

# CIA

Raghul V\_21122048

24/03/2022

## Question A:

```
df = read.csv("/Users/raghul/Downloads/Q2.csv")
```

### Importing the dataset

```
head(df, 15)
```

### Load the dataset and display the first 15 rows of the dataset

##	Name.A1.S1	Manuf	Type	Calories	Protein	Fat	Sodium	Fiber	Carbo
## 1	100%_Bran	N	C	70	4	1	130	10.0	5.0
## 2	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0
## 3	All-Bran	K	C	70	4	1	260	9.0	7.0
## 4	All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14.0	8.0
## 5	Almond_Delight	R	C	110	2	2	200	1.0	14.0
## 6	Apple_Cinnamon_Cheerios	G	C	110	2	2	180	1.5	10.5
## 7	Apple_Jacks	K	C	110	2	0	125	1.0	11.0
## 8	Basic_4	G	C	130	3	2	210	2.0	18.0
## 9	Bran_Chex	R	C	90	2	1	200	4.0	15.0
## 10	Bran_Flakes	P	C	90	3	0	210	5.0	13.0
## 11	Cap'n'Crunch	Q	C	120	1	2	220	0.0	12.0
## 12	Cheerios	G	C	110	6	2	290	2.0	17.0
## 13	Cinnamon_Toast_Crunch	G	C	120	1	3	210	0.0	13.0
## 14	Clusters	G	C	110	3	2	140	2.0	13.0
## 15	Cocoa_Puffs	G	C	110	1	1	180	0.0	12.0

##	Sugars	Potass	Vitamins	Shelf	Weight	Cups	Rating	Cold	Nabisco	Quaker
## 1	6	280	25	3	1.00	0.33	68.40297	1	1	0
## 2	8	135	0	3	1.00	1.00	33.98368	1	0	1
## 3	5	320	25	3	1.00	0.33	59.42551	1	0	0
## 4	0	330	25	3	1.00	0.50	93.70491	1	0	0
## 5	8	NA	25	3	1.00	0.75	34.38484	1	0	0
## 6	10	70	25	1	1.00	0.75	29.50954	1	0	0
## 7	14	30	25	2	1.00	1.00	33.17409	1	0	0
## 8	8	100	25	3	1.33	0.75	37.03856	1	0	0
## 9	6	125	25	1	1.00	0.67	49.12025	1	0	0

```
## 10      5      190      25      3      1.00 0.67 53.31381      1      0      0
## 11     12       35      25      2      1.00 0.75 18.04285      1      0      1
## 12      1     105      25      1      1.00 1.25 50.76500      1      0      0
## 13      9      45      25      2      1.00 0.75 19.82357      1      0      0
## 14      7     105      25      3      1.00 0.50 40.40021      1      0      0
## 15     13      55      25      2      1.00 1.00 22.73645      1      0      0
##      Kelloggs GeneralMills Ralston AHFP
## 1          0          0          0      0
## 2          0          0          0      0
## 3          1          0          0      0
## 4          1          0          0      0
## 5          0          0          1      0
## 6          0          1          0      0
## 7          1          0          0      0
## 8          0          1          0      0
## 9          0          0          1      0
## 10         0          0          0      0
## 11         0          0          0      0
## 12         0          1          0      0
## 13         0          1          0      0
## 14         0          1          0      0
## 15         0          1          0      0
```

## A. Exploratory Data Analysis (Use appropriate plots and summary to study the data)

```
summary(df)
```

### Summarizing the data

```
##      Name.A1.S1      Manuf      Type      Calories
## Length:77      Length:77      Length:77      Min.   : 50.0
## Class :character Class :character Class :character 1st Qu.:100.0
## Mode  :character Mode  :character Mode  :character Median :110.0
##                                         Mean  :106.9
##                                         3rd Qu.:110.0
##                                         Max.   :160.0
##
##      Protein      Fat      Sodium      Fiber
## Min.   :1.000   Min.   :0.000   Min.   : 0.0   Min.   : 0.000
## 1st Qu.:2.000   1st Qu.:0.000   1st Qu.:130.0   1st Qu.: 1.000
## Median :3.000   Median :1.000   Median :180.0   Median : 2.000
## Mean   :2.545   Mean   :1.013   Mean   :159.7   Mean   : 2.152
## 3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:210.0   3rd Qu.: 3.000
## Max.   :6.000   Max.   :5.000   Max.   :320.0   Max.   :14.000
##
##      Carbo      Sugars      Potass      Vitamins
## Min.   : 5.0   Min.   : 0.000   Min.   : 15.00   Min.   : 0.00
## 1st Qu.:12.0   1st Qu.: 3.000   1st Qu.: 42.50   1st Qu.: 25.00
## Median :14.5   Median : 7.000   Median : 90.00   Median : 25.00
## Mean   :14.8   Mean   : 7.026   Mean   : 98.67   Mean   : 28.25
```

```
## 3rd Qu.:17.0 3rd Qu.:11.000 3rd Qu.:120.00 3rd Qu.: 25.00
## Max. :23.0 Max. :15.000 Max. :330.00 Max. :100.00
## NA's :1 NA's :1 NA's :2
## Shelf Weight Cups Rating Cold
## Min. :1.000 Min. :0.50 Min. :0.250 Min. :18.04 Min. :0.000
## 1st Qu.:1.000 1st Qu.:1.00 1st Qu.:0.670 1st Qu.:33.17 1st Qu.:1.000
## Median :2.000 Median :1.00 Median :0.750 Median :40.40 Median :1.000
## Mean :2.208 Mean :1.03 Mean :0.821 Mean :42.67 Mean :0.961
## 3rd Qu.:3.000 3rd Qu.:1.00 3rd Qu.:1.000 3rd Qu.:50.83 3rd Qu.:1.000
## Max. :3.000 Max. :1.50 Max. :1.500 Max. :93.70 Max. :1.000
##
## Nabisco Quaker Kelloggs GeneralMills
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000
## Median :0.00000 Median :0.0000 Median :0.0000 Median :0.0000
## Mean :0.07792 Mean :0.1039 Mean :0.2987 Mean :0.2857
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000
##
## Ralston AHFP
## Min. :0.0000 Min. :0.00000
## 1st Qu.:0.0000 1st Qu.:0.00000
## Median :0.0000 Median :0.00000
## Mean :0.1039 Mean :0.01299
## 3rd Qu.:0.0000 3rd Qu.:0.00000
## Max. :1.0000 Max. :1.00000
##
```

```
dim(df)
```

