Assignment 1

GROUP 19
John Felix 23IM3FP21
Bera Preeth 23IM3FP19
Madhav Agarwal 23IM3FP23
Tanush Agarwal 23IM3FP13

Question 1:
Import the dataset food_coded.csv. Identify the number of observations and variables. Check for missing values (in numerical columns), duplicate records, and incorrect data types. Report the types of the data columns and their measures of central tendency and dispersion.

Number of observations = 125
Number of variables = 61

```
print(f"number of observations:{df.shape[0]}")
print(f"number of variables:{df.shape[1]}")
```

```
number of observations:125
number of variables:61
```

Total number of missing values = 99

```
df.select_dtypes(include="number").isnull().sum().sum()
```

```
99
```

Number of missing values per column is given by:

```
df.select_dtypes(include="number").isnull().sum()
```

```
Gender                          0
breakfast                       0
calories_chicken                0
calories_day                   19
calories_scone                  1
coffee                          0
comfort_food_reasons_coded     19
cook                            3
comfort_food_reasons_coded.1    0
cuisine                        17
diet_current_coded              0
drink                           2
eating_changes_coded            0
eating_changes_coded1           0
eating_out                      0
employment                      9
ethnic_food                     0
exercise                       13
```

```
father_education                1
fav_cuisine_coded               0
fav_food                        2
fries                           0
fruit_day                       0
grade_level                     0
greek_food                      0
healthy_feeling                 0
ideal_diet_coded                0
income                          1
indian_food                     0
italian_food                    0
```

There are no duplicated rows

```
df.duplicated().sum()
```

```
0
```

There were a few columns that contained numerical data but were stored as objects (strings) and so they were converted back into the correct datatypes.
Converted incorrect datatypes to the correct ones as follows:

```python
for col in df.select_dtypes(include='object').columns:
    temp=pd.to_numeric(df[col],errors='coerce')
    conversion_rate=temp.notna().mean()
    if(conversion_rate>=0.9):
        df[col]=temp
        print(f"converted {col} to numeric with conversion ratio: {conversion_rate}")
    else:
        print(f"failed to convert {col} to numeric with converstion ratio: {conversion_rate}")
```

```
converted GPA to numeric with conversion ratio: 0.96
failed to convert comfort_food to numeric with converstion ratio: 0.0
failed to convert comfort_food_reasons to numeric with converstion ratio: 0.0
failed to convert diet_current to numeric with converstion ratio: 0.0
failed to convert eating_changes to numeric with converstion ratio: 0.0
failed to convert father_profession to numeric with converstion ratio: 0.0
failed to convert fav_cuisine to numeric with converstion ratio: 0.0
failed to convert food_childhood to numeric with converstion ratio: 0.0
failed to convert healthy_meal to numeric with converstion ratio: 0.0
failed to convert ideal_diet to numeric with converstion ratio: 0.0
failed to convert meals_dinner_friend to numeric with converstion ratio: 0.0
failed to convert mother_profession to numeric with converstion ratio: 0.0
failed to convert type_sports to numeric with converstion ratio: 0.0
converted weight to numeric with conversion ratio: 0.96
```

Printing the datatypes of each column:

```python
df.dtypes
```

✓ 0.0s

```
GPA                 float64
Gender                int64
breakfast             int64
calories_chicken      int64
calories_day        float64
                      ...
type_sports          object
veggies_day           int64
vitamins              int64
waffle_calories       int64
weight              float64
Length: 61, dtype: object
```

Finding the measures of central tendency and dispersion of various columns:

```
df.describe()
```

✓ 0.0s Python

	GPA	Gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food_reasons_coded	cook	comfort_food_reas
count	120.000000	125.000000	125.000000	125.000000	106.000000	124.000000	125.00000	106.000000	122.000000	
mean	3.415558	1.392000	1.112000	577.320000	3.028302	505.241935	1.75200	2.698113	2.786885	
std	0.390139	0.490161	0.316636	131.214156	0.639308	230.840506	0.43359	1.972042	1.038351	
min	2.200000	1.000000	1.000000	265.000000	2.000000	315.000000	1.00000	1.000000	1.000000	
25%	3.200000	1.000000	1.000000	430.000000	3.000000	420.000000	2.00000	2.000000	2.000000	
50%	3.500000	1.000000	1.000000	610.000000	3.000000	420.000000	2.00000	2.000000	3.000000	
75%	3.700000	2.000000	1.000000	720.000000	3.000000	420.000000	2.00000	3.000000	3.000000	
max	4.000000	2.000000	2.000000	720.000000	4.000000	980.000000	2.00000	9.000000	5.000000	

