2-dong, Nam-ku, Ulsan, 680-749, Korea
Tel : 052)259-2374, Fax : 052)259-1699, E-mail : smhong@mail.ulsan.ac.kr

2004

대한영양사협회 학술지 10(2):129-142, 2004
Journal of the Korean Dietetic Association

인터넷 기반의 일상식품의 e-식품교환량 (ENECC/E-Food Exchange)과 검색 시스템 개선

홍순명[†] · 조희선 · 김곤*
울산대학교 식품영양전공 · 울산대학교 컴퓨터정보통신공학전공*

Improvements in e-Food Exchange of commonly used Foods and Search System (ENECC/E-Food Exchange) based on Internet

Soon-Myung Hong[†] · Hee-Sun Cho · Gon Kim*
Dept. of Food & Nurition, University of Ulsan
*Dept. of Computer & Information Communication, University of Ulsan**

ABSTRACT

The food exchanges are frequently used to nutrition education and counseling for diabetes, weight control, hyperlipidema and etc. But it is difficult to find out food exchange lists with food exchange groups in the food composition table. This study was conducted to select e-food exchange of commonly used foods(456 foods) and improve search system based on internet. Also, the e-food exchange database was developed as having the proximate composition, mineral and vitamin content such as energy, moisture, protein, fat, carbohydrate, ash, calcium, phosphorus, iron, sodium, potassium, retinol equivalent, retinol, β-carotene, thiamin, riboflavin, niacin, ascorbic acid, refuse per 100g of each food. The e-food exchange database developed is basically based on the 6th food composition table(2001) of the National Rural Living Science Institution in Rural Development Administration, Korea. The 456 commonly used foods or 2,262, all foods can be searched easily in a new system. A specific food of e-food exchange can be searched by a given set of food groups or food name and can give information about food one exchange lists, weight and nutrient value per 100g of each food user-friendly on internet. It can be used to make a nutritionally balanced meal plan, nutrition education and counseling.

Key Words : e-food exchange database, search system of e-food exchange based on internet, commoly used e-food exchange

서 론

최근에 사회 환경이 급변하면서 새로운 편의식품의 개발, 식품의 유용성 증가, 생활방식, 신체활동, 평균 체위 등 여러 면에서 변화가 있었다. 성인병이라고 불리던 질병들이 생활습관병이라고 일컬어지면서 식사에 의한 영양섭취에 관심이 높아지게 되어 건강에 대한 명확한 기준을 요구하게 되었으며, 1986년 WHO의 건강증진을 위한 Ottawa협정 이후 각 나라들이 건강 증진법을 제정하게 되었으며 우리나라도 1995년 국민건강 증진법을 만들게 되었다[1]. 한편, 21세기 정보화시대를 맞이하여 영양치료를 위한 영양교육 및 상담의 효과가 나타나면서 영양상태판정 및 식단작성 등의 정보처리에 컴퓨터의 중요성이 대두되고 있다[2-4]. 컴퓨터는 영양관리, 업무관리와 영양관련 연구에 있어

접수일 : 2004년 1월 12일, 채택일 : 2004년 2월 12일
†Corresponding author : Soon-Myung Hong, Department of food and Nutrition, College of Human Ecology, University of Ulsan, san 29, Moogeo 2-dong, Nam-ku, Ulsan, 680-749, Korea
Tel : 052) 259-2374, Fax : 052)259-1699,
E-mail : smhong@mail.ulsan.ac.kr

2005

J Community Nutrition 7(2) : 107-113, 2005 □ Original Article □

Web Expert System for Nutrition Counseling and Menu Management*

Soon-Myung Hong,[1)†] Gon Kim[2)]
Department of Food & Nutrition,[1)] School of Computer & Information Technology,[2)] University of Ulsan, Ulsan, Korea

ABSTRACT

This study was conducted to develop a web expert system for nutrition counseling and menu management. This program manipulates a food, dish and menu and search database that has been developed. Clients can select a recommended general and therapeutic menu using this system. The web expert system can analyze nutrients in menus and compare nutrient contents of menus with Korean Recommended Dietary Allowances. It can access the food, dish and menu database. The expert menu database can insert, store and generate the synthetic information of age, sex, and therapeutic purpose of disease. With investigation and analysis of the client's needs, the menu planning program on the internet has been continuously developed. This system consists of the database that reaches to the food composition, the dishes and the menu. Clients can search food composition and conditional food based on nutrient name and amounts. This system is able to draw up the food with its order in dish. The menu planning can be organized and nutrients analysis can be compared with Korea Recommended Allowance. This system is able to read the nutrient composition of the each food, the dish and the menu. The results of analysis is presented quickly and accurately. Therefore it can be used by not only usual people but also dietitians and nutritionists who take charge of making a menu and experts in the field of food and nutrition. It is expected that the web expert system can be useful of nutrition education, nutrition counseling and expert menu management. (*J Community Nutrition* 7(2) : 107-113, 2005)

KEY WORDS : web expert system · nutrition counseling · menu management.

Introduction

Some off-line programs on menu planning, nutrition analysis and nutrition education counseling have been developed and used(Han, Rhee 1993b ; Han 1997a ; Han 1997b ; Hong 1989 ; Hong 1996 ; Kang et al. 1998 ; Kang et al. 1999 ; Kolasa, Miller 1996 ; Peter et al. 1998). Similar programs on the internet are insufficient(Choi 2000). Therefore the system based on the internet that has a friendlly user interface and accepts the needs of clients is required as soon as possible(Hong, Hwang 2001).

In the case of the USA, as a program based on internet, the food composition table from the USDA is commonly used. Cyberdiet services food nutrition information and menu and some web sites have meal analysis functions, but those are insufficient(Lee, Nieman 2003). Nutrition related programs based on internet were developed such as the analysis of a food intake and nutrition screening system(Han 2000), a web-based internet program for nutritional counseling and diet management of patient with diabetes mellitus(Han, Jeong 2004), food exchange database construction and search system(ENECC/E-Food Exchange) based on internet(Hong et al. 2003), e-food exchange database construction of commonly used foods and search system(Hong et al. 2004) and menu planning and recommended menu search system(MenuGen) of the National Rural Living Science Institution in Rural Development Administration(Hong et al. 2004 ; National Rural Living Institute). In addition, there are the statuses of eating habits, analysis of eating habits, obesity, calorie expenditure and nutrient analysis of menus in the internet program of the nutrition computing(Hong, Kim 2004). But the developed programs are insufficient in menu planning, menu analysis and management of recommended menu. Users

*This work was supported by 2004 Research Fund of University of Ulsan.
† **Corresponding author** : Soon-Myung Hong, Department of Food & Nutrition, University of Ulsan, San 29 Muger 2-dong, Nam-gu, Ulsan 680-749, Korea
Tel : (052) 259-2374, Fax : (052) 259-1699
E-mail : smhong@ulsan.ac.kr

2010

한국영양학회지(*Korean J Nutr*) 2011 [illegible]4(6): [illegible]7 ~ 591 Report
DOI 10.4163/kjn.2011.44.6.577

2010 당뇨병 환자를 위한 식품교환표 개정

주달래[1] · 장학철[2§] · 조영연[3] · 조재원[3] · 유혜숙[4]
최경숙[5] · 우미혜[6] · 손정민[7] · 박유경[8,9] · 조여원[8,9§]
서울대학교병원 급식영양과,[1] 분당서울대학교병원 내과,[2] 삼성서울병원 영양팀,[3] 중앙대학교병원 영양팀,[4] 대진대학교 식품영양학과,[5] 경희대학교병원 영양팀,[6] 원광대학교 식품영양학과,[7] 경희대학교 동서의학대학원,[8] 경희대학교 임상영양연구소[9]

Korean Food Exchange Lists for Diabetes: Revised 2010

Ju, Dal Lae[1] · Jang, Hak Chul[2§] · Cho, Young Yun[3] · Cho, Jae Won[3] · Yoo, Hye Sook[4]
Choi, Kyung Suk[5] · Woo, Mi Hye[6] · Sohn, Cheong Min[7] · Park, Yoo Kyoung[8,9] · Choue, Ryowon[8,9§]
[1]*Department of Food Service and Nutrition, Seoul National University Hospital, Seoul 151-741, Korea*
[2]*Department of Internal Medicine, Seoul National University Bundang Hospital, Seoul National University College of Medicine, Seongnam 463-707, Korea*
[3]*Department of Dietetics, Samsung Medical Center, Seoul 135-710, Korea*
[4]*Department of Food Service and Nutrition, Chung-Ang University Hospital, Seoul 156-755, Korea*
[5]*Department of Food Science and Nutrition, Daejin University, Pocheon 487-711, Korea*
[6]*Department of Nutrition, Kyung Hee University Medical Center, Seoul 130-702, Korea*
[7]*Major in Food and Nutrition, Wonkwang University, Jeonbuk 570-749, Korea*
[8]*Department of Medical Nutrition, Kyung Hee University, Yongin 446-701, Korea*
[9]*Research Institute of Medical Nutrition, Kyung Hee University, Seoul 130-701, Korea*

ABSTRACT

A food exchange system for diabetes is a useful tool for meal planning and nutritional education. The first edition of the Korean food exchange lists was developed in 1988 and the second edition was revised in 1995. With recent changes in the food marketplace and eating patterns of Koreans, the third edition of food exchange lists was revised in 2010 by the Korean Diabetes Association, the Korean Nutrition Society, the Korean Society of Community Nutrition, the Korean Dietetic Association and the Korean Association of Diabetes Dietetic Educators through a joint research effort. The third edition is based on nutritional recommendations for people with diabetes and focuses on adding foods to implement personalized nutrition therapy considering individual preferences in diverse dietary environment. Foods were selected based on scientific evidence including the 2007 Korea National Health and Nutrition Examination Survey data analysis and survey responses from 53 diabetes dietetic educators. While a few foods were deleted, a number of foods were added, with 313 food items in food group lists and 339 food items in the appendix. Consistent with previous editions, the third edition of the food exchange lists included six food categories (grains, meat, vegetables, fats and oils, milk, and fruits). The milk group was subdivided into whole milk group and low fat milk. The standard nutrient content in one exchange from each food group was almost the same as the previous edition. Korea Food & Drug Administration's FANTASY (Food And Nutrient daTA SYstem) database was used to obtain nutrient values for each individual food and to determine the serving size most appropriate for matching reference nutrients values by each food group. The revised food exchange lists were subjected to a public hearing by experts. The third edition of the food exchange lists will be a helpful tool for educating people with diabetes to select the kinds and amounts of foods for glycemic control, which will eventually lead to preventing complications while maintaining the pleasure of eating. (*Korean J Nutr* 2011: 44(6): 577 ~ 591)

KEY WORDS: food exchange lists, diabetes mellitus, Korean, meal plan, serving size.