Dimension of the dataset (Total no.of.rows & Total no.of columns)

```
## [1] 77 23
```

```
str(df)
```

```
## 'data.frame': 77 obs. of 23 variables:
## $ Name.A1.S1 : chr "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-Bran_with_Extra_Fiber" ...
## $ Manuf : chr "N" "Q" "K" "K" ...
## $ Type : chr "C" "C" "C" "C" ...
## $ Calories : int 70 120 70 50 110 110 110 130 90 90 ...
## $ Protein : int 4 3 4 4 2 2 2 3 2 3 ...
## $ Fat : int 1 5 1 0 2 2 0 2 1 0 ...
## $ Sodium : int 130 15 260 140 200 180 125 210 200 210 ...
## $ Fiber : num 10 2 9 14 1 1.5 1 2 4 5 ...
## $ Carbo : num 5 8 7 8 14 10.5 11 18 15 13 ...
## $ Sugars : int 6 8 5 0 8 10 14 8 6 5 ...
## $ Potass : int 280 135 320 330 NA 70 30 100 125 190 ...
## $ Vitamins : int 25 0 25 25 25 25 25 25 25 25 ...
## $ Shelf : int 3 3 3 3 3 1 2 3 1 3 ...
```

```
## $ Weight      : num  1 1 1 1 1 1 1 1.33 1 1 ...
## $ Cups        : num  0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
## $ Rating      : num  68.4 34 59.4 93.7 34.4 ...
## $ Cold        : int   1 1 1 1 1 1 1 1 1 1 ...
## $ Nabisco     : int   1 0 0 0 0 0 0 0 0 0 ...
## $ Quaker      : int   0 1 0 0 0 0 0 0 0 0 ...
## $ Kelloggs    : int   0 0 1 1 0 0 1 0 0 0 ...
## $ GeneralMills: int   0 0 0 0 0 1 0 1 0 0 ...
## $ Ralston     : int   0 0 0 0 1 0 0 0 1 0 ...
## $ AHFP        : int   0 0 0 0 0 0 0 0 0 0 ...
```

```
nrow(df)
```

```
## [1] 77
```

```
ncol(df)
```

```
## [1] 23
```

```
is.na(df)
```

```
##      Name.A1.S1 Manuf  Type Calories Protein  Fat Sodium Fiber Carbo Sugars
## [1,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [2,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [3,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [4,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [5,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [6,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [7,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [8,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [9,]      FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [10,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [11,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [12,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [13,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [14,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [15,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [16,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [17,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [18,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [19,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [20,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [21,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [22,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [23,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [24,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [25,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [26,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [27,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [28,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [29,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
## [30,]     FALSE FALSE FALSE      FALSE      FALSE FALSE FALSE FALSE FALSE FALSE
```

[illegible]

[illegible]

##	[61,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[62,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[63,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[64,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[65,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[66,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[67,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[68,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[69,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[70,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[71,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[72,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[73,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[74,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[75,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[76,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	[77,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	GeneralMills Ralston AHFP									
##	[1,]	FALSE	FALSE	FALSE						
##	[2,]	FALSE	FALSE	FALSE						
##	[3,]	FALSE	FALSE	FALSE						
##	[4,]	FALSE	FALSE	FALSE						
##	[5,]	FALSE	FALSE	FALSE						
##	[6,]	FALSE	FALSE	FALSE						
##	[7,]	FALSE	FALSE	FALSE						
##	[8,]	FALSE	FALSE	FALSE						
##	[9,]	FALSE	FALSE	FALSE						
##	[10,]	FALSE	FALSE	FALSE						
##	[11,]	FALSE	FALSE	FALSE						
##	[12,]	FALSE	FALSE	FALSE						
##	[13,]	FALSE	FALSE	FALSE						
##	[14,]	FALSE	FALSE	FALSE						
##	[15,]	FALSE	FALSE	FALSE						
##	[16,]	FALSE	FALSE	FALSE						
##	[17,]	FALSE	FALSE	FALSE						
##	[18,]	FALSE	FALSE	FALSE						
##	[19,]	FALSE	FALSE	FALSE						
##	[20,]	FALSE	FALSE	FALSE						
##	[21,]	FALSE	FALSE	FALSE						
##	[22,]	FALSE	FALSE	FALSE						
##	[23,]	FALSE	FALSE	FALSE						
##	[24,]	FALSE	FALSE	FALSE						
##	[25,]	FALSE	FALSE	FALSE						
##	[26,]	FALSE	FALSE	FALSE						
##	[27,]	FALSE	FALSE	FALSE						
##	[28,]	FALSE	FALSE	FALSE						
##	[29,]	FALSE	FALSE	FALSE						
##	[30,]	FALSE	FALSE	FALSE						
##	[31,]	FALSE	FALSE	FALSE						
##	[32,]	FALSE	FALSE	FALSE						
##	[33,]	FALSE	FALSE	FALSE						
##	[34,]	FALSE	FALSE	FALSE						
##	[35,]	FALSE	FALSE	FALSE						
##	[36,]	FALSE	FALSE	FALSE						

```
## [37,]      FALSE      FALSE FALSE
## [38,]      FALSE      FALSE FALSE
## [39,]      FALSE      FALSE FALSE
## [40,]      FALSE      FALSE FALSE
## [41,]      FALSE      FALSE FALSE
## [42,]      FALSE      FALSE FALSE
## [43,]      FALSE      FALSE FALSE
## [44,]      FALSE      FALSE FALSE
## [45,]      FALSE      FALSE FALSE
## [46,]      FALSE      FALSE FALSE
## [47,]      FALSE      FALSE FALSE
## [48,]      FALSE      FALSE FALSE
## [49,]      FALSE      FALSE FALSE
## [50,]      FALSE      FALSE FALSE
## [51,]      FALSE      FALSE FALSE
## [52,]      FALSE      FALSE FALSE
## [53,]      FALSE      FALSE FALSE
## [54,]      FALSE      FALSE FALSE
## [55,]      FALSE      FALSE FALSE
## [56,]      FALSE      FALSE FALSE
## [57,]      FALSE      FALSE FALSE
## [58,]      FALSE      FALSE FALSE
## [59,]      FALSE      FALSE FALSE
## [60,]      FALSE      FALSE FALSE
## [61,]      FALSE      FALSE FALSE
## [62,]      FALSE      FALSE FALSE
## [63,]      FALSE      FALSE FALSE
## [64,]      FALSE      FALSE FALSE
## [65,]      FALSE      FALSE FALSE
## [66,]      FALSE      FALSE FALSE
## [67,]      FALSE      FALSE FALSE
## [68,]      FALSE      FALSE FALSE
## [69,]      FALSE      FALSE FALSE
## [70,]      FALSE      FALSE FALSE
## [71,]      FALSE      FALSE FALSE
## [72,]      FALSE      FALSE FALSE
## [73,]      FALSE      FALSE FALSE
## [74,]      FALSE      FALSE FALSE
## [75,]      FALSE      FALSE FALSE
## [76,]      FALSE      FALSE FALSE
## [77,]      FALSE      FALSE FALSE
```