8 rows × 49 columns

```
#Fixing GPA and weight as their data is of incorrect type
food = read.csv("food_coded.csv")
str(food)
#describe(food)
food$GPA <- as.numeric(food$GPA)
food$weight = as.numeric(gsub("[^0-9.]", "", food$weight))
sum(food$weight[is.na(food$weight)])

str(food)
colSums(is.na(food))


#imputing Numeric and integer Columns
num_columns = sapply(food, is.numeric)

library(mice)
med = make.method(food)
med[num_columns] = "pmm"
med[!num_columns] = ""

impute_object = mice(food, method = med, m = 1)
food2 = complete(impute_object, action = "long")
colSums(is.na(food2))


#Check Duplicates
anyDuplicated(food2)
#food3 = food2[!duplicated(food), ]


#Return Data types and Central Tendency of Food Data
str(food2)
describe(food2)


#statistical Summary of Numerical Cols
summary(food2[,num_columns])
#dispersion of Numerical Columns
sds = sapply(food2[num_columns],sd)
sds^2
#Data type of All Columns
sapply(food, class)
```

```
> str(food)
'data.frame':   125 obs. of  61 variables:
 $ GPA                           : chr  "2.4" "3.654" "3.3" "3.2" ...
 $ Gender                        : int  2 1 1 1 1 1 2 1 1 1 ...
 $ breakfast                     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ calories_chicken              : int  430 610 720 430 720 610 610 720 430 430 ...
 $ calories_day                  : num  NaN 3 4 3 2 3 3 3 NaN 3 ...
 $ calories_scone                : num  315 420 420 420 420 420 980 420 420 420 315 ...
 $ coffee                        : int  1 2 2 2 2 2 2 1 1 2 ...
 $ comfort_food                  : chr  "none" "chocolate, chips, ice cream" "frozen yogurt, pizza, fast food" "Pizza, Mac and cheese, ice cream" ...
 $ comfort_food_reasons          : chr  "we dont have comfort " "stress, bored, anger" "stress, sadness" "Boredom" ...
 $ comfort_food_reasons_coded    : int  9 1 1 2 1 4 1 1 2 1 ...
 $ cook                          : num  2 3 1 2 1 3 2 3 3 3 ...
 $ comfort_food_reasons_coded.1: int  9 1 1 2 1 4 1 1 2 1 ...
 $ cuisine                       : num  NaN 1 3 2 2 NaN 1 1 1 1 ...
 $ diet_current                  : chr  "eat good and exercise" "I eat about three times a day with some snacks. I try to eat healthy but it doesn't always work out that- somet"| __truncated__ "t
oast and fruit for breakfast, salad for lunch, usually grilled chicken and veggies (or some variation) for dinner" "College diet, cheap and easy foods most nights. Weekends traditionally, cook
better homomade meals  " ...
 $ diet_current_coded            : int  1 2 3 2 2 2 3 1 1 1 ...
 $ drink                         : num  1 2 1 2 2 2 2 1 2 1 1 ...
 $ eating_changes                : chr  "eat faster " "I eat out more than usual. " "sometimes choosing to eat fast food instead of cooking simply for convenience" "Accepting cheap and premade/st
ore bought foods" ...
 $ eating_changes_coded          : int  1 1 1 1 3 1 2 2 2 1 ...
 $ eating_changes_coded1         : int  1 2 3 3 4 3 5 5 8 3 ...
 $ eating_out                    : int  3 2 2 2 2 1 2 2 5 3 ...
 $ employment                    : num  3 2 3 3 2 3 3 2 2 3 ...
 $ ethnic_food                   : int  1 4 5 5 4 4 5 2 5 5 ...
 $ exercise                      : num  1 1 2 3 1 2 1 2 NaN 1 ...
 $ father_education              : num  5 2 2 2 4 1 4 3 5 5 ...
 $ father_profession             : chr  "profesor " "Self employed " "owns business" "Mechanic " ...
 $ fav_cuisine                   : chr  "Arabic cuisine" "Italian" "italian" "Turkish " ...
 $ fav_cuisine_coded             : int  3 1 1 3 1 6 4 5 1 1 ...
 $ fav_food                      : num  1 1 3 1 3 3 1 1 3 1 ...
 $ food_childhood                : chr  "rice  and chicken " "chicken and biscuits, beef soup, baked beans" "mac and cheese, pizza, tacos" "Beef stroganoff, tacos, pizza" ...
 $ fries                         : int  2 1 1 2 1 1 1 1 1 1 ...
 $ fruit_day                     : int  5 4 5 4 4 2 4 5 4 5 ...
 $ grade_level                   : int  2 4 3 4 4 2 4 2 1 1 ...
 $ greek_food                    : int  5 4 5 5 4 2 5 3 5 5 ...
 $ healthy_feeling               : int  2 5 6 7 6 4 4 3 7 3 ...
 $ healthy_meal                  : chr  "looks not oily " "Grains, Veggies, (more of grains and veggies), small protein and fruit with dairy " "usually includes natural ingredients; nonprocessed
food" "Fresh fruits& vegetables, organic meats " ...
 $ ideal_diet                    : chr  "being healthy " "Try to eat 5-6 small meals a day. While trying to properly distribute carbs, protein, fruits, veggies, and dairy.  " "i would say my idea
l diet is my current diet" "Healthy, fresh veggies/fruits & organic foods " ...
 $ ideal_diet_coded              : int  8 3 6 2 2 2 2 2 6 2 ...
```


```
 $ healthy_meal                  : chr  "looks not oily " "Grains, Veggies, (more of grains and veggies), small protein and fruit with dairy " "usually includes natural ingredients; nonprocessed
food" "Fresh fruits& vegetables, organic meats " ...
 $ ideal_diet                    : chr  "being healthy " "Try to eat 5-6 small meals a day. While trying to properly distribute carbs, protein, fruits, veggies, and dairy.  " "i would say my idea
l diet is my current diet" "Healthy, fresh veggies/fruits & organic foods " ...
 $ ideal_diet_coded              : int  8 3 6 2 2 2 2 2 6 2 ...
```

```
 $ income                        : num  5 4 6 6 6 1 4 5 5 4 ...
 $ indian_food                   : int  5 4 5 5 2 5 5 1 5 4 ...
 $ italian_food                  : int  5 4 5 5 5 5 5 3 5 5 ...
 $ life_rewarding                : num  1 1 7 2 1 4 8 3 8 3 ...
 $ marital_status                : num  1 2 2 2 1 2 1 1 2 2 ...
 $ meals_dinner_friend           : chr  "rice, chicken,  soup" "Pasta, steak, chicken " "chicken and rice with veggies, pasta, some kind of healthy recipe" "Grilled chicken \nStuffed Shells\nHome
made Chili" ...
 $ mother_education              : num  1 4 2 4 5 1 4 2 5 5 ...
 $ mother_profession             : chr  "unemployed" "Nurse RN " "owns business" "Special Education Teacher" ...
 $ nutritional_check             : int  5 4 4 2 3 1 4 4 2 5 ...
 $ on_off_campus                 : num  1 1 2 1 1 1 2 1 1 1 ...
 $ parents_cook                  : int  1 1 1 1 1 2 2 1 2 3 ...
 $ pay_meal_out                  : int  2 4 3 2 4 5 2 5 3 3 ...
 $ persian_food                  : num  5 4 5 5 2 5 5 1 5 4 ...
 $ self_perception_weight        : num  3 3 6 5 4 5 4 3 4 3 ...
 $ soup                          : num  1 1 1 1 1 1 1 1 2 1 ...
 $ sports                        : num  1 1 2 2 1 2 1 2 2 1 ...
 $ thai_food                     : int  1 2 5 5 4 4 5 1 5 4 ...
 $ tortilla_calories             : num  1165 725 1165 725 940 ...
 $ turkey_calories               : int  345 690 500 690 500 345 690 500 345 345 ...
 $ type_sports                   : chr  "car racing" "Basketball " "none" "nan" ...
 $ veggies_day                   : int  5 4 5 3 4 1 4 4 3 5 ...
 $ vitamins                      : int  1 2 1 1 2 2 1 2 2 1 ...
 $ waffle_calories               : int  1315 900 900 1315 760 1315 1315 1315 760 900 ...
 $ weight                        : chr  "187" "155" "I'm not answering this. " "Not sure, 240" ...
> #describe(food)
> food$GPA <- as.numeric(food$GPA)
```

```
Warning message:
NAs introduced by coercion
```

```
> food$weight = as.numeric(gsub("[^0-9.]", "", food$weight))
```

```
Warning message:
NAs introduced by coercion
```

```
> sum(food$weight[is.na(food$weight)])
[1] NA
>
> str(food)
'data.frame':   125 obs. of  61 variables:
 $ GPA                           : num  2.4 3.65 3.3 3.2 3.5 ...
 $ Gender                        : int  2 1 1 1 1 1 2 1 1 1 ...
```

```
 $ ethnic_food                : int  1 4 5 5 4 4 5 2 5 5 ...
 $ exercise                   : num  1 1 2 3 1 2 1 2 NaN 1 ...
 $ father_education           : num  5 2 2 2 4 1 4 3 5 5 ...
 $ father_profession          : chr  "profesor " "Self employed " "owns business" "Mechanic " ...
 $ fav_cuisine                : chr  "Arabic cuisine" "Italian" "italian" "Turkish " ...
 $ fav_cuisine_coded          : int  3 1 1 3 1 6 4 5 1 1 ...
 $ fav_food                   : num  1 1 3 1 3 3 1 1 3 1 ...
 $ food_childhood             : chr  "rice  and chicken " "chicken and biscuits, beef soup, baked beans" "mac and cheese, pizza, tacos" "Beef stroganoff, tacos, pizza" ...
 $ fries                      : int  2 1 1 2 1 1 1 1 1 1 ...
 $ fruit_day                  : int  5 4 5 4 4 2 4 5 4 5 ...
 $ grade_level                : int  2 4 3 4 4 2 4 2 1 1 ...
 $ greek_food                 : int  5 4 5 5 4 2 5 3 5 5 ...
 $ healthy_feeling            : int  2 5 6 7 6 4 4 3 7 3 ...
 $ healthy_meal               : chr  "looks not oily " "Grains, Veggies, (more of grains and veggies), small protein and fruit with dairy " "usually includes natural ingredients; nonprocessed
food" "Fresh fruits& vegetables, organic meats " ...
 $ ideal_diet                 : chr  "being healthy " "Try to eat 5-6 small meals a day. While trying to properly distribute carbs, protein, fruits, veggies, and dairy.  " "i would say my idea
l diet is my current diet" "Healthy, fresh veggies/fruits & organic foods " ...
 $ ideal_diet_coded           : int  8 3 6 2 2 2 2 2 6 2 ...
 $ income                     : num  5 4 6 6 6 1 4 5 5 4 ...
 $ indian_food                : int  5 4 5 5 2 5 5 1 5 4 ...
 $ italian_food               : int  5 4 5 5 5 5 5 3 5 5 ...
 $ life_rewarding             : num  1 1 7 2 1 4 8 3 8 3 ...
 $ marital_status             : num  1 2 2 2 1 2 1 1 2 2 ...
 $ meals_dinner_friend        : chr  "rice, chicken,  soup" "Pasta, steak, chicken " "chicken and rice with veggies, pasta, some kind of healthy recipe" "Grilled chicken \nStuffed Shells\nHome
made Chili" ...
 $ mother_education           : num  1 4 2 4 5 1 4 2 5 5 ...
 $ mother_profession          : chr  "unemployed" "Nurse RN " "owns business" "Special Education Teacher" ...
 $ nutritional_check          : int  5 4 4 2 3 1 4 4 2 5 ...
 $ on_off_campus              : num  1 1 2 1 1 1 2 1 1 1 ...
 $ parents_cook               : int  1 1 1 1 1 2 2 1 2 3 ...
 $ pay_meal_out               : int  2 4 3 2 4 5 2 5 3 3 ...
 $ persian_food               : num  5 4 5 5 2 5 5 1 5 4 ...
 $ self_perception_weight     : num  3 3 6 5 4 5 4 3 4 3 ...
 $ soup                       : num  1 1 1 1 1 1 1 1 2 1 ...
 $ sports                     : num  1 1 2 2 1 2 1 2 2 1 ...
 $ thai_food                  : int  1 2 5 5 4 4 5 1 5 4 ...
 $ tortilla_calories          : num  1165 725 1165 725 940 ...
 $ turkey_calories            : int  345 690 500 690 500 345 690 500 345 345 ...
 $ type_sports                : chr  "car racing" "Basketball " "none" "nan" ...
 $ veggies_day                : int  5 4 5 3 4 1 4 4 3 5 ...
 $ vitamins                   : int  1 2 1 1 2 2 1 2 2 1 ...
 $ waffle_calories            : int  1315 900 900 1315 760 1315 1315 1315 760 900 ...
 $ weight                     : num  187 155 NA 240 190 190 180 137 180 125 ...
> colSums(is.na(food))
                       GPA                     Gender                  breakfast
                         5                          0                          0
           calories_chicken              calories_day             calories_scone
```

```
> colSums(is.na(food))
                       GPA                     Gender                  breakfast
                         5                          0                          0
           calories_chicken              calories_day             calories_scone
                         0                         19                          1
                    coffee             comfort_food        comfort_food_reasons
                         0                          0                          0
  comfort_food_reasons_coded                       cook comfort_food_reasons_coded.1
                        19                          3                          0
                   cuisine             diet_current        diet_current_coded
                        17                          0                          0
                     drink          eating_changes      eating_changes_coded
                         2                          0                          0
      eating_changes_coded1               eating_out               employment
                         0                          0                          9
              ethnic_food                  exercise          father_education
                         0                         13                          1
         father_profession               fav_cuisine       fav_cuisine_coded
                         0                          0                          0
                  fav_food             food_childhood                    fries
                         2                          0                          0
                 fruit_day               grade_level               greek_food
                         0                          0                          0
            healthy_feeling             healthy_meal               ideal_diet
                         0                          0                          0
          ideal_diet_coded                    income              indian_food
                         0                          1                          0
             italian_food             life_rewarding            marital_status
                         0                          1                          1
       meals_dinner_friend          mother_education         mother_profession
                         0                          3                          0
         nutritional_check            on_off_campus              parents_cook
                         0                          1                          0
              pay_meal_out            persian_food     self_perception_weight
                         0                          1                          1
                     soup                   sports               thai_food
                         1                          2                          0
         tortilla_calories          turkey_calories              type_sports
                         1                          0                          0
               veggies_day                 vitamins           waffle_calories
                         0                          0                          0
                    weight
                         3

>
>
> #imputing Numeric and integer Columns
> num_columns = sapply(food, is.numeric)
>
> library(mice)
```

```
                         1                          0                          0
               veggies_day                 vitamins           waffle_calories
                         0                          0                          0
                    weight
                         3

>
>
> #imputing Numeric and integer Columns
> num_columns = sapply(food, is.numeric)
>
> library(mice)
```

```
Attaching package: 'mice'

The following object is masked from 'package:stats':

    filter

The following objects are masked from 'package:base':

    cbind, rbind
```

```
> med = make.method(food)
> med[num_columns] = "pmm"
> med[!num_columns] = ""
>
> impute_object = mice(food, method = med, m = 1)

 iter imp variable
  1   1  GPA  calories_day  calories_scone  comfort_food_reasons_coded  cook  cuisine  drink  employment  exercise  father_education  fav_food  income  life_rewarding  marital_status  mother_ed
ucation  on_off_campus  persian_food  self_perception_weight  soup  sports  tortilla_calories  weight
  2   1  GPA  calories_day  calories_scone  comfort_food_reasons_coded  cook  cuisine  drink  employment  exercise  father_education  fav_food  income  life_rewarding  marital_status  mother_ed
ucation  on_off_campus  persian_food  self_perception_weight  soup  sports  tortilla_calories  weight
  3   1  GPA  calories_day  calories_scone  comfort_food_reasons_coded  cook  cuisine  drink  employment  exercise  father_education  fav_food  income  life_rewarding  marital_status  mother_ed
ucation  on_off_campus  persian_food  self_perception_weight  soup  sports  tortilla_calories  weight
  4   1  GPA  calories_day  calories_scone  comfort_food_reasons_coded  cook  cuisine  drink  employment  exercise  father_education  fav_food  income  life_rewarding  marital_status  mother_ed
ucation  on_off_campus  persian_food  self_perception_weight  soup  sports  tortilla_calories  weight
  5   1  GPA  calories_day  calories_scone  comfort_food_reasons_coded  cook  cuisine  drink  employment  exercise  father_education  fav_food  income  life_rewarding  marital_status  mother_ed
ucation  on_off_campus  persian_food  self_perception_weight  soup  sports  tortilla_calories  weight
```

```
Warning message:
Number of logged events: 12
```

```
> food2 = complete(impute_object, action = "long")
> colSums(is.na(food2))
                      GPA                    Gender                 breakfast
                        0                         0                         0
         calories_chicken             calories_day            calories_scone
                        0                         0                         0
```

```
> colSums(is.na(food2))
                      GPA                    Gender                 breakfast
                        0                         0                         0
         calories_chicken             calories_day            calories_scone
                        0                         0                         0
                   coffee             comfort_food       comfort_food_reasons
                        0                         0                         0
comfort_food_reasons_coded                      cook comfort_food_reasons_coded.1
                        0                         0                         0
                  cuisine             diet_current       diet_current_coded
                        0                         0                         0
                    drink          eating_changes     eating_changes_coded
                        0                         0                         0
     eating_changes_coded1               eating_out                 employment
                        0                         0                         0
              ethnic_food                 exercise          father_education
                        0                         0                         0
        father_profession              fav_cuisine          fav_cuisine_coded
                        0                         0                         0
                 fav_food           food_childhood                    fries
                        0                         0                         0
                fruit_day              grade_level                greek_food
                        0                         0                         0
           healthy_feeling             healthy_meal                ideal_diet
                        0                         0                         0
        ideal_diet_coded                   income                indian_food
                        0                         0                         0
            italian_food           life_rewarding             marital_status
                        0                         0                         0
      meals_dinner_friend          mother_education          mother_profession
                        0                         0                         0
       nutritional_check              on_off_campus              parents_cook
                        0                         0                         0
            pay_meal_out              persian_food      self_perception_weight
                        0                         0                         0
                    soup                    sports                 thai_food
                        0                         0                         0
        tortilla_calories           turkey_calories               type_sports
                        0                         0                         0
             veggies_day                  vitamins          waffle_calories
                        0                         0                         0
                  weight                      .imp                       .id
                        0                         0                         0
```

```
>
>
> #Check Duplicates
> anyDuplicated(food2)
[1] 0
```