*This document was written to introduce the 3rd edition of Korean food exchange lists, which was revised in 2010 by the Korean Diabetes Association, the Korean Nutrition Society, the Korean Society of Community Nutrition, the Korean Dietetic Association and the Korean Association of Diabetes Dietetic Educators through a joint resarch effort. The revision of Korean Food Exchange Lists was supported by the Korean Diabetes Association.
Simultaneous publication: This article is being simultaneously published in 2011 in the Korean Journal of Nutrition and the Journal of Korean Diabetes.
접수일: 2011년 11월 20일 / 채택일: 2011년 11월 28일
§To whom correspondence should be addressed.
E-mail: 장학철(janghak@snu.ac.kr), 조여원(rwcho@khu.ac.kr)

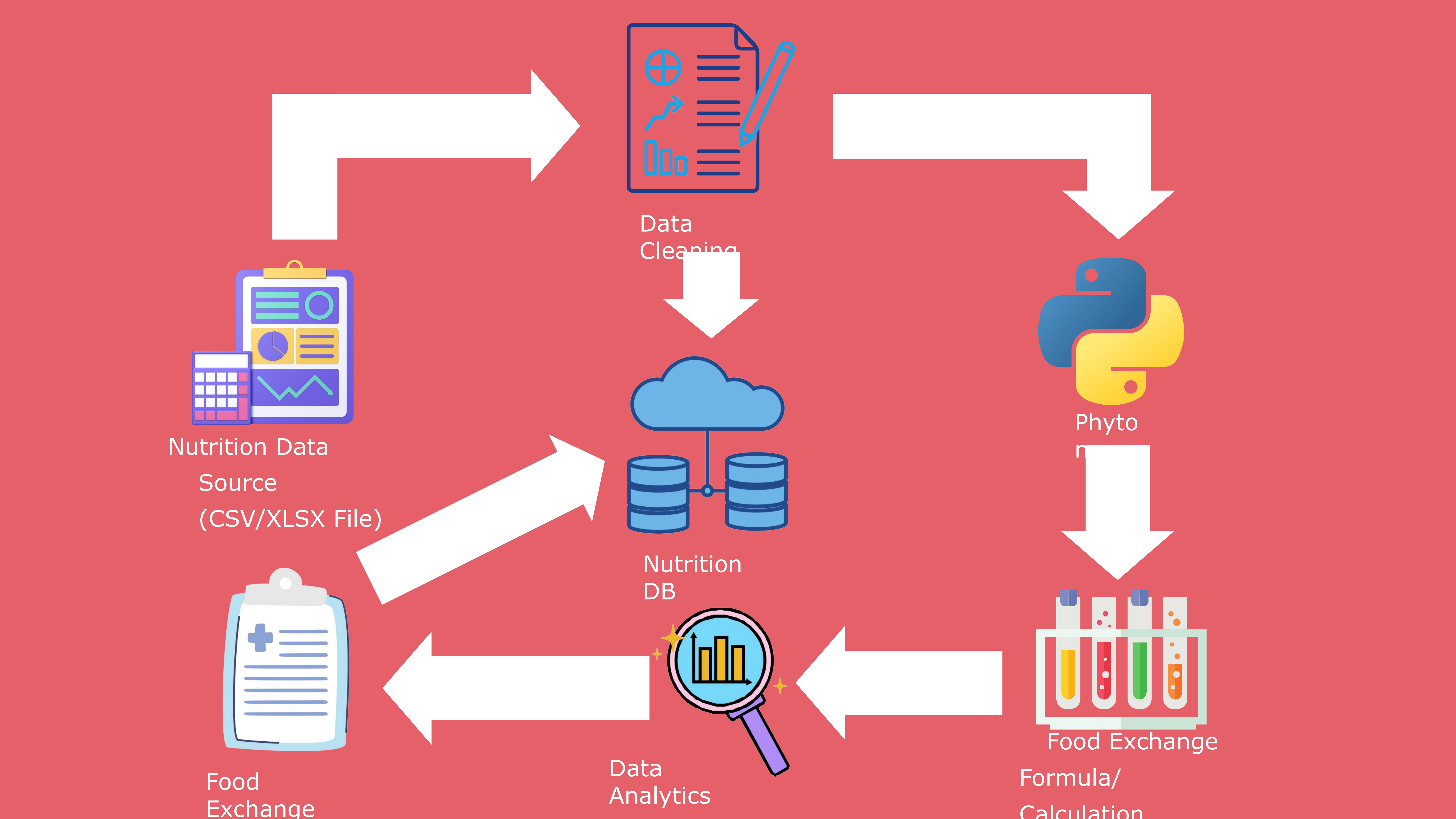
Sun Myung Hong (2003), proposed 6 Main Groups of Food Exchange

1 Grain
2. Meat (Low, Medium, High)
3. Vegetables
4. Fats and Oils
5. Milk
6. Fruits

FOOD EXCHANGE GROUP

FOOD EXCHANGE ALGORITHM

To analyze 6 Main Group of Food Exchange, researcher develop/mantain an algorithm to calculate the Nutrition food/ each food in each category



Data
Cleaning
Nutrition Data
Source
(CSV/XLSX File)
Phyto
n
Nutrition
DB
Food Exchange
Formula/
Calculation
Data
Analytics
Food
Exchange

FOOD EXHCANGE PROCESS

Analysis and vizualiation will done using csv/ xlsx file as a sample of Korean Food Data

R E A D D A T A

READ DATA

[64]:

```
data = pd.read_csv('/kaggle/input/testing-6-category/English Name.csv' )
test = pd.read_csv('/kaggle/input/testing-6-category/English Name.csv' )
```

[67]:

```
data
```

[67…

	No	category	Food Name	energy	moisture	protein	lipids	ash	carbohydrate	total sugars	saccharose	glucose	fruit sugar	lactose	maltose	galactose	gun dietery fiber	soluble dietery fiber	insoluble dietery table	calcium	steel	magnesium	sign	potassium	salt	zink	pack	magnase	salenium
0	1	grain	rice	373	9.4	11.40	3.70	2.00	33.2	0.00	0	0	0	0.0	0	0.0	0.0	0	0	16	6.6	0	175	574	2	0	0.000	0.000	0
1	2	grain	barley rice	371	9.7	14.30	3.80	1.80	37.3	0.00	0	0	0	0.0	0	0.0	0.0	0	0	18	7	0	183	385	3	0	0.000	0.000	0
2	3	grain	brown rice	382	12.0	13.20	8.20	1.70	37.3	0.00	0	0	0	0.0	0	0.0	18.8	5.2	13.6	60	5.8	0	381	383	4	0	0.000	0.000	0
3	4	grain	rice porridge	360	11.3	11.20	1.90	1.00	15.6	0.00	0	0	0	0.0	0	0.0	0.0	0	0	15	2.8	0	226	233	6	0	0.000	0.000	0
4	5	grain	millet	365	11.3	12.46	3.09	1.24	74.6	0.00	0	0	0	0.0	0	0.0	4.6	0.4	4.2	8	2.71	106	286	203	2	1.87	0.241	0.665	7.2
5	6	grain	green gram	363	13.1	13.64	3.38	2.04	23.3	0.50	0	0	0	0.0	0	0.0	6.3	0.7	5.6	21	2.78	244	453	444	1	3.09	0.541	1.469	3.16
6	7	grain	starch powder	374	10.5	12.96	3.29	1.89	80.1	0.37	0	0	0	0.0	0	0.0	8.5	1.9	6.6	17	2.92	227	435	432	1	2.99	0.522	1.478	2.67
7	8	grain	mixed grain powder	291	28.8	7.59	0.96	1.51	83.6	0.00	0	0	0	0.0	0	0.0	1.9	0.5	1.4	13	0.59	21	64	116	455	0.3	0.074	0.442	6.13
8	9	grain	flour	124	70.1	3.54	0.44	0.19	71.1	0.00	0	0	0	0.0	0	0.0	1.7	0.1	1.6	7	0.28	4	21	7	38	0.21	0.030	0.145	2.9
9	10	grain	polished rice	372	8.3	13.58	1.27	2.44	81.9	2.77	0	0.9	0	0.0	1.87	0.0	4.6	0.8	3.8	28	2.54	83	173	198	707	1.64	0.360	1.388	7.03
10	11	grain	barley (rice barley)	114	72.4	4.28	0.40	0.27	77.7	0.00	0	0	0	0.0	0	0.0	1.5	0.1	1.4	12	0.87	21	47	9	35	0.5	0.056	0.464	2.66
11	12	grain	pea	360	12.1	10.74	1.44	2.06	24.3	0.00	0	0	0	0.0	0	0.0	3.2	0.8	2.4	27	1.36	36	109	176	628	0.77	0.122	0.953	15.7
12	13	grain	adlay	141	66.2	4.22	0.59	0.28	70.5	0.00	0	0	0	0.0	0	0.0	1.4	0.4	1	18	0.56	15	40	17	68	0.32	0.042	0.380	3.58
13	14	grain	degree	58	85.6	0.86	0.23	0.51	79.2	0.00	0	0	0	0.0	0	0.0	1.7	0	1.7	7	1.45	13	24	38	120	0.13	0.031	0.119	3.45
14	15	grain	perilla	371	11.0	8.33	2.04	1.24	74.0	0.26	0	0	0	0.0	0	0.0	4.1	1.3	2.8	21	1.84	151	289	287	2	2.19	0.356	1.085	1.76
15	16	grain	Glutinous rice	357	14.9	6.50	2.00	0.70	81.9	0.60	0.6	Tr	0	0.0	0	0.0	1.3	0.3	1	7	0.9	51	150	150	1	1.6	0.220	1.540	2
16	17	grain	adzuki bean (red)	363	13.4	6.40	0.40	0.40	68.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	7	1.3	0	87	170	8	0	0.000	0.000	0
17	18	grain	Brown rice	362	14.0	6.20	0.90	0.40	77.1	0.10	0.1	Tr	0	0.0	0	0.0	0.6	Tr	0.6	5	0.8	23	96	89	2	1	0.190	0.750	4
18	19	grain	Naengmyeon (dry)	368	12.3	6.90	1.10	0.60	78.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	24	0.9	0	179	170	2	0	0.000	0.000	0
19	20	grain	Vermicelli (raw)	379	10.9	8.90	2.70	1.10	86.1	0.00	0	0	0	0.0	0	0.0	0.0	0	0	4	3	0	210	236	21	0	0.000	0.000	0
20	21	grain	dry noodles	374	10.6	9.60	4.60	1.90	78.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	4	2.8	0	327	396	26	0	0.000	0.000	0
21	22	grain	Buckwheat Noodles (Dried)	370	12.4	5.60	1.90	0.90	66.7	0.00	0	0	0	0.0	0	0.0	0.0	0	0	32	0	0	226	298	9	0	0.000	0.000	0
22	23	grain	Buckwheat noodles (raw)	371	12.9	5.80	4.80	1.20	61.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	14	0.8	0	305	275	1	0	0.000	0.000	0