## Importing required Libraries

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v tibble  3.1.4      v dplyr    1.0.7
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   2.0.1      v forcats 0.5.1
```



```
## v purrr 0.3.4
```

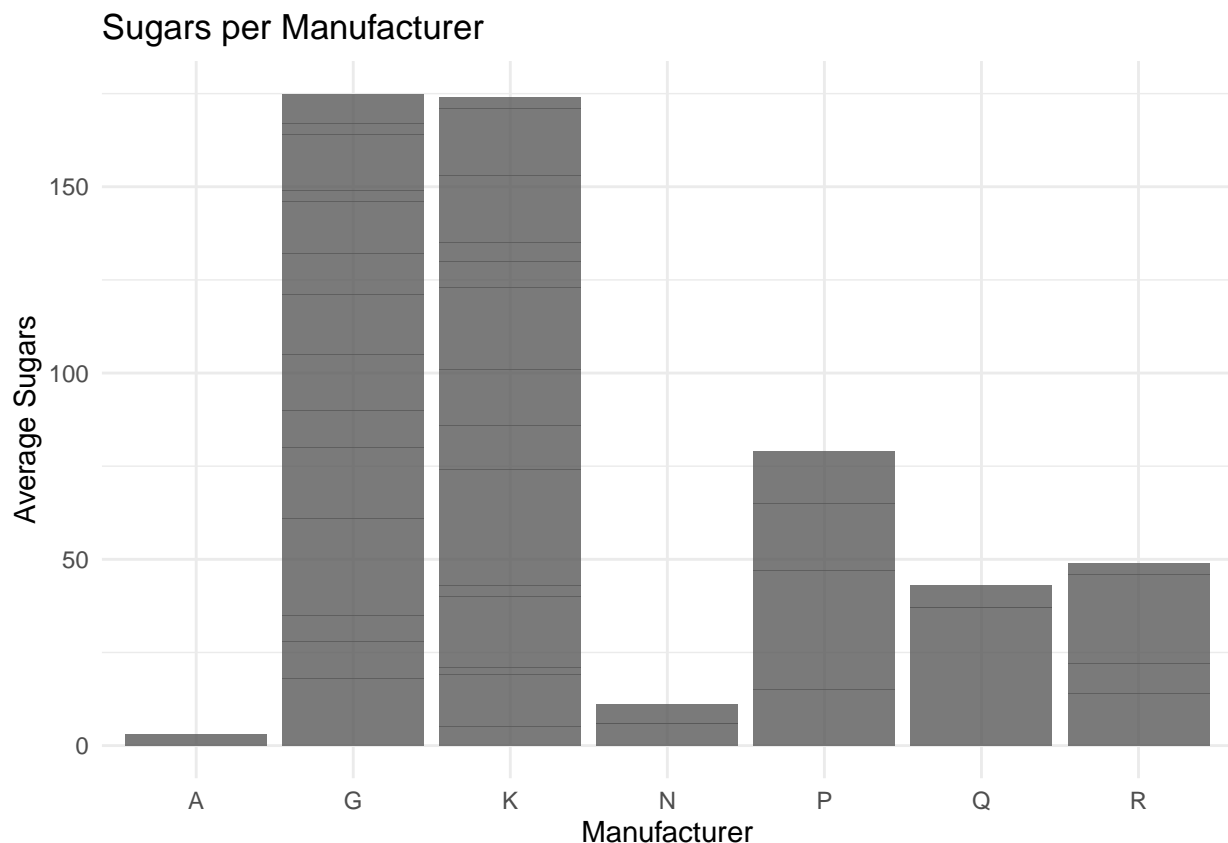
```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag() masks stats::lag()
```

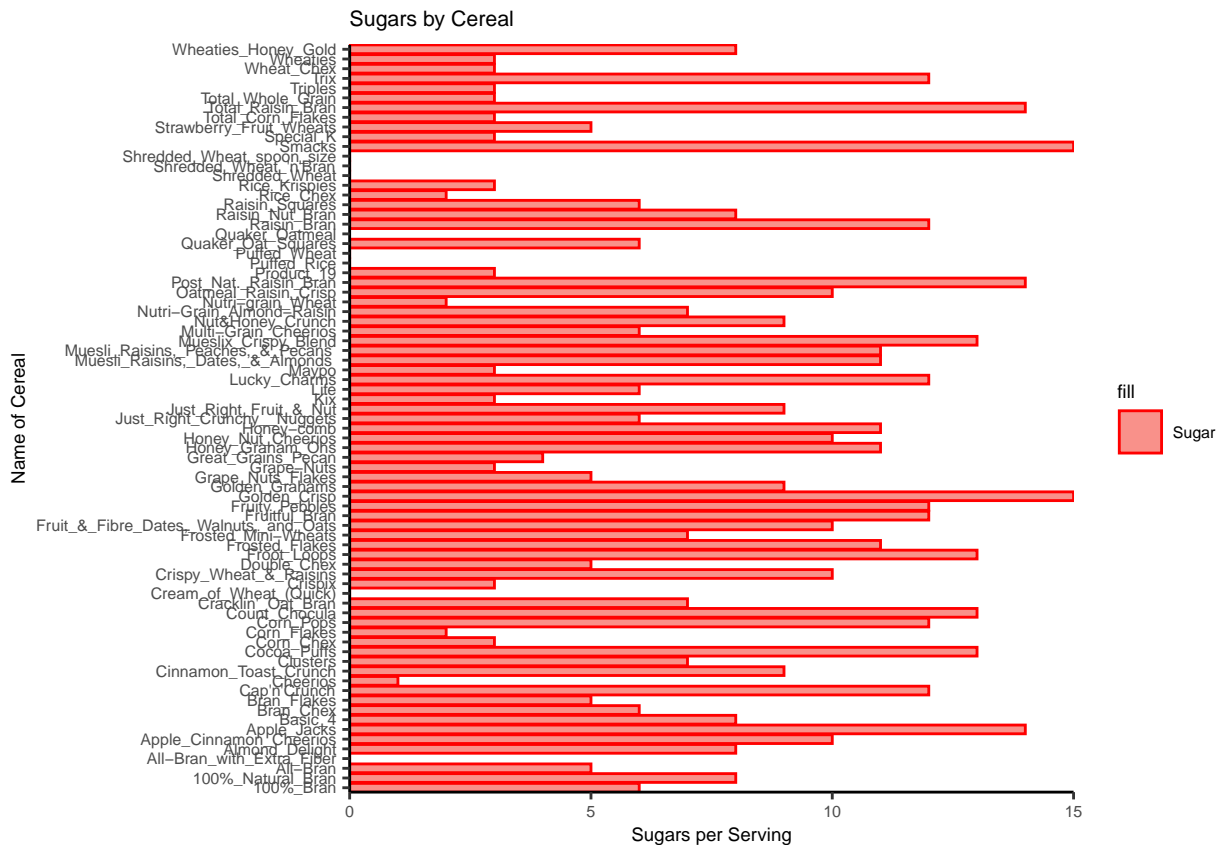
```
ggplot(df) +  
  aes(x = Manuf, y = as.integer(Sugars)) +  
  geom_bar(stat = "identity", alpha = 0.8) +  
  labs(title = "Sugars per Manufacturer",  
        y = "Average Sugars",  
        x = "Manufacturer") +  
  theme_minimal()
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```



```
ggplot(df, mapping = aes(x = Name.A1.S1 , y = as.numeric(Sugars), fill = "Sugar")) +  
  geom_bar( stat = "identity", alpha = .8, color = "red") +  
  theme_classic() +  
  theme(text = element_text(size=7)) +  
  labs(title = "Sugars by Cereal",  
        x = "Name of Cereal",  
        y = "Sugars per Serving" ) +  
  coord_flip(xlim = NULL, ylim = NULL, expand = FALSE, clip = "on")
```