[illegible]	38	125	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
indian_food	39	125	3.15	1.49	3.0	3.19	1.48	1.0	5	4.0	-0.10
italian_food	40	125	4.73	0.59	5.0	4.87	0.00	3.0	5	2.0	-2.00
life_rewarding	41	125	5.10	3.11	5.0	5.03	4.45	1.0	10	9.0	0.06
marital_status	42	125	1.50	0.55	1.0	1.49	0.00	1.0	4	3.0	0.71
meals_dinner_friend	43	125	61.12	35.06	60.0	61.03	44.48	1.0	122	121.0	0.03
mother_education	44	125	3.41	1.19	4.0	3.45	1.48	1.0	5	4.0	-0.31
mother_profession	45	125	58.00	33.07	58.0	58.32	41.51	1.0	113	112.0	-0.06
nutritional_check	46	125	3.15	1.21	3.0	3.17	1.48	1.0	5	4.0	-0.13
on_off_campus	47	125	1.32	0.68	1.0	1.15	0.00	1.0	4	3.0	2.12
parents_cook	48	125	1.53	0.75	1.0	1.40	0.00	1.0	5	4.0	1.46
pay_meal_out	49	125	3.41	1.04	3.0	3.31	0.00	2.0	6	4.0	0.97
persian_food	50	125	2.81	1.42	3.0	2.76	1.48	1.0	5	4.0	0.22
self_perception_weight	51	125	3.10	1.13	3.0	3.04	1.48	1.0	6	5.0	0.43
soup	52	125	1.22	0.41	1.0	1.15	0.00	1.0	2	1.0	1.36
sports	53	125	1.39	0.49	1.0	1.37	0.00	1.0	2	1.0	0.44
thai_food	54	125	3.34	1.44	3.0	3.42	1.48	1.0	5	4.0	-0.32
tortilla_calories	55	125	947.52	201.27	940.0	965.35	318.76	580.0	1165	585.0	-0.41
turkey_calories	56	125	555.04	152.37	500.0	548.12	229.80	345.0	850	505.0	0.23
type_sports	57	125	33.56	17.02	31.0	33.29	20.76	1.0	68	67.0	0.20
veggies_day	58	125	4.01	1.08	4.0	4.16	1.48	1.0	5	4.0	-0.89
vitamins	59	125	1.51	0.50	2.0	1.51	0.00	1.0	2	1.0	-0.05
waffle_calories	60	125	1073.40	248.67	900.0	1087.43	481.84	575.0	1315	740.0	-0.19
weight	61	125	159.75	32.34	155.0	157.65	29.65	100.0	265	165.0	0.78
.imp	62	125	1.00	0.00	1.0	1.00	0.00	1.0	1	0.0	NaN
.id	63	125	63.00	36.23	63.0	63.00	45.96	1.0	125	124.0	0.00

	kurtosis	se
GPA	0.15	0.03
Gender	-1.82	0.04
breakfast	3.94	0.03
calories_chicken	-0.01	11.74
calories_day	-0.80	0.06
calories_scone	0.45	20.62
coffee	-0.68	0.04
comfort_food	-1.23	3.24
comfort_food_reasons	-1.33	2.90
comfort_food_reasons_coded	2.76	0.18
cook	-0.33	0.09
comfort_food_reasons_coded.1	3.14	0.17
cuisine	9.09	0.09
diet_current	-1.23	3.24
diet_current_coded	0.86	0.07
drink	-1.96	0.04
eating_changes	-1.26	3.18
eating_changes_coded	1.11	0.07
eating_changes_coded1	1.71	0.23
eating_out	-0.23	0.10
employment	-1.31	0.05
[illegible]	[illegible]	[illegible]

```
eating_changes_coded          1.11  0.07
eating_changes_coded1         1.71  0.23
eating_out                   -0.23  0.10
employment                   -1.31  0.05
ethnic_food                  -0.70  0.11
exercise                     -0.86  0.06
father_education             -1.24  0.11
father_profession            -1.21  2.96
fav_cuisine                  -0.39  1.29
fav_cuisine_coded             0.18  0.17
fav_food                     -1.56  0.08
food_childhood               -1.17  2.95
fries                         6.31  0.03
fruit_day                     0.11  0.08
grade_level                  -1.39  0.10
greek_food                   -1.02  0.12
healthy_feeling              -1.16  0.23
healthy_meal                 -1.23  3.24
ideal_diet                   -1.23  3.24
ideal_diet_coded             -1.20  0.19
income                       -0.27  0.13
indian_food                  -1.37  0.13
italian_food                  2.74  0.05
life_rewarding               -1.48  0.28
marital_status                1.13  0.05
meals_dinner_friend          -1.21  3.14
mother_education             -1.08  0.11
mother_profession            -1.24  2.96
nutritional_check            -1.15  0.11
on_off_campus                 3.74  0.06
parents_cook                  2.50  0.07
pay_meal_out                  0.44  0.09
persian_food                 -1.23  0.13
self_perception_weight        0.13  0.10
soup                         -0.14  0.04
sports                       -1.82  0.04
thai_food                    -1.24  0.13
tortilla_calories            -1.08 18.00
turkey_calories              -0.93 13.63
type_sports                  -0.83  1.52
veggies_day                  -0.09  0.10
vitamins                     -2.01  0.04
waffle_calories              -1.65 22.24
weight                        0.90  2.89
.imp                           NaN  0.00
.id                          -1.23  3.24
>
>
```

```
> #statistical Summary of Numerical Cols
> summary(food2[,num_columns])
      GPA            Gender        breakfast     calories_chicken  calories_day   calories_scone
 Min.   :2.200   Min.   :1.000   Min.   :1.000   Min.   :265.0    Min.   :2.000   Min.   :315.0
 1st Qu.:3.200   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:430.0    1st Qu.:3.000   1st Qu.:420.0
 Median :3.500   Median :1.000   Median :1.000   Median :610.0    Median :3.000   Median :420.0
 Mean   :3.419   Mean   :1.392   Mean   :1.112   Mean   :577.3    Mean   :2.984   Mean   :503.7
 3rd Qu.:3.700   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:720.0    3rd Qu.:3.000   3rd Qu.:420.0
 Max.   :4.000   Max.   :2.000   Max.   :2.000   Max.   :720.0    Max.   :4.000   Max.   :980.0
     coffee      comfort_food_reasons_coded      cook       comfort_food_reasons_coded.1
 Min.   :1.000   Min.   :1.000               Min.   :1.000   Min.   :1.000
 1st Qu.:2.000   1st Qu.:2.000               1st Qu.:2.000   1st Qu.:2.000
 Median :2.000   Median :2.000               Median :3.000   Median :2.000
 Mean   :1.752   Mean   :2.736               Mean   :2.776   Mean   :2.688
 3rd Qu.:2.000   3rd Qu.:3.000               3rd Qu.:3.000   3rd Qu.:3.000
 Max.   :2.000   Max.   :9.000               Max.   :5.000   Max.   :9.000
    cuisine     diet_current_coded     drink      eating_changes_coded eating_changes_coded1
 Min.   :1.0   Min.   :1.00        Min.   :1.00   Min.   :1.000        Min.   : 1.000
 1st Qu.:1.0   1st Qu.:1.00        1st Qu.:1.00   1st Qu.:1.000        1st Qu.: 3.000
 Median :1.0   Median :2.00        Median :2.00   Median :1.000        Median : 4.000
 Mean   :1.4   Mean   :1.76        Mean   :1.56   Mean   :1.536        Mean   : 4.552
 3rd Qu.:1.0   3rd Qu.:2.00        3rd Qu.:2.00   3rd Qu.:2.000        3rd Qu.: 5.000
 Max.   :6.0   Max.   :4.00        Max.   :2.00   Max.   :4.000        Max.   :13.000
   eating_out      employment     ethnic_food      exercise     father_education fav_cuisine_coded
 Min.   :1.00   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000    Min.   :0.000
 1st Qu.:2.00   1st Qu.:2.00   1st Qu.:3.000   1st Qu.:1.000   1st Qu.:2.000    1st Qu.:1.000
 Median :2.00   Median :2.00   Median :4.000   Median :2.000   Median :4.000    Median :1.000
 Mean   :2.56   Mean   :2.44   Mean   :3.744   Mean   :1.656   Mean   :3.472    Mean   :2.424
 3rd Qu.:3.00   3rd Qu.:3.00   3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:4.000    3rd Qu.:4.000
 Max.   :5.00   Max.   :3.00   Max.   :5.000   Max.   :3.000   Max.   :5.000    Max.   :8.000
    fav_food        fries         fruit_day      grade_level      greek_food    healthy_feeling
 Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 1.000
 1st Qu.:1.00   1st Qu.:1.000   1st Qu.:4.000   1st Qu.:1.000   1st Qu.:3.000   1st Qu.: 3.000
 Median :1.00   Median :1.000   Median :5.000   Median :2.000   Median :4.000   Median : 5.000
 Mean   :1.72   Mean   :1.088   Mean   :4.224   Mean   :2.376   Mean   :3.488   Mean   : 5.456
 3rd Qu.:3.00   3rd Qu.:1.000   3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.: 8.000
 Max.   :3.00   Max.   :2.000   Max.   :5.000   Max.   :4.000   Max.   :5.000   Max.   :10.000
 ideal_diet_coded     income       indian_food     italian_food   life_rewarding   marital_status
 Min.   :1.000    Min.   :1.000   Min.   :1.000   Min.   :3.000   Min.   : 1.000   Min.   :1.000
 1st Qu.:2.000    1st Qu.:4.000   1st Qu.:2.000   1st Qu.:5.000   1st Qu.: 2.000   1st Qu.:1.000
 Median :3.000    Median :5.000   Median :3.000   Median :5.000   Median : 5.000   Median :1.000
 Mean   :3.704    Mean   :4.536   Mean   :3.152   Mean   :4.728   Mean   : 5.104   Mean   :1.504
 3rd Qu.:6.000    3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:2.000
 Max.   :8.000    Max.   :6.000   Max.   :5.000   Max.   :5.000   Max.   :10.000   Max.   :4.000
```