FOODDATA

Food Data

+ Code + Markdown

[2]:

```
import pandas as pd
fooddata = pd.read_csv('/kaggle/input/testing-6-category/English Name.csv')

#if using Food name in english use '/kaggle/input/new-latest-food-data/food (2).csv'
#if using Food name in Korea use ''
```

[3]:

```
fooddata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 666 entries, 0 to 665
Columns: 134 entries, No to No0Name03
dtypes: float64(57), int64(24), object(53)
memory usage: 697.3+ KB
```

[4]:

```
fooddata.columns
```

[4]:

```
Index(['No', 'category', 'Food Name', 'energy', 'moisture', 'protein',
       'lipids', 'ash', 'carbohydrate', 'total sugars',
       ...
       'Docosapentaenoic0Acid0(22:50(n03))',
       'Docosahexaenoic0Acid0(22:60(n03))', 'Omega03 Fatty0Acid',
       'Omega06 Fatty0Acid', 'Gun Trans Fatty0Acid', 'Trans Oleic Acid (18:1)',
       'trans Linoleic Acid (18)', 'trans Linolenic Acid', 'No0Name02',
       'No0Name03'],
      dtype='object', length=134)
```

```
fooddata
```

[6]:

	No	category	Food Name	energy	moisture	protein	lipids	ash	carbohydrate	total sugars	saccharose	glucose	fruit sugar	lactose	maltose	galactose	gun dietery fiber	soluble dietery fiber	insoluble dietery table	calcium	steel	magnesium	sign	potassium	salt	zink	pack	magnase	salenium
0	1	grain	rice	373	9.4	11.40	3.70	2.00	33.2	0.00	0	0	0	0.0	0	0.0	0.0	0	0	16	6.6	0	175	574	2	0	0.000	0.000	0
1	2	grain	barley rice	371	9.7	14.30	3.80	1.80	37.3	0.00	0	0	0	0.0	0	0.0	0.0	0	0	18	7	0	183	385	3	0	0.000	0.000	0
2	3	grain	brown rice	382	12.0	13.20	8.20	1.70	37.3	0.00	0	0	0	0.0	0	0.0	18.8	5.2	13.6	60	5.8	0	381	383	4	0	0.000	0.000	0
3	4	grain	rice porridge	360	11.3	11.20	1.90	1.00	15.6	0.00	0	0	0	0.0	0	0.0	0.0	0	0	15	2.8	0	226	233	6	0	0.000	0.000	0
4	5	grain	millet	365	11.3	12.46	3.09	1.24	74.6	0.00	0	0	0	0.0	0	0.0	4.6	0.4	4.2	8	2.71	106	286	203	2	1.87	0.241	0.665	7.2
5	6	grain	green gram	363	13.1	13.64	3.38	2.04	23.3	0.50	0	0	0	0.0	0	0.0	6.3	0.7	5.6	21	2.78	244	453	444	1	3.09	0.541	1.469	3.16
6	7	grain	starch powder	374	10.5	12.96	3.29	1.89	80.1	0.37	0	0	0	0.0	0	0.0	8.5	1.9	6.6	17	2.92	227	435	432	1	2.99	0.522	1.478	2.67
7	8	grain	mixed grain powder	291	28.8	7.59	0.96	1.51	83.6	0.00	0	0	0	0.0	0	0.0	1.9	0.5	1.4	13	0.59	21	64	116	455	0.3	0.074	0.442	6.13
8	9	grain	flour	124	70.1	3.54	0.44	0.19	71.1	0.00	0	0	0	0.0	0	0.0	1.7	0.1	1.6	7	0.28	4	21	7	38	0.21	0.030	0.145	2.9
9	10	grain	polished rice	372	8.3	13.58	1.27	2.44	81.9	2.77	0	0.9	0	0.0	1.87	0.0	4.6	0.8	3.8	28	2.54	83	173	198	707	1.64	0.360	1.388	7.03
10	11	grain	barley (rice barley)	114	72.4	4.28	0.40	0.27	77.7	0.00	0	0	0	0.0	0	0.0	1.5	0.1	1.4	12	0.87	21	47	9	35	0.5	0.056	0.464	2.66
11	12	grain	pea	360	12.1	10.74	1.44	2.06	24.3	0.00	0	0	0	0.0	0	0.0	3.2	0.8	2.4	27	1.36	36	109	176	628	0.77	0.122	0.953	15.7
12	13	grain	adlay	141	66.2	4.22	0.59	0.28	70.5	0.00	0	0	0	0.0	0	0.0	1.4	0.4	1	18	0.56	15	40	17	68	0.32	0.042	0.380	3.58
13	14	grain	degree	58	85.6	0.86	0.23	0.51	79.2	0.00	0	0	0	0.0	0	0.0	1.7	0	1.7	7	1.45	13	24	38	120	0.13	0.031	0.119	3.45
14	15	grain	perilla	371	11.0	8.33	2.04	1.24	74.0	0.26	0	0	0	0.0	0	0.0	4.1	1.3	2.8	21	1.84	151	289	287	2	2.19	0.356	1.085	1.76
15	16	grain	Glutinous rice	357	14.9	6.50	2.00	0.70	81.9	0.60	0.6	Tr	0	0.0	0	0.0	1.3	0.3	1	7	0.9	51	150	150	1	1.6	0.220	1.540	2
16	17	grain	adzuki bean (red)	363	13.4	6.40	0.40	0.40	68.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	7	1.3	0	87	170	8	0	0.000	0.000	0
17	18	grain	Brown rice	362	14.0	6.20	0.90	0.40	77.1	0.10	0.1	Tr	0	0.0	0	0.0	0.6	Tr	0.6	5	0.8	23	96	89	2	1	0.190	0.750	4
18	19	grain	Naengmyeon (dry)	368	12.3	6.90	1.10	0.60	78.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	24	0.9	0	179	170	2	0	0.000	0.000	0
19	20	grain	Vermicelli (raw)	379	10.9	8.90	2.70	1.10	86.1	0.00	0	0	0	0.0	0	0.0	0.0	0	0	4	3	0	210	236	21	0	0.000	0.000	0
20	21	grain	dry noodles	374	10.6	9.60	4.60	1.90	78.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	4	2.8	0	327	396	26	0	0.000	0.000	0
21	22	grain	Buckwheat Noodles (Dried)	370	12.4	5.60	1.90	0.90	66.7	0.00	0	0	0	0.0	0	0.0	0.0	0	0	32	0	0	226	298	9	0	0.000	0.000	0
22	23	grain	Buckwheat noodles	371	12.9	5.80	4.80	1.20	61.4	0.00	0	0	0	0.0	0	0.0	0.0	0	0	14	0.8	0	305	275	1	0	0.000	0.000	0

FOOD DATA ANLYSIS AND VISUALIZATION

1. GRAIN

METHOD TO COUNT

1. Grain I.Method to count the exchange amount of cereals [c=carbohydrate (g), p=protein (g), f=fat(g)]
I-1 = Cg(amount of sugar per 100g optional food) 23g(amount of protein in grain exchange) *Note: Since it is based on the amount of sugar, the number of exchanges per 100g is calculated by dividing the amount of sugar in the selected food by 23 I-2 = 100g(amount of food composition table) (I-1) Number of exchanges of selected food) *Note: The exchange amount is calculated by dividing 100g of food by the number of exchanges (I-2 is the exchange amount of selected Food

1. Grain

+ Code + Markdown

```python
#specific for data contains grain
df_food4 = df.drop(['No','energy','moisture','protein','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble dieter
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('grain')]
df_food4.head()
```

[26...

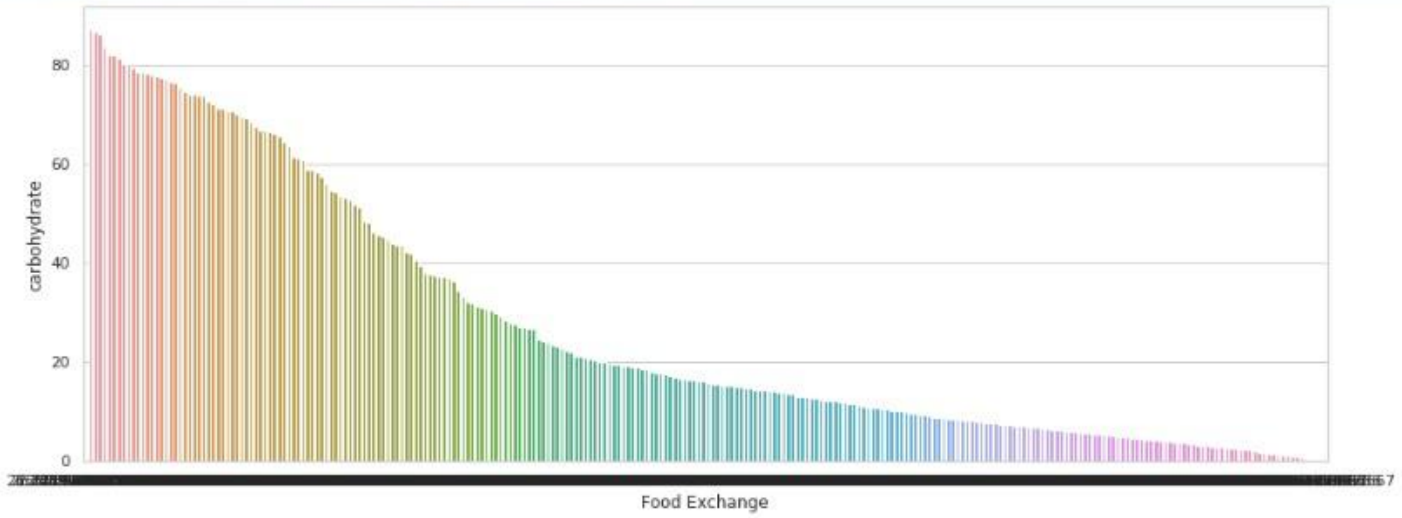
	category	Food Name	carbohydrate
0	grain	rice	33.2
1	grain	barley rice	37.3
2	grain	brown rice	37.3
3	grain	rice porridge	15.6
4	grain	millet	74.6

```python
# Food Exchange for Grain

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/23g (Professor Hong)
#I-2= 100G/I-1
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

df_food4['Food Exchange'] = 0
df_food4['denom']=df_food4['carbohydrate']/23
df_food4['Food Exchange']= df_food4['denom'].rdiv(100)
df_food4=df_food4.drop(['denom'],axis=1)
df_food4.replace([np.inf, -np.inf], 0 , inplace=True)
pd.options.display.float_format = '{:.2f}'.format
df_food4
```

