TRƯỜNG ĐẠI HỌC FPT

Protein Prediction Based on Nutrition Attributes

ADY201m

Group 5:

SE203029 - Nguyễn Khánh Tường
SE201451- Hồ Nguyễn Trọng Nghĩa

Lecturer : Doan Nguyen Thanh Hoa

1. Understand the Problem

1.1Main Objective

- Explore and compare the nutritional composition of fast-food items across companies (e.g., McDonald's, KFC, Burger King, Wendy's).
- Answer practical questions such as:
  - **Compare the average overall energy (calorie) levels** across products from different fast-food companies.
  - **Examine the correlations** between key nutritional components (e.g., fat, protein, carbohydrates, sodium, etc.).

  - **Evaluate which brand may pose a higher cardiovascular risk**, based on nutrients related to heart health (e.g., saturated fat, trans fat, cholesterol, sodium).

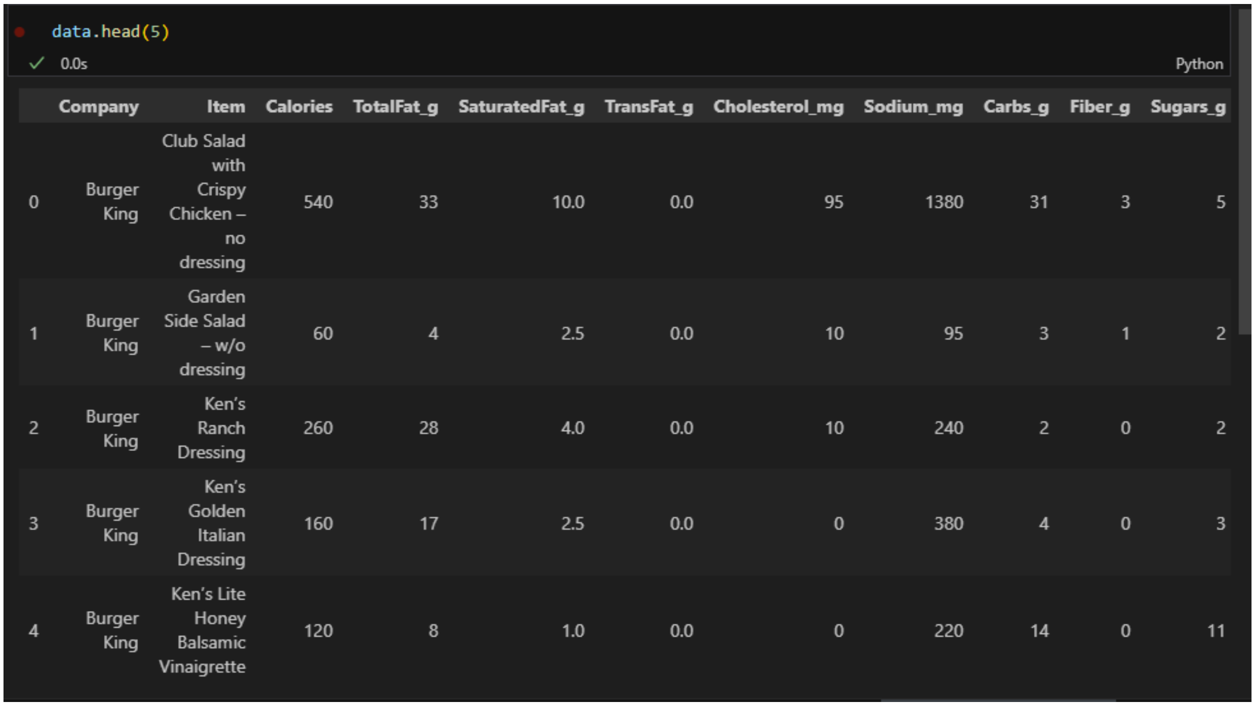
  - **Assess the "quality" of energy intake**, with the assumption that higher protein relative to total calories indicates better nutritional quality.

  - **Identify the "best" menu item** according to balanced and healthy nutritional criteria.
  -
- (Extended goal) Build a predictive model to estimate **Protein** from other nutritional attributes.

2. Data Understanding

2.1Dataset Overview

- **Size:** 859 rows × 13 columns.