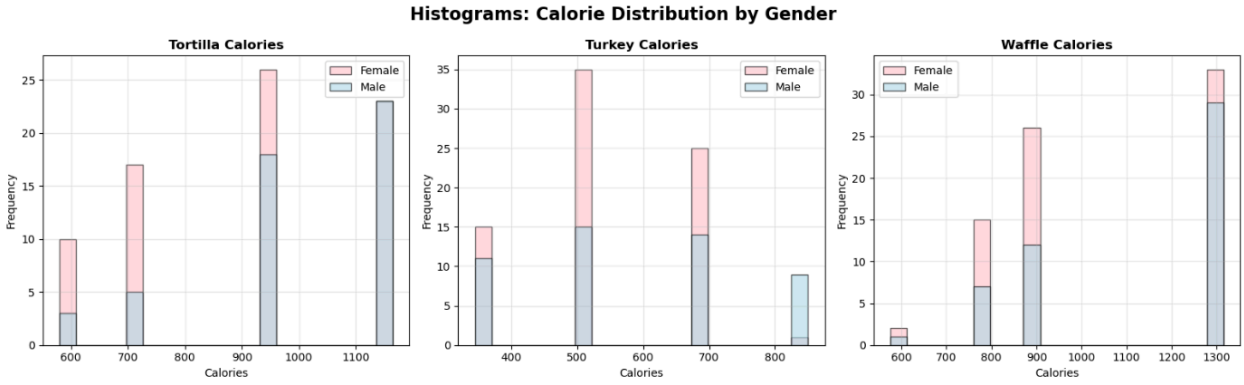
```
      Mean   :2.56   Mean   :2.44   Mean   :3.744   Mean   :1.656   Mean   :3.472   Mean   :2.424
      3rd Qu.:3.00   3rd Qu.:3.00   3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:4.000
      Max.   :5.00   Max.   :3.00   Max.   :5.000   Max.   :3.000   Max.   :5.000   Max.   :8.000
       fav_food         fries         fruit_day      grade_level     greek_food    healthy_feeling
      Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 1.000
      1st Qu.:1.00   1st Qu.:1.000   1st Qu.:4.000   1st Qu.:1.000   1st Qu.:3.000   1st Qu.: 3.000
      Median :1.00   Median :1.000   Median :5.000   Median :2.000   Median :4.000   Median : 5.000
      Mean   :1.72   Mean   :1.088   Mean   :4.224   Mean   :2.376   Mean   :3.488   Mean   : 5.456
      3rd Qu.:3.00   3rd Qu.:1.000   3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.: 8.000
      Max.   :3.00   Max.   :2.000   Max.   :5.000   Max.   :4.000   Max.   :5.000   Max.   :10.000
      ideal_diet_coded     income       indian_food     italian_food   life_rewarding   marital_status
      Min.   :1.000    Min.   :1.000   Min.   :1.000   Min.   :3.000   Min.   : 1.000   Min.   :1.000
      1st Qu.:2.000    1st Qu.:4.000   1st Qu.:2.000   1st Qu.:5.000   1st Qu.: 2.000   1st Qu.:1.000
      Median :3.000    Median :5.000   Median :3.000   Median :5.000   Median : 5.000   Median :1.000
      Mean   :3.704    Mean   :4.536   Mean   :3.152   Mean   :4.728   Mean   : 5.104   Mean   :1.504
      3rd Qu.:6.000    3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:2.000
      Max.   :8.000    Max.   :6.000   Max.   :5.000   Max.   :5.000   Max.   :10.000   Max.   :4.000
      mother_education nutritional_check on_off_campus   parents_cook    pay_meal_out    persian_food
      Min.   :1.000    Min.   :1.000     Min.   :1.00   Min.   :1.000   Min.   :2.000   Min.   :1.000
      1st Qu.:2.000    1st Qu.:2.000     1st Qu.:1.00   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:2.000
      Median :4.000    Median :3.000     Median :1.00   Median :1.000   Median :3.000   Median :3.000
      Mean   :3.408    Mean   :3.152     Mean   :1.32   Mean   :1.528   Mean   :3.408   Mean   :2.808
      3rd Qu.:4.000    3rd Qu.:4.000     3rd Qu.:1.00   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:4.000
      Max.   :5.000    Max.   :5.000     Max.   :4.00   Max.   :5.000   Max.   :6.000   Max.   :5.000
      self_perception_weight      soup           sports         thai_food     tortilla_calories
      Min.   :1.000          Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 580.0
      1st Qu.:2.000          1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.: 725.0
      Median :3.000          Median :1.000   Median :1.000   Median :3.000   Median : 940.0
      Mean   :3.104          Mean   :1.216   Mean   :1.392   Mean   :3.336   Mean   : 947.5
      3rd Qu.:4.000          3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:5.000   3rd Qu.:1165.0
      Max.   :6.000          Max.   :2.000   Max.   :2.000   Max.   :5.000   Max.   :1165.0
      turkey_calories  veggies_day      vitamins     waffle_calories     weight          .imp
      Min.   :345     Min.   :1.000   Min.   :1.000   Min.   : 575    Min.   :100.0   Min.   :1
      1st Qu.:500     1st Qu.:3.000   1st Qu.:1.000   1st Qu.: 900    1st Qu.:135.0   1st Qu.:1
      Median :500     Median :4.000   Median :2.000   Median : 900    Median :155.0   Median :1
      Mean   :555     Mean   :4.008   Mean   :1.512   Mean   :1073    Mean   :159.8   Mean   :1
      3rd Qu.:690     3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:1315    3rd Qu.:180.0   3rd Qu.:1
      Max.   :850     Max.   :5.000   Max.   :2.000   Max.   :1315    Max.   :265.0   Max.   :1
           .id
      Min.   :  1
      1st Qu.: 32
      Median : 63
      Mean   : 63
      3rd Qu.: 94
      Max.   :125
> #dispersion of Numerical Columns
```

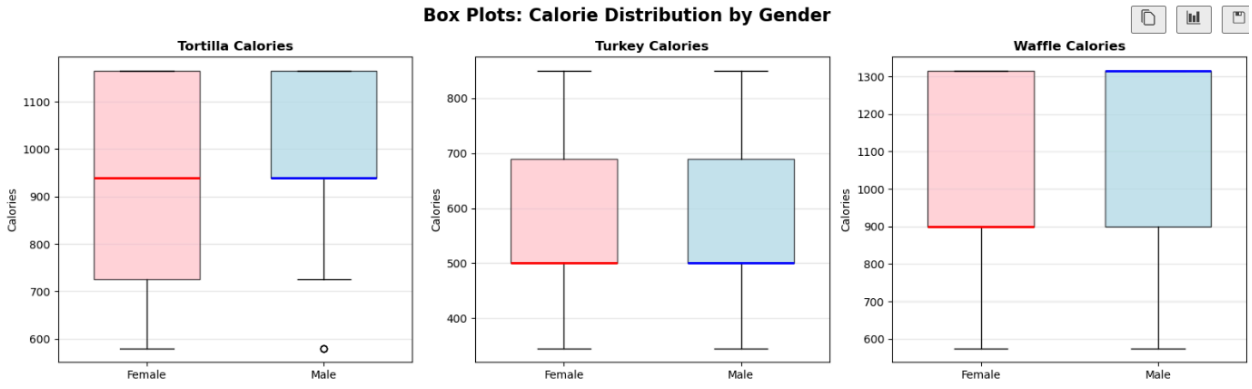
Question 2:

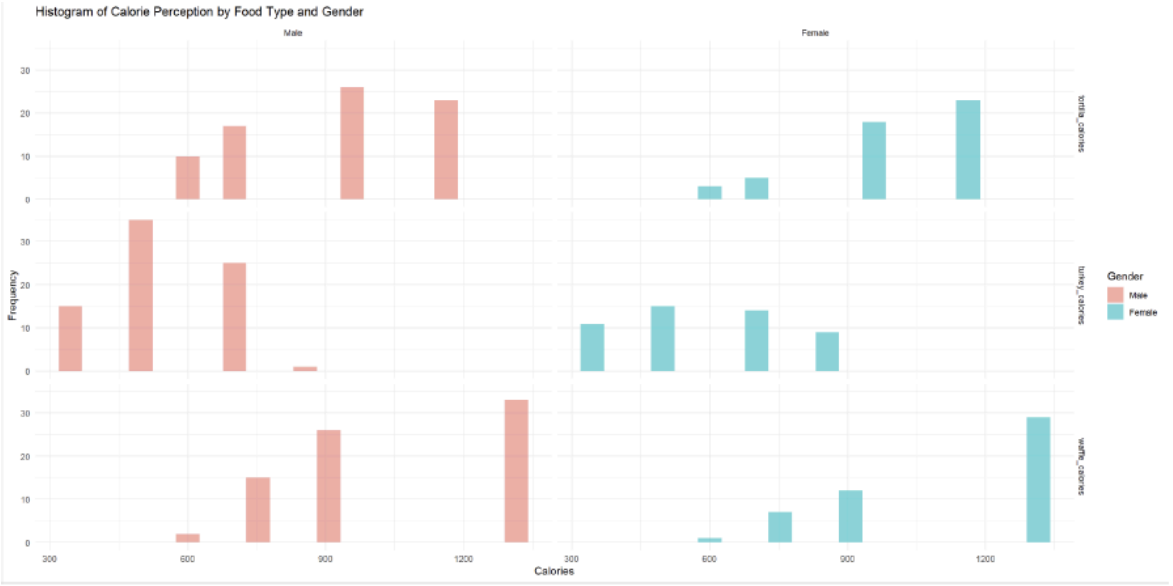
Use histograms and box plots to study calorie perception of 3 types of food based on gender.