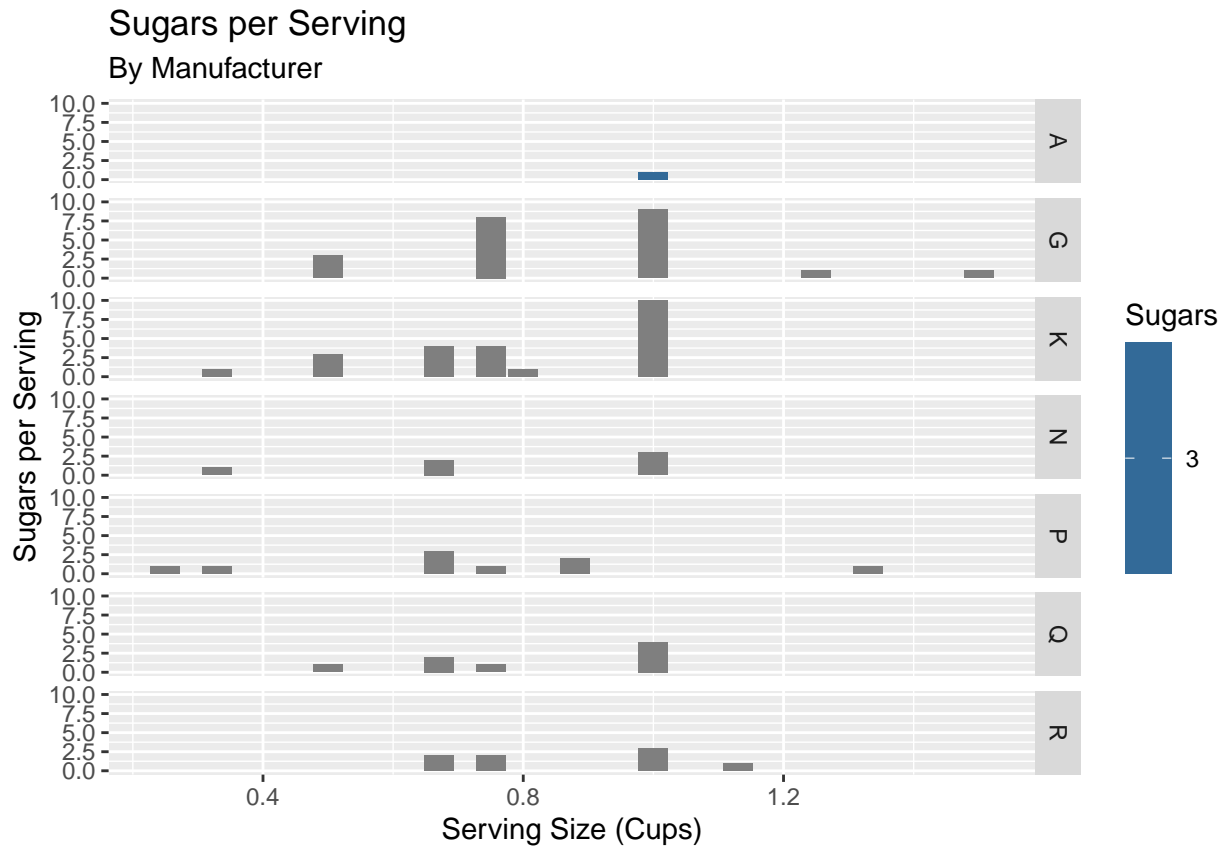
```
## Warning: Removed 1 rows containing missing values (position_stack).
```