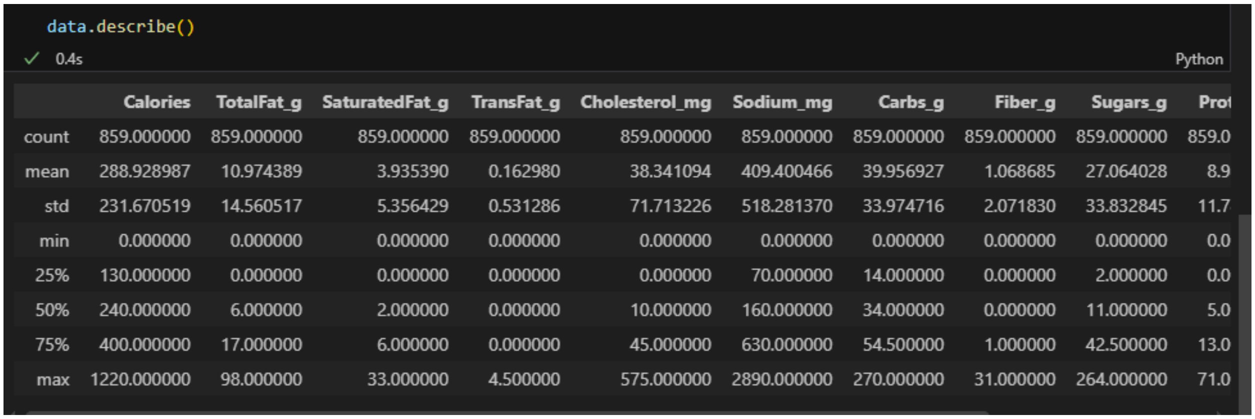


```
data.head(5)
```

✓ 0.0s Python

	Company	Item	Calories	TotalFat_g	SaturatedFat_g	TransFat_g	Cholesterol_mg	Sodium_mg	Carbs_g	Fiber_g	Sugars_g
0	Burger King	Club Salad with Crispy Chicken – no dressing	540	33	10.0	0.0	95	1380	31	3	5
1	Burger King	Garden Side Salad – w/o dressing	60	4	2.5	0.0	10	95	3	1	2
2	Burger King	Ken's Ranch Dressing	260	28	4.0	0.0	10	240	2	0	2
3	Burger King	Ken's Golden Italian Dressing	160	17	2.5	0.0	0	380	4	0	3
4	Burger King	Ken's Lite Honey Balsamic Vinaigrette	120	8	1.0	0.0	0	220	14	0	11

- **Data basic information**



```
data.describe()
```

✓ 0.4s Python

	Calories	TotalFat_g	SaturatedFat_g	TransFat_g	Cholesterol_mg	Sodium_mg	Carbs_g	Fiber_g	Sugars_g	Prot
count	859.000000	859.000000	859.000000	859.000000	859.000000	859.000000	859.000000	859.000000	859.000000	859.0
mean	288.928987	10.974389	3.935390	0.162980	38.341094	409.400466	39.956927	1.068685	27.064028	8.9
std	231.670519	14.560517	5.356429	0.531286	71.713226	518.281370	33.974716	2.071830	33.832845	11.7
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	130.000000	0.000000	0.000000	0.000000	0.000000	70.000000	14.000000	0.000000	2.000000	0.0
50%	240.000000	6.000000	2.000000	0.000000	10.000000	160.000000	34.000000	0.000000	11.000000	5.0
75%	400.000000	17.000000	6.000000	0.000000	45.000000	630.000000	54.500000	1.000000	42.500000	13.0
max	1220.000000	98.000000	33.000000	4.500000	575.000000	2890.000000	270.000000	31.000000	264.000000	71.0

  - **Columns:**

Company – The name of the fast food chain or restaurant that offers the menu item (e.g., McDonald's, Burger King, KFC). This attribute allows brand-level comparisons and grouping in analysis.

Item – The specific name or description of the menu item. This is a categorical identifier used to distinguish different food or beverage products.

Calories – The total energy provided by the food item, measured in kilocalories (kcal). This represents the amount of energy a consumer gains from eating the item.

TotalFat_g – The total fat content in grams. Fat contributes to energy intake and plays a significant role in determining the overall caloric value of the item.

SaturatedFat_g – The portion of fat that is saturated, measured in grams. High levels of saturated fat are associated with increased cholesterol and cardiovascular risk.

TransFat_g – The amount of trans fat in grams. Trans fats are artificially produced fats that negatively affect heart health and are often restricted in modern food standards.

Cholesterol_mg – The amount of cholesterol contained in the item, measured in milligrams. Cholesterol levels are important for assessing potential cardiovascular health impacts.

Sodium_mg – The sodium (salt) content in milligrams. Excessive sodium intake is linked to high blood pressure and heart disease, making this an essential nutritional indicator.

Carbs_g – The total carbohydrate content in grams, including starches and sugars. Carbohydrates are the main source of energy in most diets.

Fiber_g – The amount of dietary fiber in grams. Fiber aids digestion, promotes satiety, and helps regulate blood sugar and cholesterol levels.

Sugars_g – The total sugar content in grams, including both natural and added sugars. High sugar levels can contribute to obesity, diabetes, and other metabolic issues.

Protein_g – The protein content in grams. Protein is a critical nutrient for muscle repair, growth, and metabolic processes. This is the **target variable** for prediction in this project.

WeightWatchers_Points – A point-based nutritional scoring system used by the Weight Watchers program to assess the overall dietary impact of a food item. It summarizes calories, fat, and fiber into a single score to guide healthier eating decisions.

- **Missing data:** none
- **Top companies by item count:** McDonald's (~325), KFC (~200), Burger King (~180), Wendy's (~150).