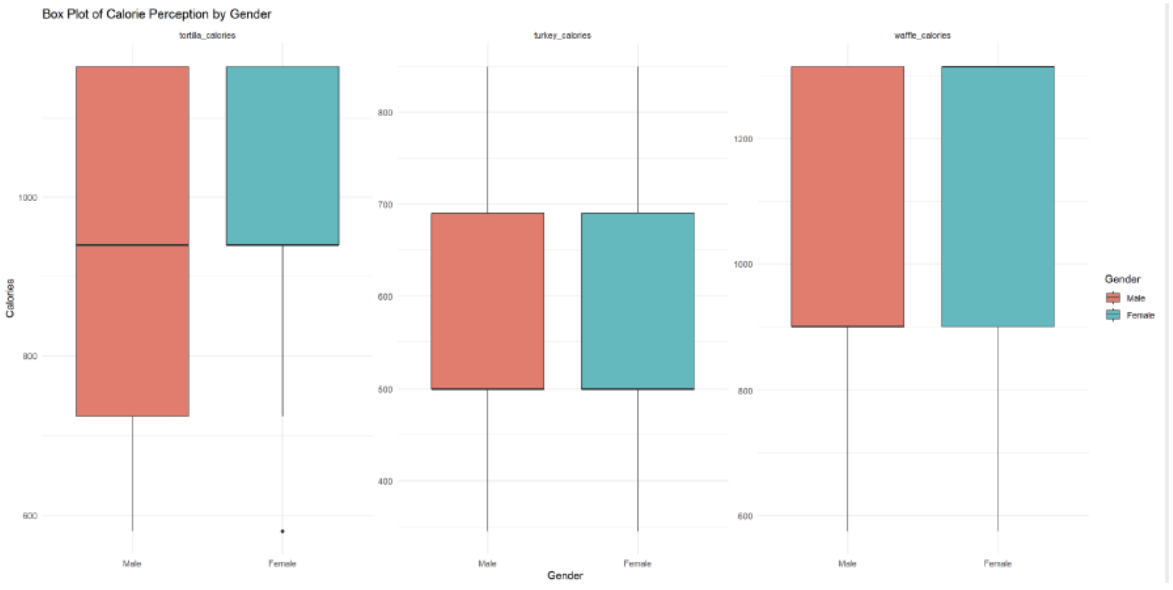
The median values of calorie perception of male and female is more or less the same for tortilla and turkey but significantly different for waffles. This can be seen in the plot given below.

```python
female_data = df[df['Gender'] == 1]
male_data = df[df['Gender'] == 2]
calorie_columns = ['tortilla_calories', 'turkey_calories', 'waffle_calories']
fig1, axes1 = plt.subplots(1, 3, figsize=(16, 5))
fig1.suptitle('Histograms: Calorie Distribution by Gender', fontsize=16, fontweight='bold')
for i, col in enumerate(calorie_columns):
    axes1[i].hist(female_data[col], bins=20, color='pink', alpha=0.6,
                  edgecolor='black', label='Female')
    axes1[i].hist(male_data[col], bins=20, color='lightblue', alpha=0.6,
                  edgecolor='black', label='Male')
    axes1[i].set_title(f'{col.replace("_", " ").title()}', fontsize=12, fontweight='bold')
    axes1[i].set_xlabel('Calories')
    axes1[i].set_ylabel('Frequency')
    axes1[i].legend()
    axes1[i].grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
fig2, axes2 = plt.subplots(1, 3, figsize=(16, 5))
fig2.suptitle('Box Plots: Calorie Distribution by Gender', fontsize=16, fontweight='bold')
for i, col in enumerate(calorie_columns):
    data_to_plot = [female_data[col].dropna(), male_data[col].dropna()]
    bp = axes2[i].boxplot(data_to_plot, labels=['Female', 'Male'],
                          patch_artist=True, widths=0.6)
    bp['boxes'][0].set_facecolor('pink')
    bp['boxes'][0].set_alpha(0.7)
    bp['boxes'][1].set_facecolor('lightblue')
    bp['boxes'][1].set_alpha(0.7)
    bp['medians'][0].set_color('red')
    bp['medians'][0].set_linewidth(2)
    bp['medians'][1].set_color('blue')
    bp['medians'][1].set_linewidth(2)
    axes2[i].set_title(f'{col.replace("_", " ").title()}', fontsize=12, fontweight='bold')
    axes2[i].set_ylabel('Calories')
    axes2[i].grid(True, alpha=0.3, axis='y')
```




Box Plots: Calorie Distribution by Gender
Tortilla Calories
Turkey Calories
Waffle Calories
Calories
Female
Male


Histogram of Calorie Perception by Food Type and Gender
Male
Female
Frequency
Calories
tortilla_calories
turkey_calories
waffle_calories
Gender
Male
Female

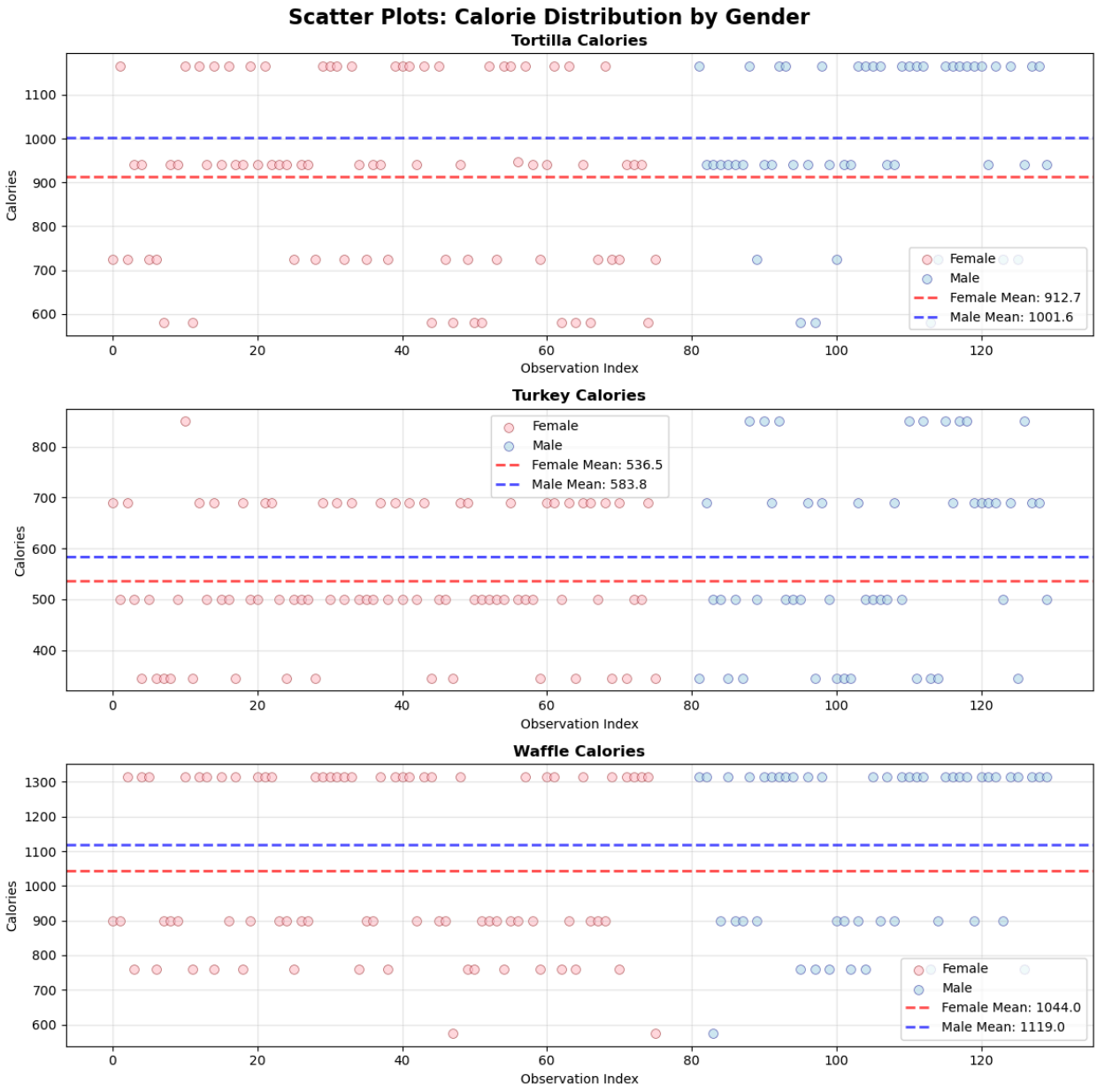

Box Plot of Calorie Perception by Gender
tortilla_calories
turkey_calories
waffle_calories
Calories
Gender
Male
Female

```r
# Question No 2
food = food2
colnames(food)

#plotting histogram
library(ggplot2)
library(tidyr)

food$Gender = factor(food$Gender,
                     levels = c(1, 2),
                     labels = c("Male", "Female"))

food_long = food %>%
  pivot_longer(
    cols = c(waffle_calories, tortilla_calories, turkey_calories),
    names_to = "Food_Type",
    values_to = "Calories"
  )

#Histogram
ggplot(food_long, aes(x = Calories, fill = Gender)) +
  geom_histogram(bins = 20, alpha = 0.6, position = "identity") +
  facet_grid(Food_Type ~ Gender) +
  labs(
    title = "Histogram of Calorie Perception by Food Type and Gender",
    x = "Calories",
    y = "Frequency"
  ) +
  theme_minimal()

#BoxPlot
ggplot(food_long, aes(x = Gender, y = Calories, fill = Gender)) +
  geom_boxplot() +
  facet_wrap(~ Food_Type, scales = "free_y") +
  labs(
    title = "Box Plot of Calorie Perception by Gender",
    x = "Gender",
    y = "Calories"
  ) +
  theme_minimal()
```

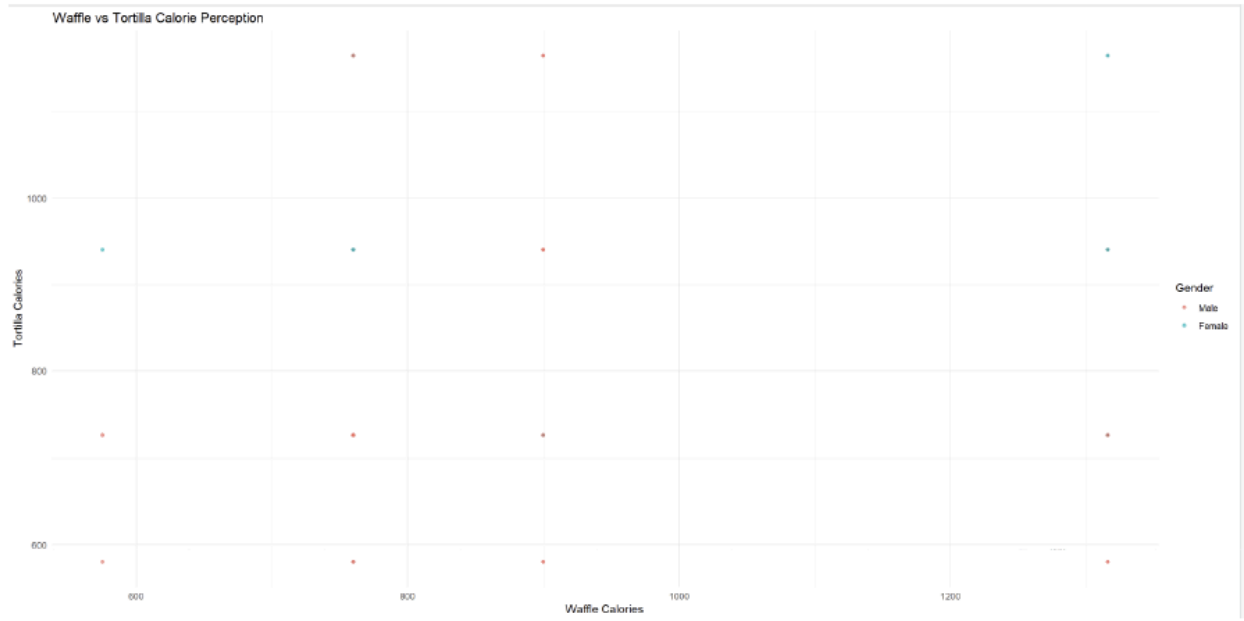
Question 3:
Create scatter plots for different calorie perceptions for food items chosen in Q2.