	category	Food Name	carbohydrate	Food Exchange
0	grain	rice	33.20	69.28
1	grain	barley rice	37.30	61.66
2	grain	brown rice	37.30	61.66
3	grain	rice porridge	15.60	147.44
4	grain	millet	74.60	30.83
5	grain	green gram	23.30	98.71
6	grain	starch powder	80.10	28.71
7	grain	mixed grain powder	83.60	27.51
8	grain	flour	71.10	32.35
9	grain	polished rice	81.90	28.08
10	grain	barley (rice barley)	77.70	29.60

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'carbohydrate', data=df_food4);
```

```
/opt/conda/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



2 .MEAT

METHOD TO COUNT

2. Meat In the fish and meat group, the exchange is 8g of protein, 2 g (low), 5g (medium), and 8g (high) of lipid, 50kcal, 75kcal, and 100kcal of energy and the amount of protein Since it is constant, it was calculated based on this. I.Calculation Method [c=carbohydrate (g), p=protein (g), f=fat(g)]
I-1 = Pg (amount of protein per 100g of optional food) 8g (amount of protein in 1 exchange of fish and meat) *Note : Since it is based on the amount of protein, dividing the amount of protein in the selected food by 8 is equivalent to 100g food. One exchange number is calculated I-2 = 100g (amount of food component table value) (I-1) (number of exchange in selected food) *Note : Divide 100g of food by the number of exchanges to calculate the exchange amount. (I-2) is the exchange amount of the selected food

2.1 MEAT LOW FAT

2.1 Meat Low Fat

+ Code   + Markdown

```
#specific for data contains meat low fat
df_food4 = df.drop(['No','energy','moisture','carbohydrate','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble d
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('meat low')]
df_food4
```

[30...

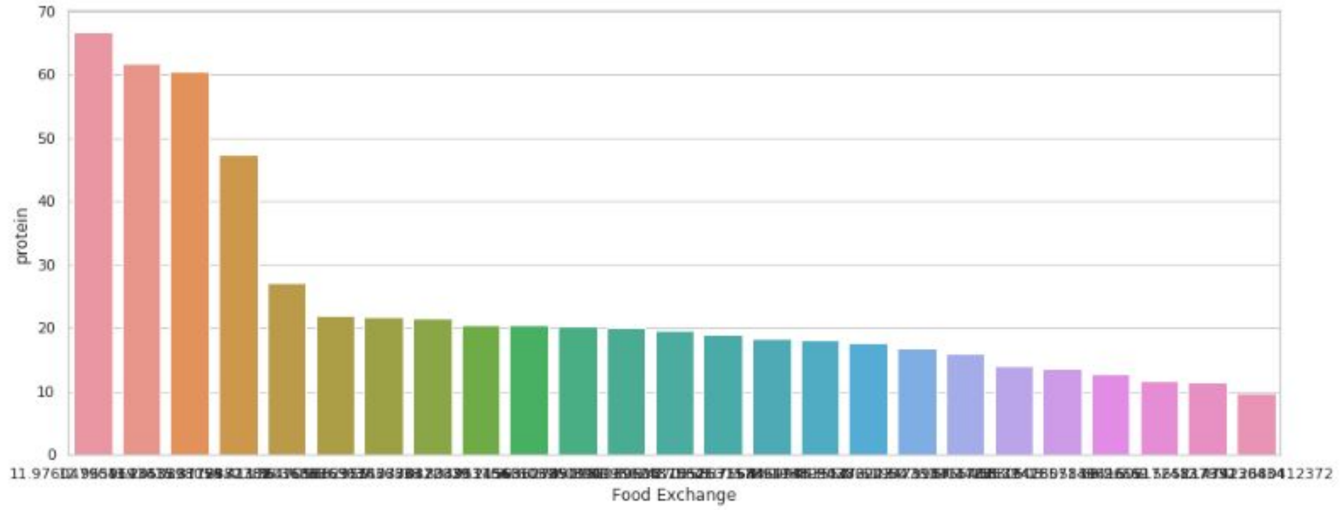
	category	Food Name	protein
554	meat low	beef, meat, domestic, landslide	20.20
555	meat low	pork, meat, avalanche, raw	22.00
556	meat low	Beef, cooked by-products, liver, raw	19.00
557	meat low	Whitefish, Po	60.40
558	meat low	Pollack, dried fish, adult fish (pollack fish)	61.70
559	meat low	Cuttlefish (squid), dried	66.80
560	meat low	anchovy, self-produced, big anchovy	47.40
561	meat low	flounder, raw, flounder	16.80
562	meat low	Halibut (flatfish), raw	20.40
563	meat low	Pollack, frozen product (frozen)	15.90

```python
#Food Exchange for Meat Low Fat

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/8g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['protein']/8)
df_food4
```

[31…

	category	Food Name	protein	Food Exchange
554	meat low	beef, meat, domestic, landslide	20.20	39.60
555	meat low	pork, meat, avalanche, raw	22.00	36.36
556	meat low	Beef, cooked by-products, liver, raw	19.00	42.11
557	meat low	Whitefish, Po	60.40	13.25
558	meat low	Pollack, dried fish, adult fish (pollack fish)	61.70	12.97
559	meat low	Cuttlefish (squid), dried	66.80	11.98
560	meat low	anchovy, self-produced, big anchovy	47.40	16.88
561	meat low	flounder, raw, flounder	16.80	47.62
562	meat low	Halibut (flatfish), raw	20.40	39.22
563	meat low	Pollack, frozen product (frozen)	15.90	50.31
564	meat low	Dolphin, red snapper, raw	21.60	37.04
565	meat low	tuna, bluefin tuna, raw, adult fish	27.20	29.41
566	meat low	blowfish, raw, whitefish	15.90	50.31
567	meat low	Early (reference), raw	18.30	43.72
568	meat low	Shrimp, lobster, raw	18.10	44.20
569	meat low	cockle	14.00	57.14
570	meat low	abalone, raw, horse abalone	12.80	62.50
571	meat low	Mussels, raw	9.70	82.47
572	meat low	crab, blue crab, raw	13.70	58.39
573	meat low	oysters, raw, wild oysters, wild oysters	11.60	68.97
574	meat low	Octopus (three-legged octopus)	11.50	69.57

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'protein', data=df_food4);
```



2.2 MEAT MEDIUM FAT

2.2 Meat Medium Fat

+ Code + Markdown

```
#specific for data contains meat medium fat
df_food4 = df.drop(['No','energy','moisture','carbohydrate','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble d
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('meat medium')]
df_food4
```

[33...

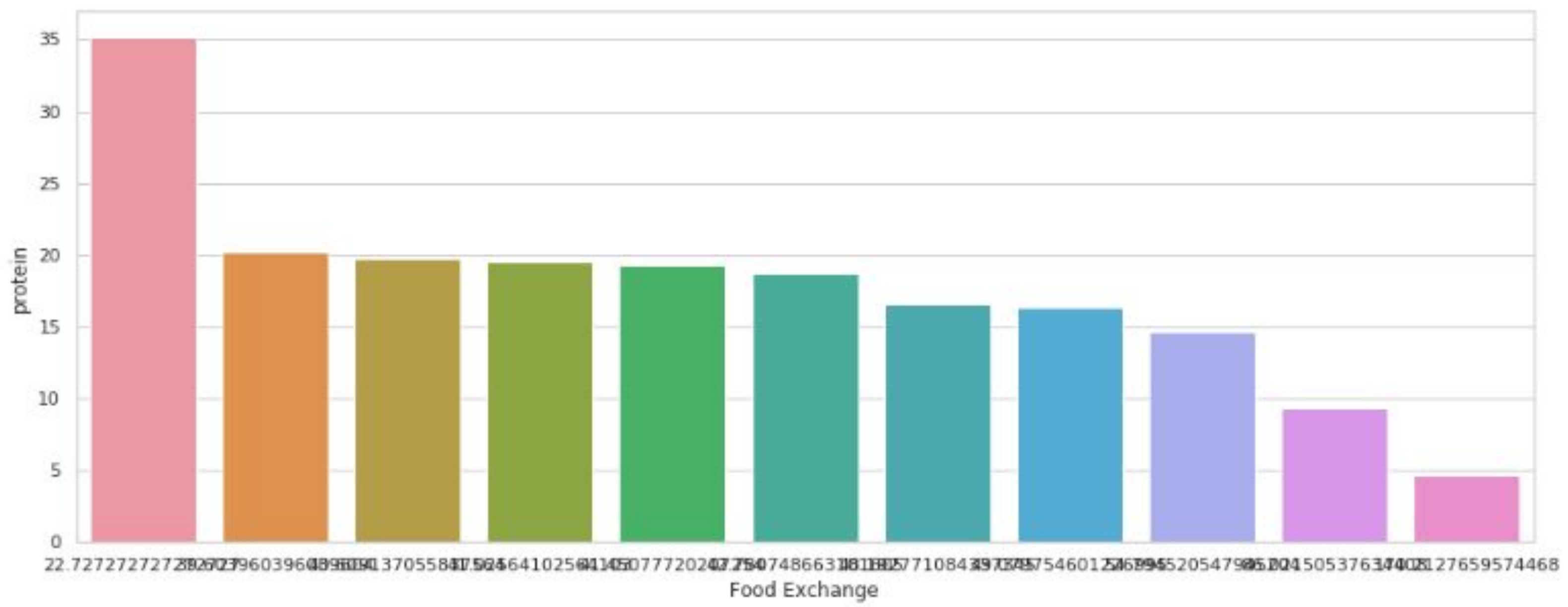
	category	Food Name	protein
581	meat medium	sandfish, raw	14.60
582	meat medium	herring, raw	19.30
583	meat medium	Pork, pork products, ham, slices	18.70
584	meat medium	mackerel, raw	20.20
585	meat medium	saury, raw	19.50
586	meat medium	freshwater fish, raw	19.70
587	meat medium	eel, hagfish (spike eel)	16.60