#### We can observe that “Smacks” and “Golden crisp” are the products with high sugar content.

### Sugar Per Serving

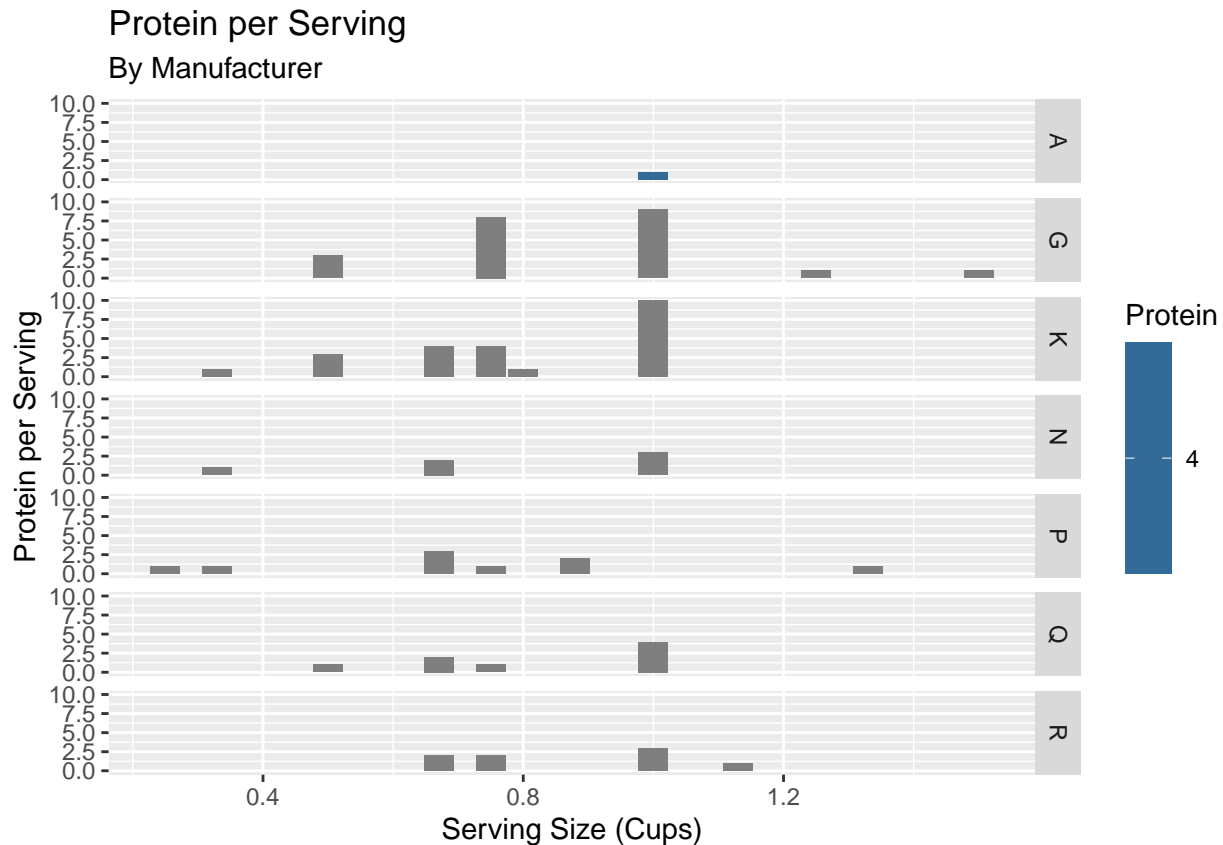
```
ggplot(df) +
  aes(x = Cups, fill = Sugars) +
  geom_bar() +
  facet_grid(Manuf~.) +
  labs(title = "Sugars per Serving",
        subtitle = "By Manufacturer",
        x = "Serving Size (Cups)",
        y = "Sugars per Serving" )
```



#### We can observe that then “Kellogg” and “General Mills” have the most amount of sugars in relation to their serving sizes.

### Protein Per Serving

```
ggplot(df) +
  aes(x = Cups, fill = Protein) +
  geom_bar(position = 'dodge') +
  facet_grid(Manuf~.) +
  labs(title = "Protein per Serving",
        subtitle = "By Manufacturer",
        x = "Serving Size (Cups)",
        y = "Protein per Serving" )
```



#### We can observe that General Mills has a wide variance in the protein content for their cereals.

### Question B:

1. List the names of the items with maximum calories

```
cal=df[order(df$Calories, decreasing=TRUE), ]
cal
```

##	Name.A1.S1	Manuf	Type	Calories	Protein	Fat
## 47	Mueslix_Crispy_Blend	K	C	160	3	2
## 45	Muesli_Raisins,_Dates,_&_Almonds	R	C	150	4	3
## 46	Muesli_Raisins,_Peaches,_&_Pecans	R	C	150	4	3
## 40	Just_Right_Fruit_&_Nut	K	C	140	3	1
## 50	Nutri-Grain_Almond-Raisin	K	C	140	3	2
## 71	Total_Raisin_Bran	G	C	140	3	1
## 8	Basic_4	G	C	130	3	2
## 52	Oatmeal_Raisin_Crisp	G	C	130	3	2
## 2	100%_Natural_Bran	Q	C	120	3	5
## 11	Cap'n'Crunch	Q	C	120	1	2
## 13	Cinnamon_Toast_Crunch	G	C	120	1	3
## 28	Fruit_&_Fibre_Dates,_Walnuts,_and_Oats	P	C	120	3	2
## 29	Fruitful_Bran	K	C	120	3	0
## 35	Great_Grains_Pecan	P	C	120	3	3
## 36	Honey_Graham_Ohs	Q	C	120	1	2

## 49	Nut&Honey_Crunch	K	C	120	2	1
## 53	Post_Nat._Raisin_Bran	P	C	120	3	1
## 59	Raisin_Bran	K	C	120	3	1
## 5	Almond_Delight	R	C	110	2	2
## 6	Apple_Cinnamon_Cheerios	G	C	110	2	2
## 7	Apple_Jacks	K	C	110	2	0
## 12	Cheerios	G	C	110	6	2
## 14	Clusters	G	C	110	3	2
## 15	Cocoa_Puffs	G	C	110	1	1
## 16	Corn_Chex	R	C	110	2	0
## 18	Corn_Pops	K	C	110	1	0
## 19	Count_Chocula	G	C	110	1	1
## 20	Cracklin'_Oat_Bran	K	C	110	3	3
## 22	Crispix	K	C	110	2	0
## 25	Froot_Loops	K	C	110	2	1
## 26	Frosted_Flakes	K	C	110	1	0
## 30	Fruity_Pebbles	P	C	110	1	1
## 32	Golden_Grahams	G	C	110	1	1
## 34	Grape-Nuts	P	C	110	3	0
## 37	Honey_Nut_Cheerios	G	C	110	3	1
## 38	Honey-comb	P	C	110	1	0
## 39	Just_Right_Crunchy__Nuggets	K	C	110	2	1
## 41	Kix	G	C	110	2	1
## 43	Lucky_Charms	G	C	110	2	1
## 62	Rice_Chex	R	C	110	1	0
## 63	Rice_Krispies	K	C	110	2	0
## 67	Smacks	K	C	110	2	1
## 68	Special_K	K	C	110	6	0
## 70	Total_Corn_Flakes	G	C	110	2	1
## 73	Triples	G	C	110	2	1
## 74	Trix	G	C	110	1	1
## 77	Wheaties_Honey_Gold	G	C	110	2	1
## 17	Corn_Flakes	K	C	100	2	0
## 21	Cream_of_Wheat_(Quick)	N	H	100	3	0
## 23	Crispy_Wheat_&_Raisins	G	C	100	2	1
## 24	Double_Chex	R	C	100	2	0
## 27	Frosted_Mini-Wheats	K	C	100	3	0
## 31	Golden_Crisp	P	C	100	2	0
## 33	Grape_Nuts_Flakes	P	C	100	3	1
## 42	Life	Q	C	100	4	2
## 44	Maypo	A	H	100	4	1
## 48	Multi-Grain_Cheerios	G	C	100	2	1
## 54	Product_19	K	C	100	3	0
## 57	Quaker_Oat_Squares	Q	C	100	4	1
## 58	Quaker_Oatmeal	Q	H	100	5	2
## 60	Raisin_Nut_Bran	G	C	100	3	2
## 72	Total_Whole_Grain	G	C	100	3	1
## 75	Wheat_Chex	R	C	100	3	1
## 76	Wheaties	G	C	100	3	1
## 9	Bran_Chex	R	C	90	2	1
## 10	Bran_Flakes	P	C	90	3	0
## 51	Nutri-grain_Wheat	K	C	90	3	0
## 61	Raisin_Squares	K	C	90	2	0
## 65	Shredded_Wheat_'n'Bran	N	C	90	3	0