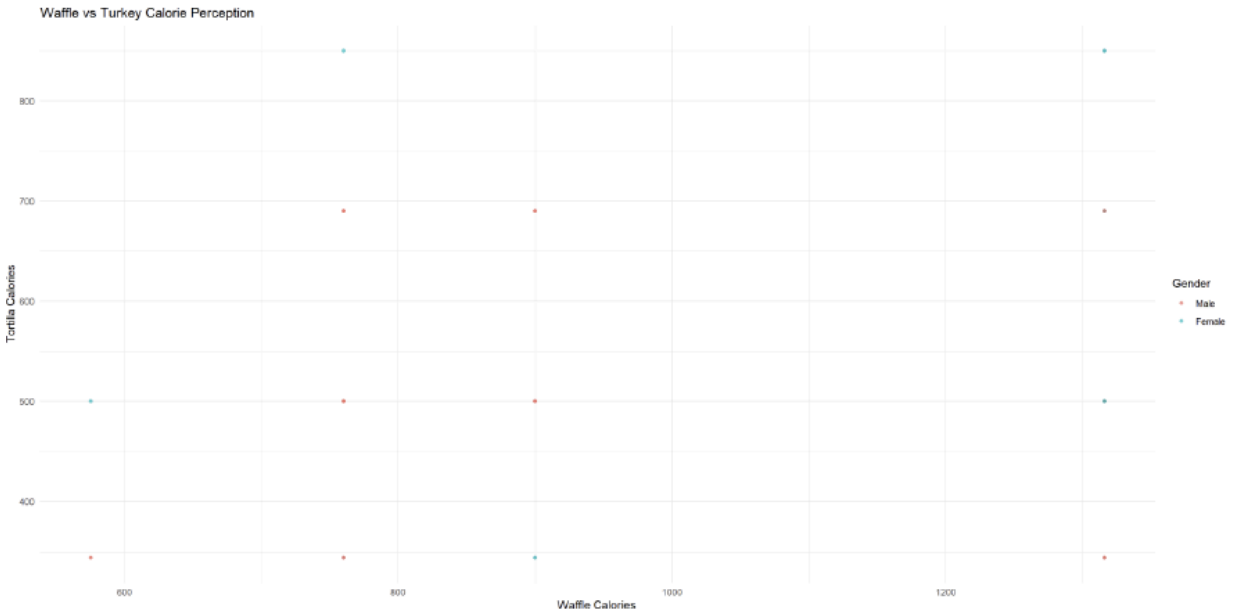
Even though the median values were more or less the same for both male and female for tortillas and turkey, the mean values are significantly different for both.

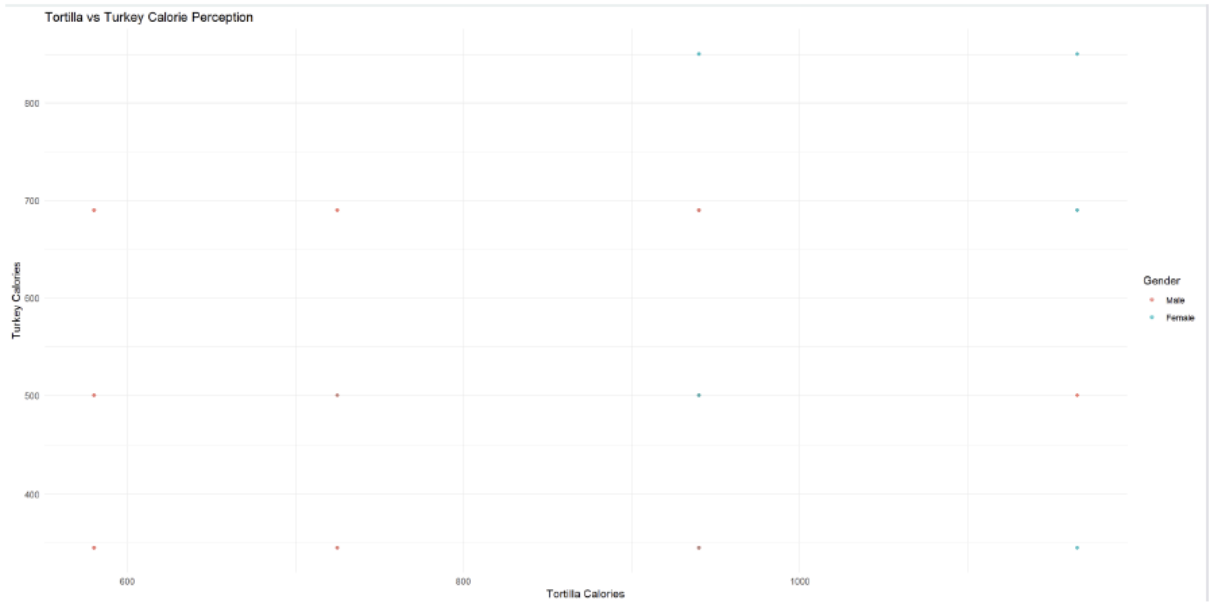
```python
fig3, axes3 = plt.subplots(3, 1, figsize=(12, 12))
fig3.suptitle('Scatter Plots: Calorie Distribution by Gender', fontsize=16, fontweight='bold')
for i, col in enumerate(calorie_columns):
    female_indices = np.arange(len(female_data))
    male_indices = np.arange(len(male_data))
    axes3[i].scatter(female_indices, female_data[col].values,
                     color='pink', alpha=0.6, s=50, label='Female', edgecolors='darkred', linewidth=0.5)
    axes3[i].scatter(male_indices + len(female_data) + 5, male_data[col].values,
                     color='lightblue', alpha=0.6, s=50, label='Male', edgecolors='darkblue', linewidth=0.5)
    axes3[i].axhline(y=female_data[col].mean(), color='red', linestyle='--',
                     linewidth=2, alpha=0.7, label=f'Female Mean: {female_data[col].mean():.1f}')
    axes3[i].axhline(y=male_data[col].mean(), color='blue', linestyle='--',
                     linewidth=2, alpha=0.7, label=f'Male Mean: {male_data[col].mean():.1f}')
    axes3[i].set_title(f'{col.replace("_", " ").title()}', fontsize=12, fontweight='bold')
    axes3[i].set_xlabel('Observation Index')
    axes3[i].set_ylabel('Calories')
    axes3[i].legend(loc='best')
    axes3[i].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

print("\n=== Summary Statistics ===\n")
for col in calorie_columns:
    print(f"\n{col.replace('_', ' ').title()}:")
    print(f"Female - Mean: {female_data[col].mean():.2f}, Median: {female_data[col].median():.2f}, Std: {female_data[col].std():.2f}")
    print(f"Male   - Mean: {male_data[col].mean():.2f}, Median: {male_data[col].median():.2f}, Std: {male_data[col].std():.2f}")
```


Scatter Plots: Calorie Distribution by Gender
Tortilla Calories
Calories
Observation Index
Female
Male
Female Mean: 912.7
Male Mean: 1001.6
Turkey Calories
Calories
Observation Index
Female
Male
Female Mean: 536.5
Male Mean: 583.8
Waffle Calories
Calories
Observation Index
Female
Male
Female Mean: 1044.0
Male Mean: 1119.0

```r
#############################################################################

# Question No 3
library(dplyr)

#Waffle vs Tortilla
ggplot(food, aes(x = waffle_calories, y = tortilla_calories, color = Gender)) +
  geom_point(alpha = 0.7) +
  labs(
    title = "Waffle vs Tortilla Calorie Perception",
    x = "Waffle Calories",
    y = "Tortilla Calories"
  ) +
  theme_minimal()


#Waffle vs Turkey
ggplot(food, aes(x = waffle_calories, y = turkey_calories, color = Gender)) + geom_point(alpha = 0.7) + labs(
  title = "Waffle vs Turkey Calorie Perception",
  x = "Waffle Calories",
  y = "Tortilla Calories"
) + theme_minimal()

#Tortilla vs Turkey
ggplot(food, aes(x = tortilla_calories, y = turkey_calories, color = Gender)) +
  geom_point(alpha = 0.7) +
  labs(
    title = "Tortilla vs Turkey Calorie Perception",
    x = "Tortilla Calories",
    y = "Turkey Calories"
  ) +
  theme_minimal()


#############################################################################################
```



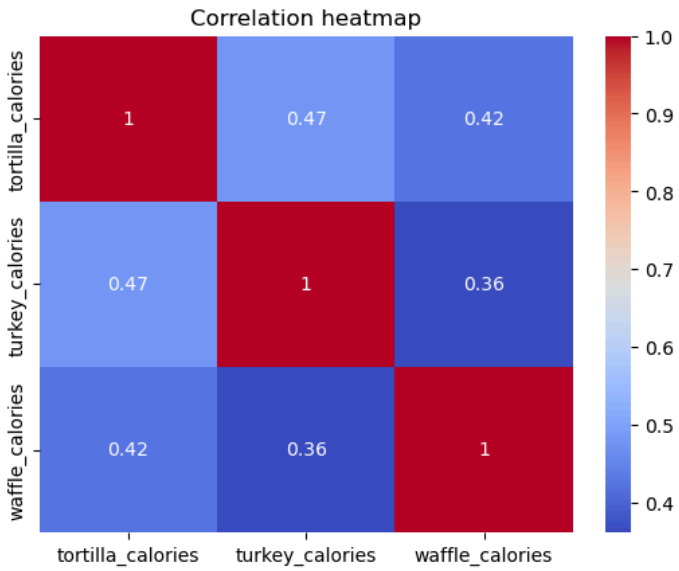




Question 4:
Create a heatmap for the variables used in Q2.