```python
#Food Exchange for Meat Medium Fat

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/8g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['protein']/8)
df_food4
```

[34…

	category	Food Name	protein	Food Exchange
581	meat medium	sandfish, raw	14.60	54.79
582	meat medium	herring, raw	19.30	41.45
583	meat medium	Pork, pork products, ham, slices	18.70	42.78
584	meat medium	mackerel, raw	20.20	39.60
585	meat medium	saury, raw	19.50	41.03
586	meat medium	freshwater fish, raw	19.70	40.61
587	meat medium	eel, hagfish (spike eel)	16.60	48.19
588	meat medium	Soybean, black bean, dried, black bean	35.20	22.73
589	meat medium	soybean, tofu, silken tofu	4.70	170.21
590	meat medium	mackerel, canned food	16.30	49.08
591	meat medium	soybean, tofu, tofu	9.30	86.02

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'protein', data=df_food4);
```



2.3 MEAT HIGH FA

2.3 Meat High Fat

[36]:

```
#specific for data contains meat medium fat
df_food4 = df.drop(['No','energy','moisture','carbohydrate','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble d
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('meat high')]
df_food4
```

[36...

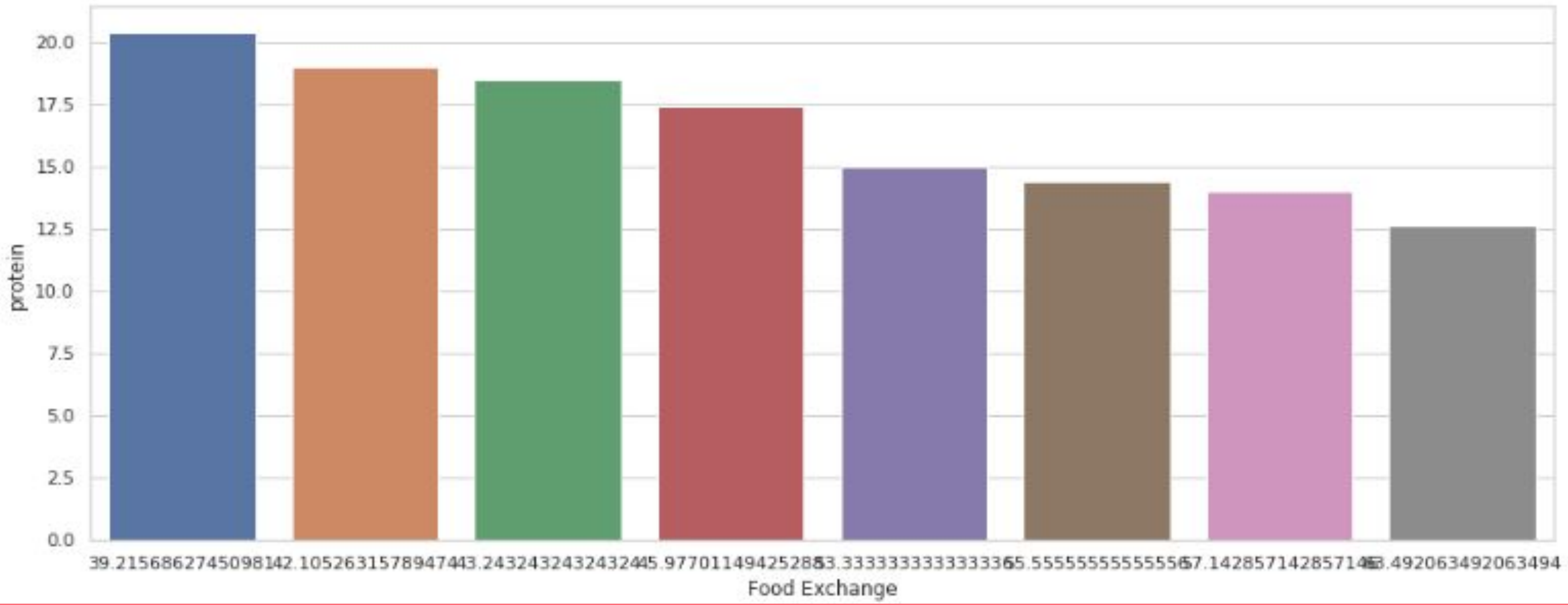
	category	Food Name	protein
592	meat high	dog meat	19.00
593	meat high	Quail eggs, whole eggs, raw	12.60
594	meat high	Soybean, Tofu, Fried Tofu (Tofu)	20.40
595	meat high	Beef, meat, imported, ribs, raw	18.50
596	meat high	beef, beef by-products, tail	17.40
597	meat high	Pork, pork products, sausages, frankfurt	14.40
598	meat high	saury, canned food	14.00
599	meat high	Eel, eel (device), raw	14.40
600	meat high	Pork, pork products, ham, luncheon meat	15.00

```
#Food Exchange for Meat Medium Fat

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/8g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['protein']/8)
df_food4
```

[38…

	category	Food Name	protein	Food Exchange
592	meat high	dog meat	19.00	42.11
593	meat high	Quail eggs, whole eggs, raw	12.60	63.49
594	meat high	Soybean, Tofu, Fried Tofu (Tofu)	20.40	39.22
595	meat high	Beef, meat, imported, ribs, raw	18.50	43.24
596	meat high	beef, beef by-products, tail	17.40	45.98
597	meat high	Pork, pork products, sausages, frankfurt	14.40	55.56
598	meat high	saury, canned food	14.00	57.14
599	meat high	Eel, eel (device), raw	14.40	55.56
600	meat high	Pork, pork products, ham, luncheon meat	15.00	53.33

```
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'protein', data=df_food4);
```



3 .VEGETABLES

METHOD TO COUNT

3. Vegetables Unit of vegetable group is 3g of carbohydrate, 2 g of protein, and 20 kcal of calories, and it was calculated based on carbohydrate I. Calculation method of exchange amount of vegetable group [c=carbohydrate (g), P=protein(g), F=fat(g)] I-1 = Cg (amount of sugar per 100g of selected food) 3g (amount of sugar in 1 vegetable group replacement) *Note: Since it based on sugar the sugar content on the selected food dividing the amount of by 3 yields the number of exchanges per 100g. I-2 = 100g (amount of vegetable ingredient table) I-1 (Number of exchange of selected food) *Note : divide 100g of food by the number of exchanges to calculate the exchange amount. (I-2 is the exchange amount of selected food)

3. Vegetables

```
#specific for data contains vegetables
df_food4 = df.drop(['No','energy','moisture','protein','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble dieter
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('vegetables')]
df_food4
```

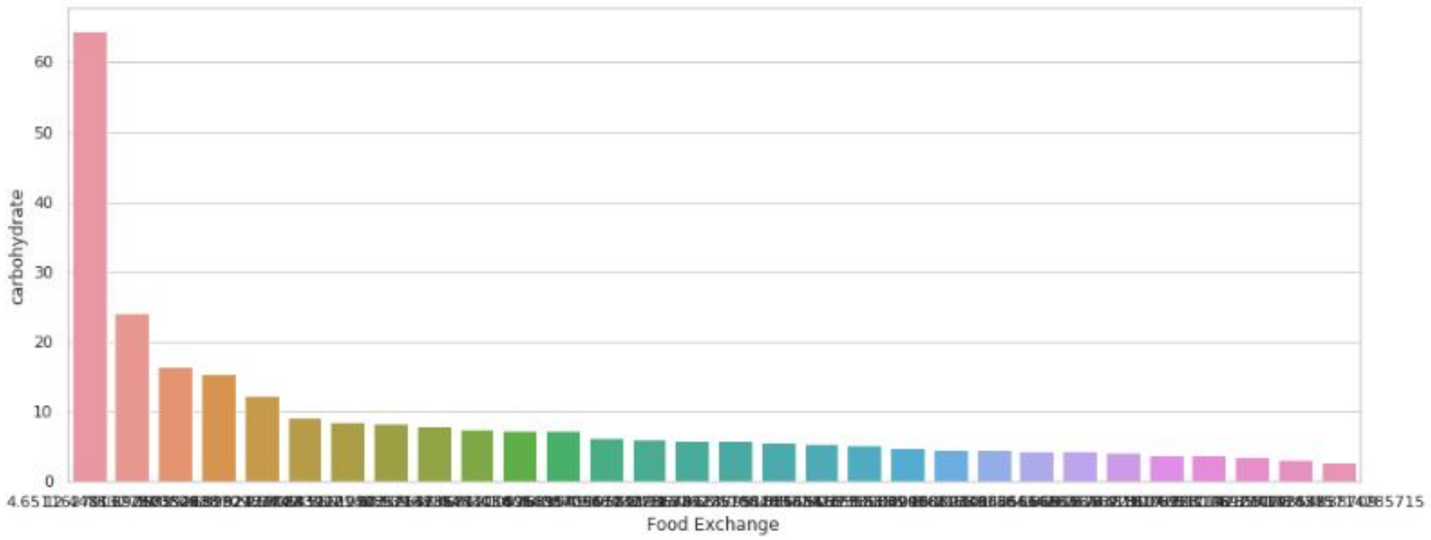
	category	Food Name	carbohydrate
601	vegetables	radish, dried radish	64.50
602	vegetables	Red pepper leaves, raw	9.30
603	vegetables	deodeok, raw	12.30
604	vegetables	Burdock, raw	15.50
605	vegetables	cold sore	5.40
606	vegetables	bellflower, raw	24.10
607	vegetables	gulsa, ground gulch, whole, raw	4.30
608	vegetables	Radish, Korean radish, leaf (radish green)	4.50
609	vegetables	Shiitake, oak, raw (raw)	6.10
610	vegetables	Onion, raw, domestic	8.40
611	vegetables	Kale	7.30
612	vegetables	Perilla leaves, raw	7.90

```python
#Food Exchange for Vegetables

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/3g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['carbohydrate']/3)
df_food4
```