## 66	Shredded_Wheat_spoon_size	N	C	90	3	0
## 69	Strawberry_Fruit_Wheats	N	C	90	2	0
## 64	Shredded_Wheat	N	C	80	2	0
## 1	100%_Bran	N	C	70	4	1
## 3	All-Bran	K	C	70	4	1
## 4	All-Bran_with_Extra_Fiber	K	C	50	4	0
## 55	Puffed_Rice	Q	C	50	1	0
## 56	Puffed_Wheat	Q	C	50	2	0
##	Sodium Fiber Carbo Sugars Potass Vitamins Shelf Weight Cups Rating Cold					
## 47	150 3.0 17.0 13 160 25 3 1.50 0.67 30.31335 1					
## 45	95 3.0 16.0 11 170 25 3 1.00 1.00 37.13686 1					
## 46	150 3.0 16.0 11 170 25 3 1.00 1.00 34.13976 1					
## 40	170 2.0 20.0 9 95 100 3 1.30 0.75 36.47151 1					
## 50	220 3.0 21.0 7 130 25 3 1.33 0.67 40.69232 1					
## 71	190 4.0 15.0 14 230 100 3 1.50 1.00 28.59278 1					
## 8	210 2.0 18.0 8 100 25 3 1.33 0.75 37.03856 1					
## 52	170 1.5 13.5 10 120 25 3 1.25 0.50 30.45084 1					
## 2	15 2.0 8.0 8 135 0 3 1.00 1.00 33.98368 1					
## 11	220 0.0 12.0 12 35 25 2 1.00 0.75 18.04285 1					
## 13	210 0.0 13.0 9 45 25 2 1.00 0.75 19.82357 1					
## 28	160 5.0 12.0 10 200 25 3 1.25 0.67 40.91705 1					
## 29	240 5.0 14.0 12 190 25 3 1.33 0.67 41.01549 1					
## 35	75 3.0 13.0 4 100 25 3 1.00 0.33 45.81172 1					
## 36	220 1.0 12.0 11 45 25 2 1.00 1.00 21.87129 1					
## 49	190 0.0 15.0 9 40 25 2 1.00 0.67 29.92429 1					
## 53	200 6.0 11.0 14 260 25 3 1.33 0.67 37.84059 1					
## 59	210 5.0 14.0 12 240 25 2 1.33 0.75 39.25920 1					
## 5	200 1.0 14.0 8 NA 25 3 1.00 0.75 34.38484 1					
## 6	180 1.5 10.5 10 70 25 1 1.00 0.75 29.50954 1					
## 7	125 1.0 11.0 14 30 25 2 1.00 1.00 33.17409 1					
## 12	290 2.0 17.0 1 105 25 1 1.00 1.25 50.76500 1					
## 14	140 2.0 13.0 7 105 25 3 1.00 0.50 40.40021 1					
## 15	180 0.0 12.0 13 55 25 2 1.00 1.00 22.73645 1					
## 16	280 0.0 22.0 3 25 25 1 1.00 1.00 41.44502 1					
## 18	90 1.0 13.0 12 20 25 2 1.00 1.00 35.78279 1					
## 19	180 0.0 12.0 13 65 25 2 1.00 1.00 22.39651 1					
## 20	140 4.0 10.0 7 160 25 3 1.00 0.50 40.44877 1					
## 22	220 1.0 21.0 3 30 25 3 1.00 1.00 46.89564 1					
## 25	125 1.0 11.0 13 30 25 2 1.00 1.00 32.20758 1					
## 26	200 1.0 14.0 11 25 25 1 1.00 0.75 31.43597 1					
## 30	135 0.0 13.0 12 25 25 2 1.00 0.75 28.02576 1					
## 32	280 0.0 15.0 9 45 25 2 1.00 0.75 23.80404 1					
## 34	170 3.0 17.0 3 90 25 3 1.00 0.25 53.37101 1					
## 37	250 1.5 11.5 10 90 25 1 1.00 0.75 31.07222 1					
## 38	180 0.0 14.0 11 35 25 1 1.00 1.33 28.74241 1					
## 39	170 1.0 17.0 6 60 100 3 1.00 1.00 36.52368 1					
## 41	260 0.0 21.0 3 40 25 2 1.00 1.50 39.24111 1					
## 43	180 0.0 12.0 12 55 25 2 1.00 1.00 26.73451 1					
## 62	240 0.0 23.0 2 30 25 1 1.00 1.13 41.99893 1					
## 63	290 0.0 22.0 3 35 25 1 1.00 1.00 40.56016 1					
## 67	70 1.0 9.0 15 40 25 2 1.00 0.75 31.23005 1					
## 68	230 1.0 16.0 3 55 25 1 1.00 1.00 53.13132 1					
## 70	200 0.0 21.0 3 35 100 3 1.00 1.00 38.83975 1					
## 73	250 0.0 21.0 3 60 25 3 1.00 0.75 39.10617 1					