The calorie perception of those three foods have moderate positive correlation with each other as you can see in the heatmap.

```
import seaborn as sns
sns.heatmap(df[['tortilla_calories','turkey_calories','waffle_calories']].corr(),annot=True,cmap='coolwarm')
plt.title("Correlation heatmap")
plt.show()
```

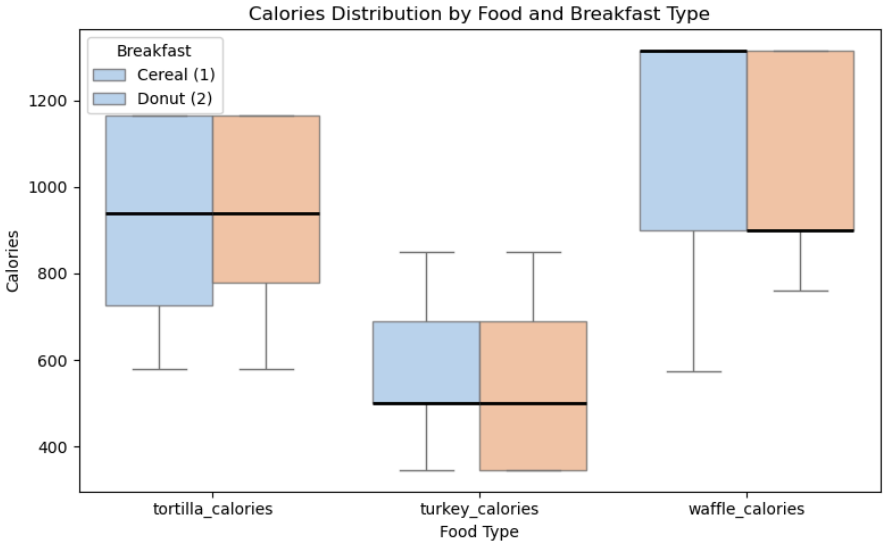



```r
#Question No 4
library(reshape2)

calorie_vars = food[, c("waffle_calories",
                        "tortilla_calories",
                        "turkey_calories")]

cor_matrix = cor(calorie_vars, use = "complete.obs")

cor_long = melt(cor_matrix)

#Create the heatmap using ggplot2
ggplot(cor_long, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(
    low = "blue",
    mid = "white",
    high = "red",
    midpoint = 0,
    limits = c(-1, 1),
    name = "Correlation"
  ) +
  labs(
    title = "Heatmap of Correlation Between Calorie Perception Variables",
    x = "",
    y = ""
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

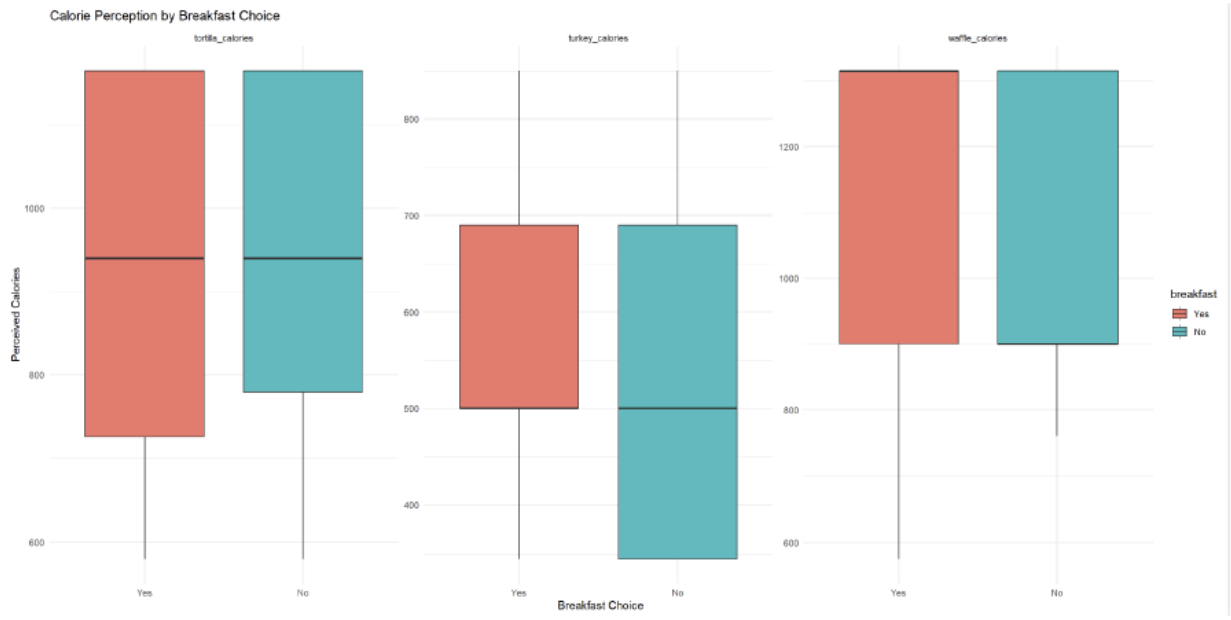
Question 5:
Create boxplot to assess if calorie perceptions for food items selected in Q2 depend on the breakfast choice.

We can see that the calorie perception by both groups are approximately same for tortilla and turkey but the cereal group has a much higher calorie perception of waffles than the donut group.

```python
import seaborn as sns
import matplotlib.pyplot as plt

cols = ['tortilla_calories', 'turkey_calories', 'waffle_calories']

long_df = df.melt(
    id_vars='breakfast',
    value_vars=cols,
    var_name='food',
    value_name='calories'
)

plt.figure(figsize=(8, 5))

sns.boxplot(
    data=long_df,
    x='food',
    y='calories',
    hue='breakfast',
    palette='pastel',
    medianprops={
        'color': 'black',
        'linewidth': 2
    },
    boxprops={'alpha': 0.8}
)

plt.title("Calories Distribution by Food and Breakfast Type")
plt.xlabel("Food Type")
plt.ylabel("Calories")
plt.legend(title="Breakfast", labels=["Cereal (1)", "Donut (2)"])
plt.tight_layout()
plt.show()
```



```r
# Question No 5

#Convert breakfast to a factor and reshape data to long format.
food = food2

food$breakfast = factor(food$breakfast,
                        levels = c(1, 2),
                        labels = c("Yes", "No"))

food_long <- food %>%
  pivot_longer(
    cols = c(waffle_calories, tortilla_calories, turkey_calories),
    names_to = "Food_Type",
    values_to = "Calories"
  )


#Create boxplots (Breakfast vs Calorie Perception)
ggplot(food_long, aes(x = breakfast, y = Calories, fill = breakfast)) +
  geom_boxplot() +
  facet_wrap(~ Food_Type, scales = "free_y") +
  labs(
    title = "Calorie Perception by Breakfast Choice",
    x = "Breakfast Choice",
    y = "Perceived Calories"
  ) +
  theme_minimal()

#Testing Dependence

#wilcox.test(waffle_calories ~ breakfast, data = food)
```



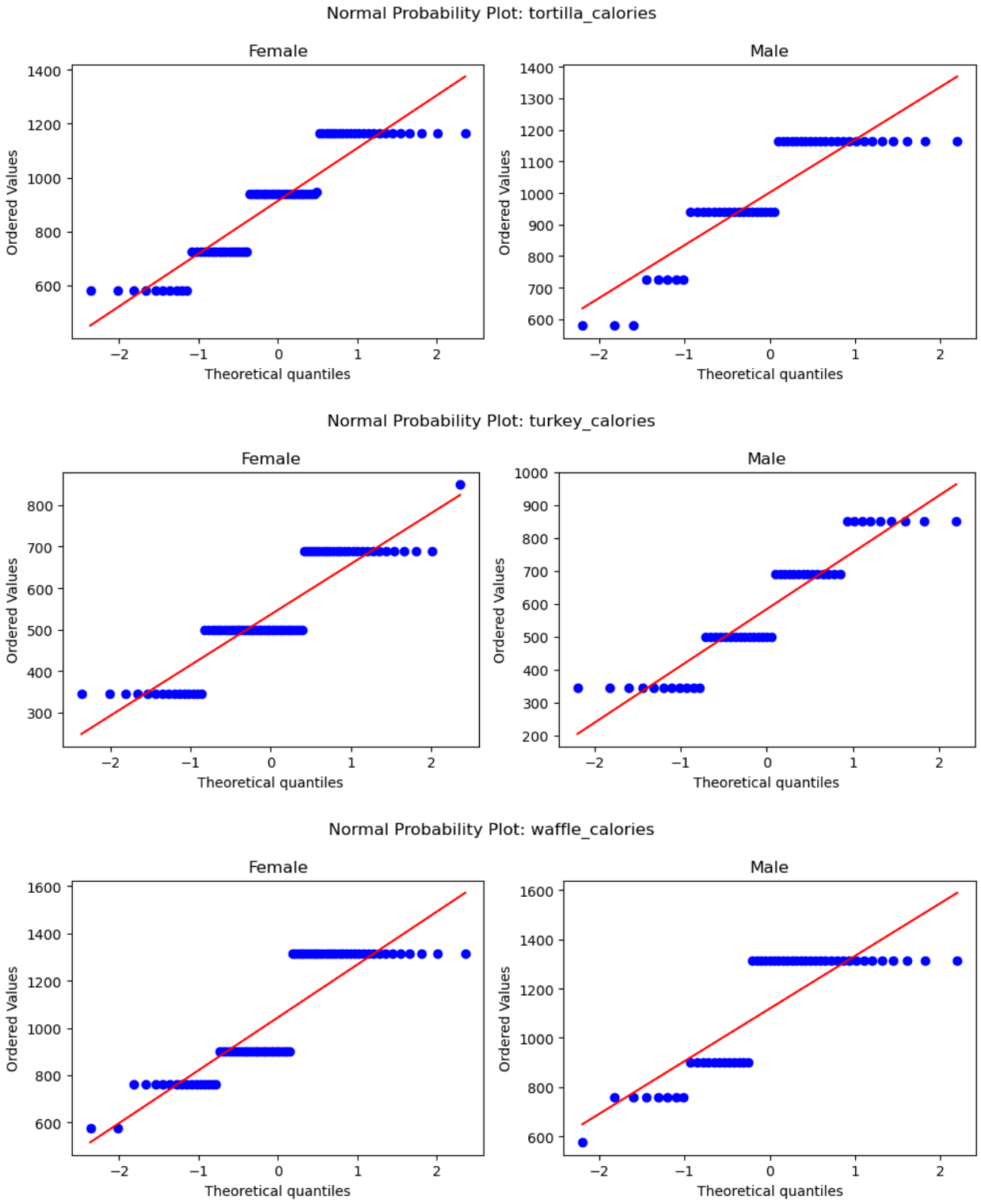
Question 6:
Calculate correlation coefficient for the pairs selected in Q2. Create Normal Probability Plot for the calorie perceptions of students of different genders (separately) for the food items selected in Q2.

We can see the exact values of correlation coefficients between the three food categories under study. From the NPP plots we can see that the data does not follow normal distribution. There are step structures for every food type. The points do not lie on the plot. The inferences that we can make are that the data is not approximately normal.