[41…

	category	Food Name	carbohydrate	Food Exchange
601	vegetables	radish, dried radish	64.50	4.65
602	vegetables	Red pepper leaves, raw	9.30	32.26
603	vegetables	deodeok, raw	12.30	24.39
604	vegetables	Burdock, raw	15.50	19.35
605	vegetables	cold sore	5.40	55.56
606	vegetables	bellflower, raw	24.10	12.45
607	vegetables	gulsa, ground gulch, whole, raw	4.30	69.77
608	vegetables	Radish, Korean radish, leaf (radish green)	4.50	66.67
609	vegetables	Shiitake, oak, raw (raw)	6.10	49.18
610	vegetables	Onion, raw, domestic	8.40	35.71
611	vegetables	Kale	7.30	41.10
612	vegetables	Perilla leaves, raw	7.90	37.97
613	vegetables	eggplant, raw	5.30	56.60
614	vegetables	Kimchi, Kkakdugi	7.40	40.54
615	vegetables	gobi, raw (wild)	6.20	48.39
616	vegetables	modern, raw	3.10	96.77
617	vegetables	Oyster mushrooms, raw	5.80	51.72

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'carbohydrate', data=df_food4);
```



4 .F A T S A N D O I L

METHOD TO COUNT

4. Fats and Oils In the fat group, the unit of 1 exchange is 5g of fat and 45 kcal of energy, The formula was written based of the amount of fat I. Calculation method for the exchange amount of the local group [c=carbohydrate (g), p=protein(g), f=fat(g)] I-1 = Fg(amount of fat per 100g of optional food) 5g (amount of lipid in 1 exchange of fat group) *Note: Since it based on fat, if the amount of fat in the selected food is divided by 5 , the exchange number for 100g is calculated I-2 = 100g(food ingredient table) Value of I-1 (number of exchange og selected food) *Note: Divide 100g of food by the number of exchanges to calculate the exchange amount. (I-2 is the exchange amount of selected food)

4.Fats and Oil

+ Code + Markdown

```
#specific for data contains oil
df_food4 = df.drop(['No','energy','moisture','protein','carbohydrate','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('fat')]
df_food4
```

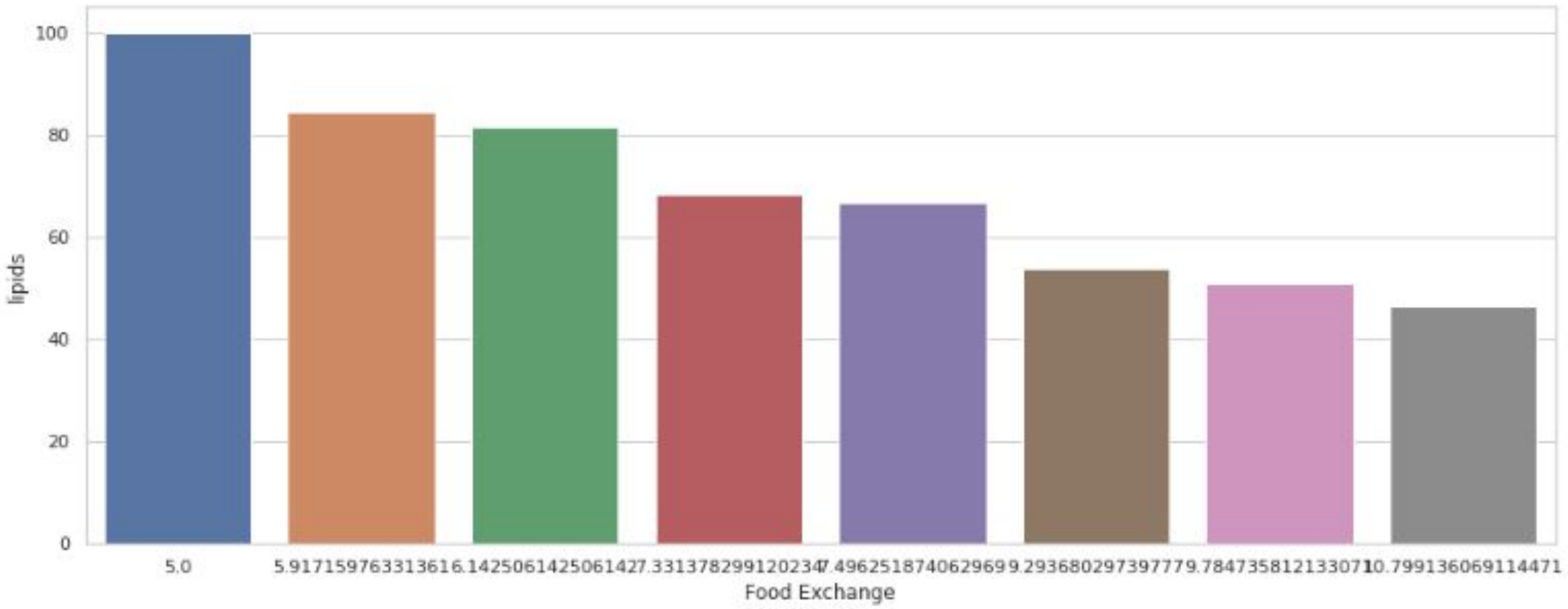
[43...

	category	Food Name	lipids
634	fat	perilla oil	100.00
635	fat	soybean oil	100.00
636	fat	margarine	81.40
637	fat	butter	84.50
638	fat	peanut butter, peanut butter	51.10
639	fat	Sesame, white sesame, roasted	53.80
640	fat	pine nuts, dry	68.20
641	fat	walnuts, dried	66.70
642	fat	Peanuts, dry, small grain	46.30

```python
#Food Exchange for Fats and Oil

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/5g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['lipids']/5)
df_food4
```

	category	Food Name	lipids	Food Exchange
634	fat	perilla oil	100.00	5.00
635	fat	soybean oil	100.00	5.00
636	fat	margarine	81.40	6.14
637	fat	butter	84.50	5.92
638	fat	peanut butter, peanut butter	51.10	9.78
639	fat	Sesame, white sesame, roasted	53.80	9.29
640	fat	pine nuts, dry	68.20	7.33
641	fat	walnuts, dried	66.70	7.50
642	fat	Peanuts, dry, small grain	46.30	10.80

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'lipids', data=df_food4);
```



5 .MILK

METHOD TO COUNT

5. Milk In the milk group, 1exchange unit is 11g of sugar, 6 g of protein, and 6g lipid. 125 kcal in calories, and contain various nutrients, so it is difficult to choose the main ingredient, so it was calculated based on the amount of sugar I. Calculation method of exchange amount of milk group [c=carbohydrate (g), p=protein(g), f=fat(g)] I-1 = Cg (amount of sugar per 100g of optional food) 11g(amount of sugar in 1 exchange of milk group) *Note: since it is based on sugar, if the amount of sugar in the selected food is divided by 11, the exchange number for 100g is come out I-2 = 100g (amount of food component table) (I- 1) (number of exchanges of selected food) *Note: Divide 100g food by the number of exchanges to calculate the exchange amount. (I-2 is the exchange amount of selected food)

5. Milk

+ Code + Markdown

```
#specific for data contains milk
df_food4 = df.drop(['No','energy','moisture','protein','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble dieter
                    'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                    'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                     'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                    'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                    'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                    'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                    'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                    'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                    'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                    'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                    'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                    'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                    'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('milk')]
df_food4
```

[46…

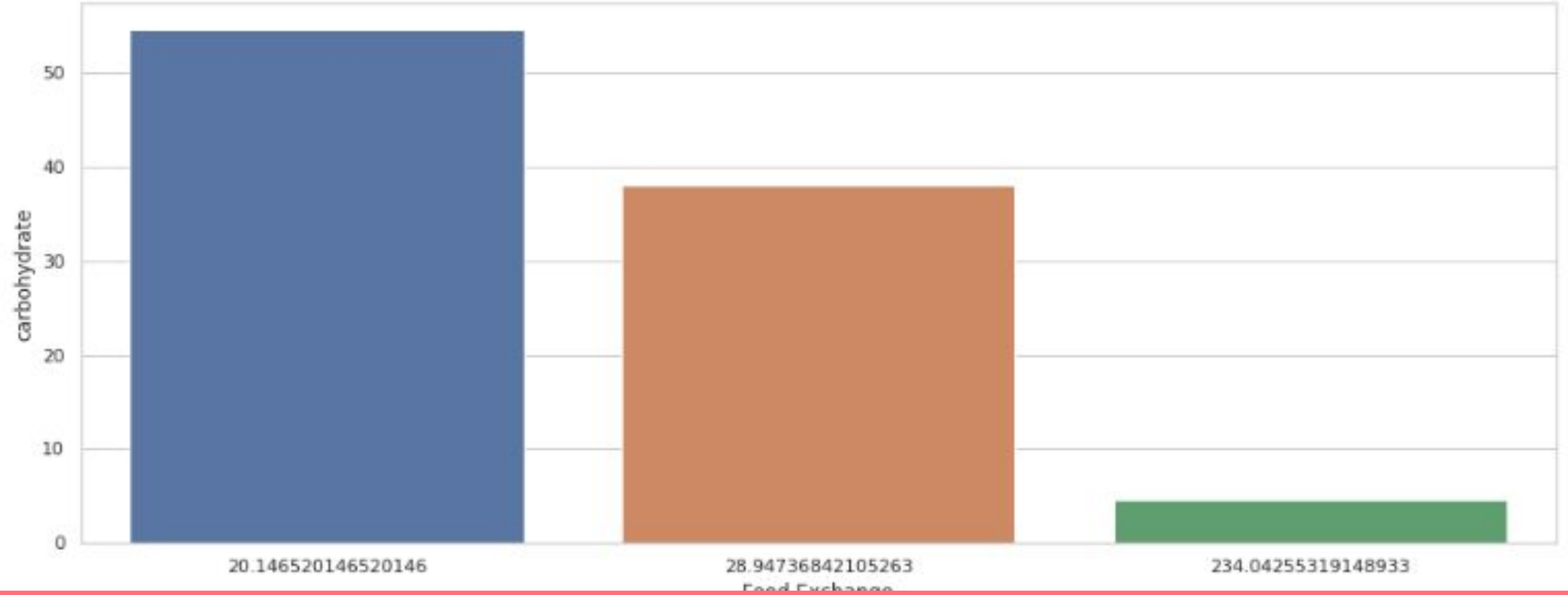
	category	Food Name	carbohydrate
643	milk	milk, milk, plain milk	4.70
644	milk	Milk, powdered milk, whole milk powder	38.00
645	milk	Milk, powdered milk, powdered milk 1st stage	54.60

```python
#Food Exchange for Milk

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/11g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['carbohydrate']/11)
df_food4
```