## 74	140	0.0	13.0	12	25	25	2	1.00	1.00	27.75330	1
## 77	200	1.0	16.0	8	60	25	1	1.00	0.75	36.18756	1
## 17	290	1.0	21.0	2	35	25	1	1.00	1.00	45.86332	1
## 21	80	1.0	21.0	0	NA	0	2	1.00	1.00	64.53382	0
## 23	140	2.0	11.0	10	120	25	3	1.00	0.75	36.17620	1
## 24	190	1.0	18.0	5	80	25	3	1.00	0.75	44.33086	1
## 27	0	3.0	14.0	7	100	25	2	1.00	0.80	58.34514	1
## 31	45	0.0	11.0	15	40	25	1	1.00	0.88	35.25244	1
## 33	140	3.0	15.0	5	85	25	3	1.00	0.88	52.07690	1
## 42	150	2.0	12.0	6	95	25	2	1.00	0.67	45.32807	1
## 44	0	0.0	16.0	3	95	25	2	1.00	1.00	54.85092	0
## 48	220	2.0	15.0	6	90	25	1	1.00	1.00	40.10596	1
## 54	320	1.0	20.0	3	45	100	3	1.00	1.00	41.50354	1
## 57	135	2.0	14.0	6	110	25	3	1.00	0.50	49.51187	1
## 58	0	2.7	NA	NA	110	0	1	1.00	0.67	50.82839	0
## 60	140	2.5	10.5	8	140	25	3	1.00	0.50	39.70340	1
## 72	200	3.0	16.0	3	110	100	3	1.00	1.00	46.65884	1
## 75	230	3.0	17.0	3	115	25	1	1.00	0.67	49.78744	1
## 76	200	3.0	17.0	3	110	25	1	1.00	1.00	51.59219	1
## 9	200	4.0	15.0	6	125	25	1	1.00	0.67	49.12025	1
## 10	210	5.0	13.0	5	190	25	3	1.00	0.67	53.31381	1
## 51	170	3.0	18.0	2	90	25	3	1.00	1.00	59.64284	1
## 61	0	2.0	15.0	6	110	25	3	1.00	0.50	55.33314	1
## 65	0	4.0	19.0	0	140	0	1	1.00	0.67	74.47295	1
## 66	0	3.0	20.0	0	120	0	1	1.00	0.67	72.80179	1
## 69	15	3.0	15.0	5	90	25	2	1.00	1.00	59.36399	1
## 64	0	3.0	16.0	0	95	0	1	0.83	1.00	68.23588	1
## 1	130	10.0	5.0	6	280	25	3	1.00	0.33	68.40297	1
## 3	260	9.0	7.0	5	320	25	3	1.00	0.33	59.42551	1
## 4	140	14.0	8.0	0	330	25	3	1.00	0.50	93.70491	1
## 55	0	0.0	13.0	0	15	0	3	0.50	1.00	60.75611	1
## 56	0	1.0	10.0	0	50	0	3	0.50	1.00	63.00565	1
##	Nabisco	Quaker	Kelloggs	GeneralMills	Ralston	AHFP					
## 47	0	0	1		0	0	0				
## 45	0	0	0		0	1	0				
## 46	0	0	0		0	1	0				
## 40	0	0	1		0	0	0				
## 50	0	0	1		0	0	0				
## 71	0	0	0		1	0	0				
## 8	0	0	0		1	0	0				
## 52	0	0	0		1	0	0				
## 2	0	1	0		0	0	0				
## 11	0	1	0		0	0	0				
## 13	0	0	0		1	0	0				
## 28	0	0	0		0	0	0				
## 29	0	0	1		0	0	0				
## 35	0	0	0		0	0	0				
## 36	0	1	0		0	0	0				
## 49	0	0	1		0	0	0				
## 53	0	0	0		0	0	0				
## 59	0	0	1		0	0	0				
## 5	0	0	0		0	1	0				
## 6	0	0	0		1	0	0				
## 7	0	0	1		0	0	0				

## 12	0	0	0	1	0	0
## 14	0	0	0	1	0	0
## 15	0	0	0	1	0	0
## 16	0	0	0	0	1	0
## 18	0	0	1	0	0	0
## 19	0	0	0	1	0	0
## 20	0	0	1	0	0	0
## 22	0	0	1	0	0	0
## 25	0	0	1	0	0	0
## 26	0	0	1	0	0	0
## 30	0	0	0	0	0	0
## 32	0	0	0	1	0	0
## 34	0	0	0	0	0	0
## 37	0	0	0	1	0	0
## 38	0	0	0	0	0	0
## 39	0	0	1	0	0	0
## 41	0	0	0	1	0	0
## 43	0	0	0	1	0	0
## 62	0	0	0	0	1	0
## 63	0	0	1	0	0	0
## 67	0	0	1	0	0	0
## 68	0	0	1	0	0	0
## 70	0	0	0	1	0	0
## 73	0	0	0	1	0	0
## 74	0	0	0	1	0	0
## 77	0	0	0	1	0	0
## 17	0	0	1	0	0	0
## 21	1	0	0	0	0	0
## 23	0	0	0	1	0	0
## 24	0	0	0	0	1	0
## 27	0	0	1	0	0	0
## 31	0	0	0	0	0	0
## 33	0	0	0	0	0	0
## 42	0	1	0	0	0	0
## 44	0	0	0	0	0	1
## 48	0	0	0	1	0	0
## 54	0	0	1	0	0	0
## 57	0	1	0	0	0	0
## 58	0	1	0	0	0	0
## 60	0	0	0	1	0	0
## 72	0	0	0	1	0	0
## 75	0	0	0	0	1	0
## 76	0	0	0	1	0	0
## 9	0	0	0	0	1	0
## 10	0	0	0	0	0	0
## 51	0	0	1	0	0	0
## 61	0	0	1	0	0	0
## 65	1	0	0	0	0	0
## 66	1	0	0	0	0	0
## 69	1	0	0	0	0	0
## 64	1	0	0	0	0	0
## 1	1	0	0	0	0	0
## 3	0	0	1	0	0	0
## 4	0	0	1	0	0	0



```
## 55      0      1      0      0      0      0
## 56      0      1      0      0      0      0
```

```
dfcal <-df[order(df$Calories, decreasing = TRUE),]
```

```
head(dfcal)
```