```
df[['tortilla_calories','turkey_calories','waffle_calories']].corr()
```
✓ 0.0s

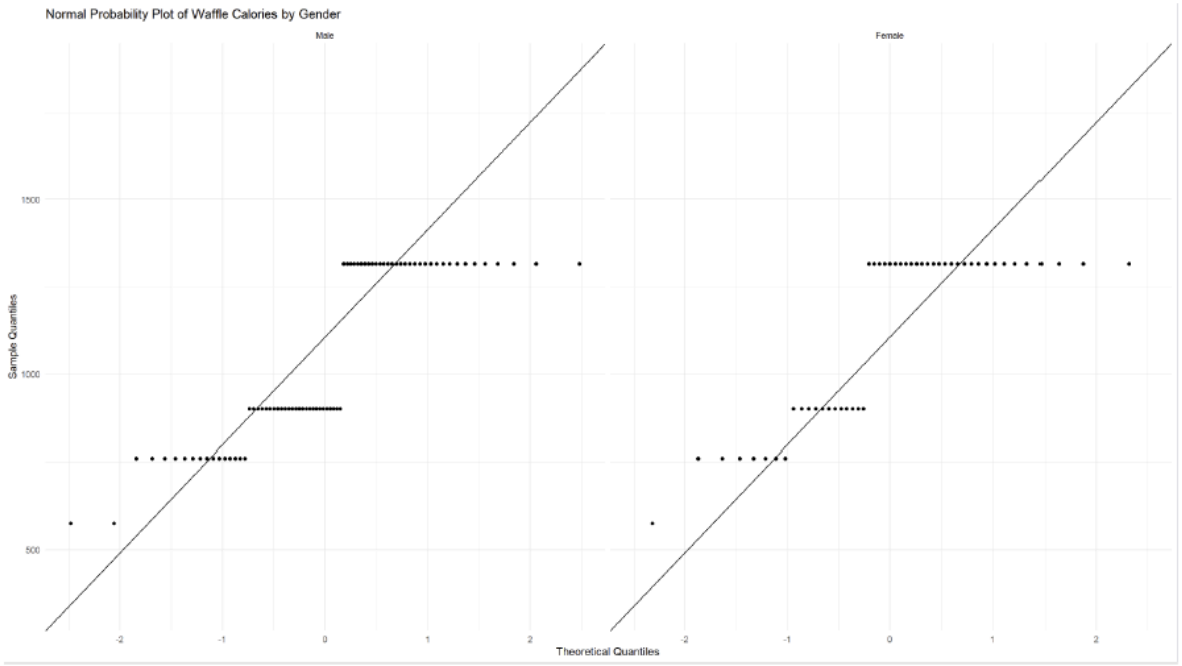
	tortilla_calories	turkey_calories	waffle_calories
tortilla_calories	1.000000	0.474794	0.421391
turkey_calories	0.474794	1.000000	0.361239
waffle_calories	0.421391	0.361239	1.000000

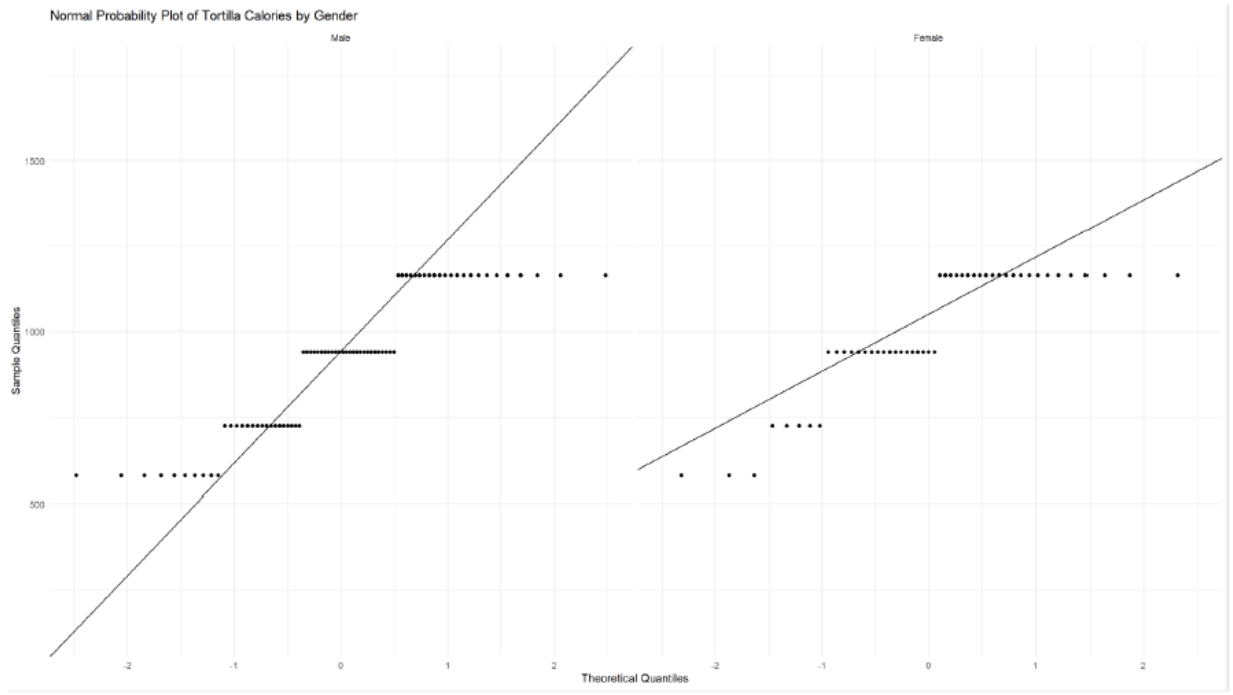
```
from scipy import stats
import matplotlib.pyplot as plt

cols = [
    'tortilla_calories',
    'turkey_calories',
    'waffle_calories'
]

gender_map = {1: 'Female', 2: 'Male'}

for col in cols:
    fig, axes = plt.subplots(1, 2, figsize=(10, 4))

    for ax, (g, label) in zip(axes, gender_map.items()):
        data = df.loc[df['Gender'] == g, col].dropna()
        stats.probplot(data, dist='norm', plot=ax)
        ax.set_title(label)

    fig.suptitle(f'Normal Probability Plot: {col}')
    plt.tight_layout()
    plt.show()
```

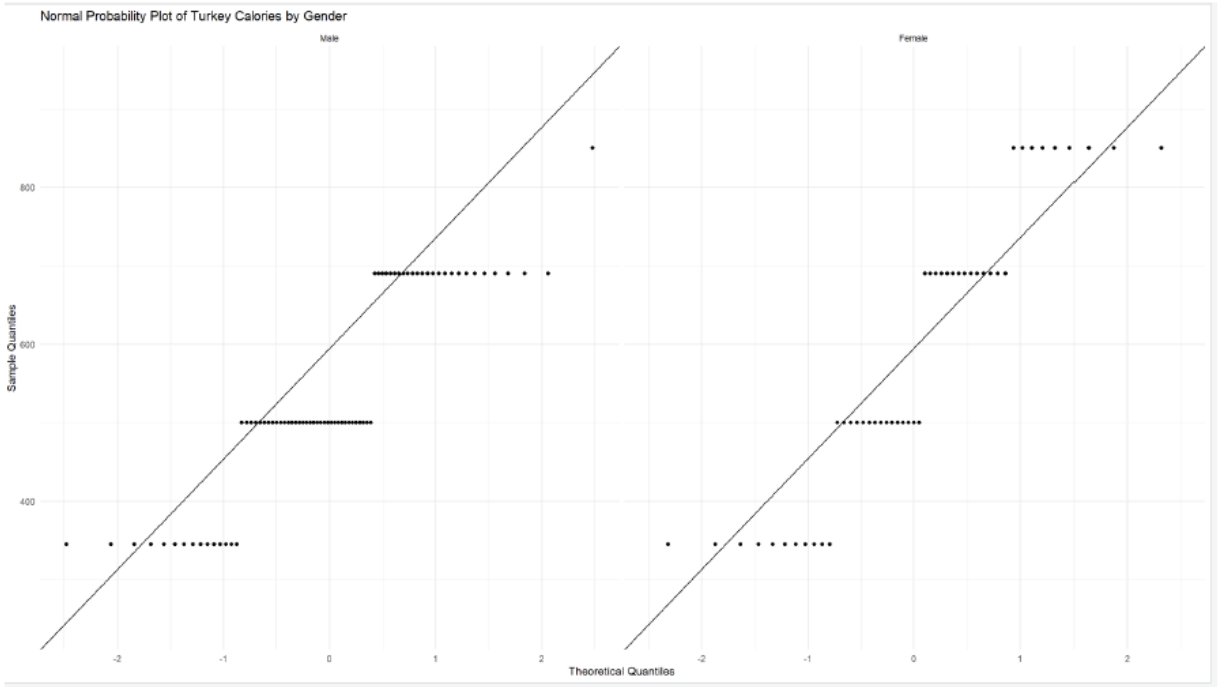

Normal Probability Plot: tortilla_calories
Female
Male
Ordered Values
Theoretical quantiles
Normal Probability Plot: turkey_calories
Female
Male
Ordered Values
Theoretical quantiles
Normal Probability Plot: waffle_calories
Female
Male
Ordered Values
Theoretical quantiles

```r
#Selecting Variable
calorie_data = food2[, c("waffle_calories","tortilla_calories", "turkey_calories")]

#Correlation Matrix
cor_matrix = cor(calorie_data, use = "complete.obs", method = "pearson")
cor_matrix

#Individual Correlation coeff
cor(food2$waffle_calories, food2$tortilla_calories, use = "complete.obs")
cor(food2$waffle_calories, food2$turkey_calories, use = "complete.obs")
cor(food2$tortilla_calories, food2$turkey_calories, use = "complete.obs")

#Preparing Gender Variable
food2$Gender = factor(food2$Gender,
                      levels = c(1, 2),
                      labels = c("Male", "Female"))

#Q-Q plots for each food item, separately by gender
#1. Waffle Calories
ggplot(food2, aes(sample = waffle_calories)) +
  stat_qq() +
  stat_qq_line() +
  facet_wrap(~ Gender) +
  labs(
    title = "Normal Probability Plot of Waffle Calories by Gender",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal()
#2. Tortilla Calories
ggplot(food2, aes(sample = tortilla_calories)) +
  stat_qq() +
  stat_qq_line() +
  facet_wrap(~ Gender) +
  labs(
    title = "Normal Probability Plot of Tortilla Calories by Gender",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal()
#3. Turkey Calories
ggplot(food2, aes(sample = turkey_calories)) +
  stat_qq() +
  stat_qq_line() +
  facet_wrap(~ Gender) +
  labs(
    title = "Normal Probability Plot of Turkey Calories by Gender",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal()
```

```
>
> ##############################################################################
>
> # Question No 6
>
> #Selecting Variable
> calorie_data = food2[, c("waffle_calories",
+                          "tortilla_calories",
+                          "turkey_calories")]
>
> #Correlation Matrix
> cor_matrix = cor(calorie_data, use = "complete.obs", method = "pearson")
> cor_matrix
                 waffle_calories tortilla_calories turkey_calories
waffle_calories        1.0000000         0.4216118       0.3612389
tortilla_calories      0.4216118         1.0000000       0.4749076
turkey_calories        0.3612389         0.4749076       1.0000000
>
> #Individual Correlation coeff
> cor(food2$waffle_calories, food2$tortilla_calories, use = "complete.obs")
[1] 0.4216118
> cor(food2$waffle_calories, food2$turkey_calories, use = "complete.obs")
[1] 0.3612389
> cor(food2$tortilla_calories, food2$turkey_calories, use = "complete.obs")
[1] 0.4749076
>
```




Normal Probability Plot of Tortilla Calories by Gender
Male
Female
Sample Quantiles
Theoretical Quantiles
500
1000
1500
-2
-1
0
1
2


Normal Probability Plot of Turkey Calories by Gender
Male
Female
Sample Quantiles
Theoretical Quantiles
400
600
800
-2
-1
0
1
2