[47...

	category	Food Name	carbohydrate	Food Exchange
643	milk	milk, milk, plain milk	4.70	234.04
644	milk	Milk, powdered milk, whole milk powder	38.00	28.95
645	milk	Milk, powdered milk, powdered milk 1st stage	54.60	20.15

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'carbohydrate', data=df_food4);
```



6 .F R U I T S

METHOD TO COUNT

6. Fruits In the fruit group , the unit of 1 exchange is 12g of sugar and 50kcal of calories, and the formula was written based onl. Calculation method of exchange amount of fruit group [c=carbohydrate (g), p=protein(g), f=fat(g)] I-1 = Cg (amount of sugar per 100g of optional food) 12g(amount of sugar in 1 exchange of milk group) *Note: since it is based on sugar, the number of exchanges for 100g is calculated by dividing the amount of sugar in selected food by 12. I-2 = 100g (amount of food component table) (I-1) (number of exchanges of selected food) *Note: Divide 100g food by the number yields of exchanges amount. (I-2 is the exchange amount of selected food)

6.Fruits

+ Code + Markdown

```
#specific for data contains oil
df_food4 = df.drop(['No','energy','moisture','protein','lipids','ash','total sugars','saccharose','glucose','fruit sugar','lactose','maltose','galactose','gun dietery fiber','soluble dieter
                   'insoluble dietery table','calcium','steel','magnesium','sign','potassium','salt','zink','pack','magnase','salenium','molybdenum','iodine','retinol','beta carotene','Vita
                   'Niacin','Niacin (NE)','Nicotinic Acid','Pantothenic Acid','Tridecanoic acid (13:0)','Myristic acid (14:0)','Pentadecanoic Acid(15:0)','Palmitic Acid(16:0)','Heptadecanoi
                    'Arachidic Acid (20:0)','Heneikosansan (21:0)','Behenic Acid (22:0)','Nicotinamide','Vitamin B6','Pyridoxine','Biotin','Vitamin B12','Vitamin C','Vitamin D',
                   'Vitamin D2','Vitamin D3','Vitamin E','Alpha Tocopherol','Beta Tocopherol','Gamma Tocopherol','Delta Tocopherol','Alpha Tocotrienol','Beta Tocotrienols',
                   'Delta Tocotrienols','Vitamin K','Vitamin K2','Vitamin K1','Favour','Essential','Isoleucine','Leucine','Lysine','Phenylalanine','Threonine','Tryptophan',
                   'Valine','Methionine','Histidine','Arginine','Non essential','Tyrosine','Cysteine','Alanine','Aspartic Acid','Glutamic Acid','Glycine','Proline','Serine',
                   'No Name 1','Gun','taurine','Butyric0Acid0(4:0)','Caproic Acid (6:0)','Caprylic Acid (8:0)','Capric Acid (10:0)','Lauric acid (12:0)','Trichosanoic Acid (23:0)',
                   'lignoceric Acid (24:0)','Myristoleic Acid (14:1)','Palmitoleic Acid (16:1)','Heptadecenoic Acid(17:1)','Oleic Acid (18:1(n09))','Bacsenic Acid (18:10(n07))',
                   'Erucic0Acid0(22:1)','Mount0Nerbon0(24:1)','Linoleic0Acid0(18:2(n06)c)','Docosapentaenoic0Acid0(22:50(n03))','Eicosapentaenoic0Acid0(20:50(n03))','Eicosatrienoic0Acid0(20
                   'Docosadienoic0Acid0(22:2)','Docosahexaenoic0Acid0(22:60(n03))','No0Name03','No0Name02','Gamma Tocotrienols','Folic Acid (DFE)','Folic Acid Food Folic Acid','Gun Essentia
                   'Gun Single Unsaturated','Gun Multiple Unsaturated 1','Gadoleic0Acid0(20:1)','Alpha0Linolenic0Acid0(18:30(n03))','Gamma0Linolenic0Acid0(18:30(n06))','Eicosadienoic0Acid0(
                   'Eicosatrienoic0Acid0(20:30(n06))','Arachidonic0Acid0(20:40(n06))','Omega03 Fatty0Acid','Omega06 Fatty0Acid','Gun Trans Fatty0Acid','Trans Oleic Acid (18:1)',
                   'trans Linoleic Acid (18)','trans Linolenic Acid','Folic  Acid Added Folic Acid'], axis=1)
df_food4

df_food4=df_food4[df_food4['category'].str.contains('fruit')]
df_food4
```

[52…

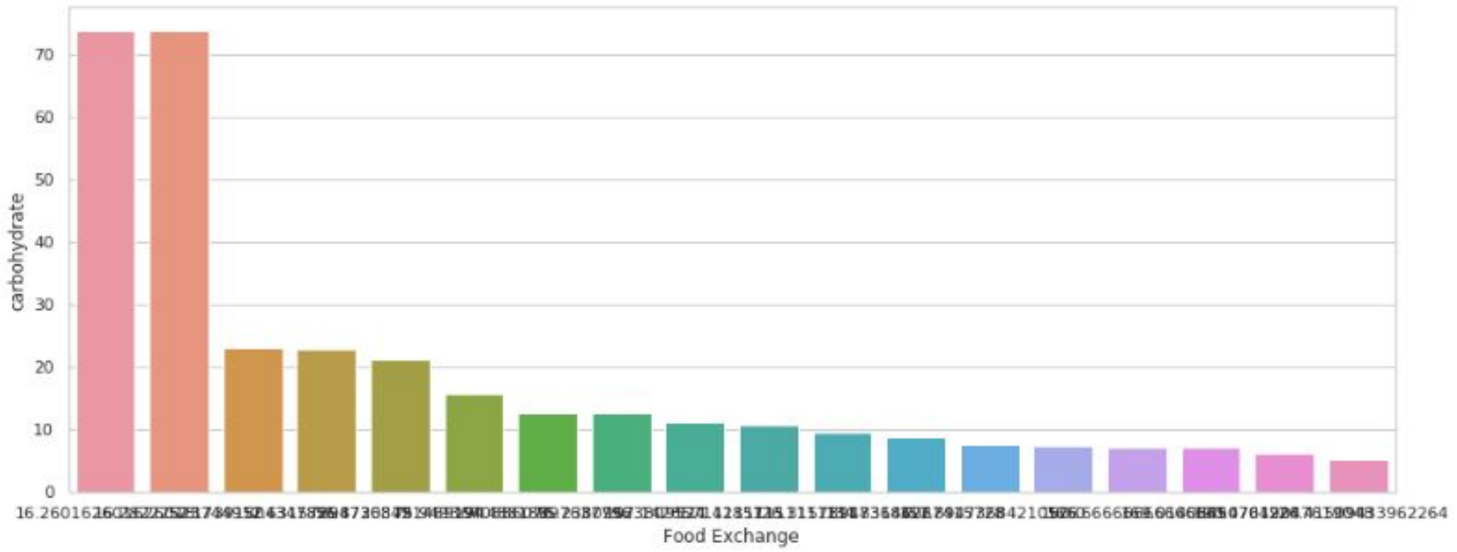
	category	Food Name	carbohydrate
646	fruit	jujube, dried	73.70
647	fruit	grapes, raisins	73.80
648	fruit	jujube, raw	22.80
649	fruit	banana, raw	21.10
650	fruit	grapes, green grapes	12.60
651	fruit	persimmon, sweet persimmon	23.00
652	fruit	plums, raw	5.30
653	fruit	Citrus fruit, mandarin orange, raw, plain (lmo...	10.80
654	fruit	Pears, Raw, Made in China	12.70
655	fruit	apple, raw, adverb (fuji)	15.80

```python
#Food Exchange for Fruit

#formula I-1
#I-1= Fg(Amount of fat per 100g of optional food)/12g (Professor Hong)
#I-2= 100G/I-1
df_food4['Food Exchange'] = 100/(df_food4['carbohydrate']/12)
df_food4
```

[53...

	category	Food Name	carbohydrate	Food Exchange
646	fruit	jujube, dried	73.70	16.28
647	fruit	grapes, raisins	73.80	16.26
648	fruit	jujube, raw	22.80	52.63
649	fruit	banana, raw	21.10	56.87
650	fruit	grapes, green grapes	12.60	95.24
651	fruit	persimmon, sweet persimmon	23.00	52.17
652	fruit	plums, raw	5.30	226.42
653	fruit	Citrus fruit, mandarin orange, raw, plain (imo...	10.80	111.11
654	fruit	Pears, Raw, Made in China	12.70	94.49
655	fruit	apple, raw, adverb (fuji)	15.80	75.95
656	fruit	orange, raw	11.20	107.14
657	fruit	pineapple, raw	6.30	190.48
658	fruit	Grapefruit (grapefruit), raw	7.60	157.89
659	fruit	melon, musk	9.50	126.32
660	fruit	cherry	7.20	166.67
661	fruit	melon, raw	7.50	160.00
662	fruit	peach, raw, yellow peach	6.30	190.48