## 2. Arrange and sort the data in descending order of calories

```
##              Name.A1.S1 Manuf Type Calories Protein Fat Sodium
## 47      Mueslix_Crispy_Blend      K   C      160        3   2    150
## 45  Muesli_Raisins,_Dates,_&_Almonds      R   C      150        4   3     95
## 46  Muesli_Raisins,_Peaches,_&_Pecans      R   C      150        4   3    150
## 40              Just_Right_Fruit_&_Nut      K   C      140        3   1    170
## 50      Nutri-Grain_Almond-Raisin      K   C      140        3   2    220
## 71      Total_Raisin_Bran      G   C      140        3   1    190
##      Fiber Carbo Sugars Potass Vitamins Shelf Weight Cups  Rating Cold Nabisco
## 47      3     17     13     160        25     3   1.50 0.67 30.31335     1      0
## 45      3     16     11     170        25     3   1.00 1.00 37.13686     1      0
## 46      3     16     11     170        25     3   1.00 1.00 34.13976     1      0
## 40      2     20      9      95       100     3   1.30 0.75 36.47151     1      0
## 50      3     21      7     130        25     3   1.33 0.67 40.69232     1      0
## 71      4     15     14     230       100     3   1.50 1.00 28.59278     1      0
##      Quaker Kelloggs GeneralMills Ralston AHFP
## 47      0          1              0      0      0
## 45      0          0              0      1      0
## 46      0          0              0      1      0
## 40      0          1              0      0      0
## 50      0          1              0      0      0
## 71      0          0              1      0      0
```

```
dfLOW<-tolower(df$Name.A1.S1)
```

```
head(dfLOW, 18)
```

## 3. Rename all the column names with lowercase letter

```
## [1] "100%_bran"          "100%_natural_bran"
## [3] "all-bran"           "all-bran_with_extra_fiber"
## [5] "almond_delight"     "apple_cinnamon_cheerios"
## [7] "apple_jacks"        "basic_4"
## [9] "bran_chex"          "bran_flakes"
## [11] "cap'n'crunch"       "cheerios"
## [13] "cinnamon_toast_crunch" "clusters"
## [15] "cocoa_puffs"        "corn_chex"
## [17] "corn_flakes"        "corn_pops"
```

```
dflof<-tolower(df[1:23])
```

```
head(dflof)
```

```
## [1] "c(\"100%_bran\", \"100%_natural_bran\", \"all-bran\", \"all-bran_with_extra_fiber\", \"almond_d
## [2] "c(\"n\", \"q\", \"k\", \"k\", \"r\", \"g\", \"k\", \"g\", \"r\", \"p\", \"q\", \"g\", \"g\", \"
## [3] "c(\"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"c\", \"
## [4] "c(70, 120, 70, 50, 110, 110, 110, 130, 90, 90, 120, 110, 120, 110, 110, 110, 100, 110, 110, 110
## [5] "c(4, 3, 4, 4, 2, 2, 2, 3, 2, 3, 1, 6, 1, 3, 1, 2, 2, 1, 1, 3, 3, 2, 2, 2, 2, 1, 3, 3, 3, 1, 2,
## [6] "c(1, 5, 1, 0, 2, 2, 0, 2, 1, 0, 2, 2, 3, 2, 1, 0, 0, 0, 1, 3, 0, 0, 1, 0, 1, 0, 0, 2, 0, 1, 0, 1,
```

```
names(df)[2] <- "Manufacturer"
```

```
head(df)
```

#### 4. Rename the column 'manuf' to 'Manufacturer'

```
##           Name.A1.S1 Manufacturer Type Calories Protein Fat Sodium Fiber
## 1           100%_Bran              N    C         70         4    1    130    10.0
## 2      100%_Natural_Bran            Q    C        120         3    5     15     2.0
## 3             All-Bran              K    C         70         4    1    260     9.0
## 4 All-Bran_with_Extra_Fiber          K    C         50         4    0    140    14.0
## 5           Almond_Delight          R    C        110         2    2    200     1.0
## 6  Apple_Cinnamon_Cheerios          G    C        110         2    2    180     1.5
##   Carbo Sugars Potass Vitamins Shelf Weight Cups   Rating Cold Nabisco Quaker
## 1   5.0      6     280       25    3      1 0.33 68.40297    1      1      0
## 2   8.0      8     135        0    3      1 1.00 33.98368    1      0      1
## 3   7.0      5     320       25    3      1 0.33 59.42551    1      0      0
## 4   8.0      0     330       25    3      1 0.50 93.70491    1      0      0
## 5  14.0      8      NA       25    3      1 0.75 34.38484    1      0      0
## 6  10.5     10       70       25    1      1 0.75 29.50954    1      0      0
##   Kelloggs GeneralMills Ralston AHFP
## 1         0             0        0    0
## 2         0             0        0    0
## 3         1             0        0    0
## 4         1             0        0    0
## 5         0             0        1    0
## 6         0             1        0    0
```

```
rating=mean(df$Rating)
rating
```

#### 5. List the names and details of the product having calories more than 100

```
## [1] 42.6657
```

```
df[df$calories<100,c(1:23),df$Rating<rating]
```

```
## Warning in if (drop) {: the condition has length > 1 and only the first element
## will be used
```

```
## Warning in if (!drop) {: the condition has length > 1 and only the first element
## will be used
```

```
## [1] Name.A1.S1 Manufacturer Type Calories Protein
## [6] Fat Sodium Fiber Carbo Sugars
## [11] Potass Vitamins Shelf Weight Cups
## [16] Rating Cold Nabisco Quaker Kelloggs
## [21] GeneralMills Ralston AHFP
## <0 rows> (or 0-length row.names)
```

6. Display the details of product made by manufacturer 'k'

C. Create a list with 4 vectors named fruits, Evennos,Mat,Amount.....

```
fruits<-c("Apple", "Orange", "Banana", "Mango")
Evennos<-c(20, 22, 24, 26, 28, 30, 32, 34, 36, 38)
Mat<-matrix(5, 4, 4)
Amount<-c(43.5, 67.8, 78.4, 99.9)
```

```
list1<-list(fruits, Evennos, Mat, Amount)
```

```
print(list1)
```

```
## [[1]]
## [1] "Apple" "Orange" "Banana" "Mango"
##
## [[2]]
## [1] 20 22 24 26 28 30 32 34 36 38
##
## [[3]]
## [,1] [,2] [,3] [,4]
## [1,] 5 5 5 5
## [2,] 5 5 5 5
## [3,] 5 5 5 5
## [4,] 5 5 5 5
##
## [[4]]
## [1] 43.5 67.8 78.4 99.9
```