```python
#Analysis of Food Exchange Value
#Salaki Reynaldo Joshua
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 6))
sns.barplot('Food Exchange', 'carbohydrate', data=df_food4);
```



SAMPLE COMPARISON RESULT

After develop the algorithm to calculate, we want to compare our result with Prevuious research by (Sun Myung Hong)

2.1 MEAT LOW FAT

Table 4. 어육류(저)군 1교환단위의 식품교환량과 ENECC/e-식품교환량

식 품	식품 교환량(g)	ENECC 실험값(g)[1]	추가 혹은 부족된 식품 교환수[2]		ENECC/e-식품 교환량(g)[3]
			지방군	과일군	
쇠고기(양지)	40	38	(+)0.2	–	40
돼지고기	40	42	(+)0.1	–	40
쇠 간	40	42		(+)0.1	40
~~닭 간~~	~~40~~	~~43~~	~~(–)0.1~~		~~45~~
~~토끼고기~~	~~40~~	~~37~~	~~(–)0.1~~		~~35~~
뱅 어 포	15	13	(–)0.1	–	15
북 어	15	13	(–)0.3	–	15
건오징어채	15	12	(–)0.2	–	15
잔 멸 치	15	19	(–)0.2	–	20
가 자 미	50	36	(–)0.1	–	35
광 어	50	39	(–)0.3	–	45
동 태	50	50	(–)0.3	–	50
참 돔	50	37	(–)0.3	–	45
~~전 갱 이~~	~~50~~	~~37~~	~~(–)0.1~~		~~35~~
~~물오징어~~	~~50~~	~~41~~	~~(–)0.3~~		~~50~~
참다랑어	50	29	(–)0.3	–	35
~~삼 치~~	~~50~~	~~42~~	~~(+)0.1~~		~~40~~
복 어	50	43	(–)0.3	–	50
조 기	50	44	(–)0.2	–	50
새우(대하)	50	44	(–)0.3	–	50
조개(바지락)	50	62	(–)0.3	–	75
참 전 복	50	53	(–)0.3	–	55
깐 홍 합	50	83	(–)0.2	(+)0.3	80
꽃 게	50	58	(–)0.3	(+)0.1	60
생 굴	80	90	(–)0.2	(+)0.3	90
낙 지	80	69	(–)0.3	–	80
~~굴 비~~	~~15~~	~~18~~	~~(+)0.1~~		~~20~~
칠 면 조	40	34	(–)0.3	–	40
민 어	50	41		–	40
연 어	50	39	(–)0.2	–	40
준 치	50	40		–	40
~~갈 치~~	~~50~~	~~43~~	~~(+)0.2~~		~~40~~
병 어	50	45	(+)0.1	–	45
참치 통조림	40	37	(–)0.2	–	40

[1] 3대영양소를 기준으로 한 ENECC실험값(g) : 1단계
[2] (+) : 기준보다 추가된 식품교환수 ┐1단계
(–) : 기준보다 부족된 식품교환수 ┘
[3] 에너지로 보정한 ENECC/e-식품교환량 : 2단계
pearson 상관계수 : 0.836(P(0.01)에서 유의

[31...	category	Food Name	protein	Food Exchange
554	meat low	beef, meat, domestic, landslide	20.20	39.60
555	meat low	pork, meat, avalanche, raw	22.00	36.36
556	meat low	Beef, cooked by-products, liver, raw	19.00	42.11
557	meat low	Whitefish, Po	60.40	13.25
558	meat low	Pollack, dried fish, adult fish (pollack fish)	61.70	12.97
559	meat low	Cuttlefish (squid), dried	66.80	11.98
560	meat low	anchovy, self-produced, big anchovy	47.40	16.88
561	meat low	flounder, raw, flounder	16.80	47.62
562	meat low	Halibut (flatfish), raw	20.40	39.22
563	meat low	Pollack, frozen product (frozen)	15.90	50.31
564	meat low	Dolphin, red snapper, raw	21.60	37.04
565	meat low	tuna, bluefin tuna, raw, adult fish	27.20	29.41
566	meat low	blowfish, raw, whitefish	15.90	50.31
567	meat low	Early (reference), raw	18.30	43.72
568	meat low	Shrimp, lobster, raw	18.10	44.20
569	meat low	cockle	14.00	57.14
570	meat low	abalone, raw, horse abalone	12.80	62.50
571	meat low	Mussels, raw	9.70	82.47
572	meat low	crab, blue crab, raw	13.70	58.39
573	meat low	oysters, raw, wild oysters, wild oysters	11.60	68.97
574	meat low	Octopus (three-legged octopus)	11.50	69.57
575	meat low	turkey, lean meat, raw	21.80	36.70
576	meat low	croaker, raw	19.70	40.61
577	meat low	salmon, raw	20.60	38.83
578	meat low	level	20.10	39.80
579	meat low	Pomfret fish	17.80	44.94
580	meat low	Tuna, Yellowfin, Canned	21.60	37.04

Final Result	
Differences	%
T(i)-I(i)	{T(i)-I(i)}/T(i)*100
1.6	4.04040404
-5.64	-15.51155116
0.11	0.261220613
0.25	1.886792453
-0.03	-0.231303007
-0.02	-0.166944908
-2.12	-12.55924171
11.62	24.40151197
0.22	0.560938297
0.31	0.616179686
0.04	0.107991361
0.41	1.394083645
7.31	14.52991453
-0.28	-0.640439158
0.2	0.452488688
-4.86	-8.505425271
9.5	15.2
-0.53	-0.642657936
0.39	0.667922589
-21.03	-30.49151805
0.57	0.819318672
2.7	7.356948229
-0.39	-0.960354592
-0.17	-0.43780582
-0.2	-0.502512563
-0.06	-0.133511348
0.04	0.107991361

2.2 MEAT MEDIUM FAT

Table 5. 어육류(중)군 1교환단위의 식품교환량과 ENECC/e-식품교환량

식품	식품 교환량(g)	ENECC 실험값(g)[1]	추가 혹은 부족된 식품 교환수[2]		ENECC/e-식품 교환량(g)[3]
			지방군	과일군	
~~닭고기~~	~~40~~	~~42~~	~~(−)0.1~~	~~−~~	~~40~~
도루묵	50	55	(−)0.2	−	55
청어	50	41	(−)0.3	−	45
햄	40	47	(−)0.1	(+)0.2	45
~~돼지발~~	~~50~~	~~42~~	~~−~~	~~−~~	~~40~~
고등어	50	40	(−)0.2	−	40
꽁치	50	41	(−)0.3		45
임연수어	50	41	(−)0.3	−	45
장어	50	41	−	−	40
검정콩	20	23	(−)0.2	(+)0.5	20
순두부	200	169	(+)0.1	(+)0.1	160
고등어통조림	50	49	−	−	50
두부	80	86	−	(+)0.1	85

[1] 3대영양소를 기준으로 한 ENECC실험값(g) : 1단계
[2] (+) : 기준보다 추가된 식품교환수 ┐1단계
(−) : 기준보다 부족된 식품교환수 ┘
[3] 에너지로 보정한 ENECC/e-식품교환량 : 2단계
pearson 상관계수 : 0.982(P⟨0.01)에서 유의

[34…

	category	Food Name	protein	Food Exchange
581	meat medium	sandfish, raw	14.60	54.79
582	meat medium	herring, raw	10.30	41.45
583	meat medium	Pork, pork products, ham, slices	18.70	42.78
584	meat medium	mackerel, raw	20.20	39.60
585	meat medium	saury, raw	19.50	41.03
586	meat medium	freshwater fish, raw	10.70	40.61
587	meat medium	eel, hagfish (spike eel)	16.60	48.19
588	meat medium	Soybean, black bean, dried, black bean	35.20	22.73
589	meat medium	soybean, tofu, silken tofu	4.70	170.21
590	meat medium	mackerel, canned food	16.30	49.08
591	meat medium	soybean, tofu, tofu	9.30	86.02

Final Result	
Differences	%
T(i)-I(i)	{T(i)-I(i)}/T(i)*100
-0.21	-0.383281621
0.45	1.085645356
-4.22	-9.864422627
-0.4	-1.01010101
0.03	0.073117231
-0.39	-0.960354592
7.19	14.92010791
-0.27	-1.187857457
1.21	0.710886552
0.08	0.162999185
0.02	0.023250407

2.3 MEAT HIGH FAT

Table 6. 어육류(고)군 1교환단위의 식품교환량과 ENECC/e-식품교환량

식 품	식품 교환량(g)	ENECC 실험값(g)[1]	추가 혹은 부족된 식품 교환수[a] 지방군	추가 혹은 부족된 식품 교환수[a] 과일군	ENECC/e-식품 교환량(g)[3]
개고기	40	42	(+0)0.1	–	40
메추리알	40	65	(−)0.1	(+)0.1	65
~~달 걀~~	~~50~~	~~64~~	~~(−)0.2~~	~~–~~	~~65~~
~~치 즈~~	~~30~~	~~30~~	~~(−)0.2~~	~~–~~	~~30~~
유 부	20	39	(+)0.8	(+)0.9	25
쇠갈비	30	43	(+0)0.2	–	45
쇠꼬리	40	46	(+)0.1	–	45
우 설	40	62	(+)0.2	–	60
프랑크소시지	40	56	(+0)0.8	(+)0.2	35
꽁치 통조림	50	57	(+)0.6	–	45
뱀장어	50	56	(+)0.3	–	55
~~런천민트~~	~~40~~	~~64~~	~~(+)2.3~~	~~(+)0.1~~	~~30~~

[1] 3대영양소를 기준으로 한 ENECC실험값(g) : 1단계
[a] (+) : 기준보다 추가된 식품교환수
(−) : 기준보다 부족된 식품교환수] 1단계
[3] 에너지로 보정한 ENECC/e-식품교환량 : 2단계
pearson 상관계수 : 0.621(P〈0.05)에서 유의

[38...	category	Food Name	protein	Food Exchange
592	meat high	dog meat	19.00	42.11
593	meat high	Quail eggs, whole eggs, raw	12.60	63.49
594	meat high	Soybean, Tofu, Fried Tofu (Tofu)	20.40	39.22
595	meat high	Beef, meat, imported, ribs, raw	18.50	43.24
596	meat high	beef, beef by-products, tail	17.40	45.98
597	meat high	Pork, pork products, sausages, frankfurt	14.40	55.56
598	meat high	saury, canned food	14.00	57.14
599	meat high	Eel, eel (device), raw	14.40	55.56
600	meat high	Pork, pork products, ham, luncheon meat	15.00	53.33

Final Result	
Differences	%
T(i)-I(i)	{T(i)-I(i)}/T(i)*100
0.11	0.261904762
-1.51	-2.323076923
0.22	0.564102564
0.24	0.558139535
-0.02	-0.043478261
-6.44	-10.38709677
1.14	2.035714286
-1.44	-2.526315789
-2.67	-4.767857143

THANK YOU