2.2 Data Quality Observations

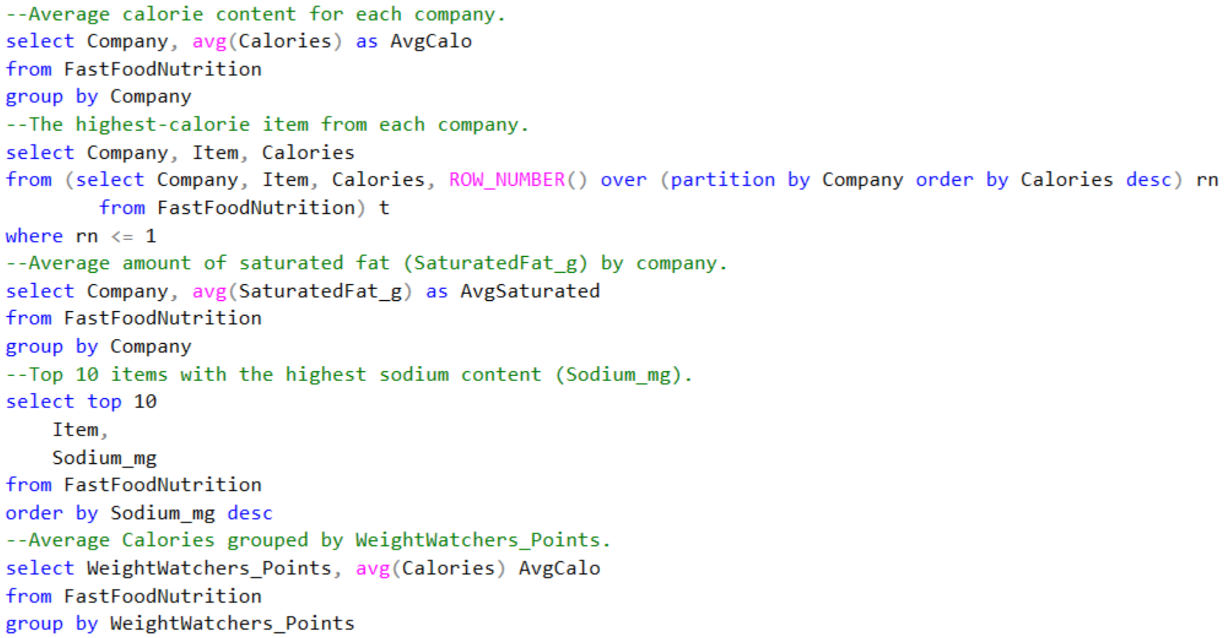
Some items have **Calories = 0**, usually drinks like water or diet soda.

2.3 Preprocessing (While importing code)

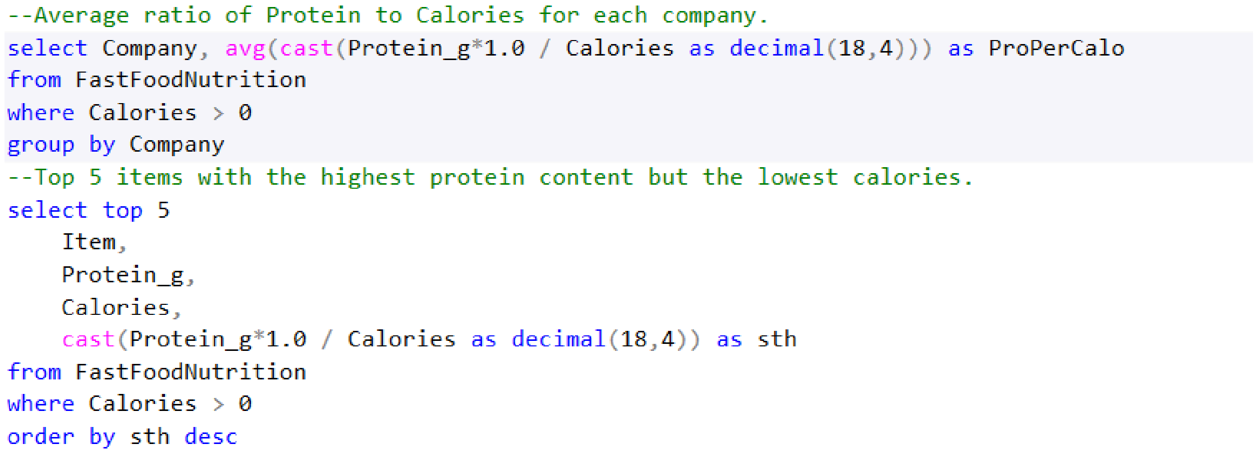
Remove every row containing null values.

Scaling (for regression later): Standardize numerical columns if using linear or ridge regression.

3. Analysis with SQL



```sql
--Average calorie content for each company.
select Company, avg(Calories) as AvgCalo
from FastFoodNutrition
group by Company
--The highest-calorie item from each company.
select Company, Item, Calories
from (select Company, Item, Calories, ROW_NUMBER() over (partition by Company order by Calories desc) rn
        from FastFoodNutrition) t
where rn <= 1
--Average amount of saturated fat (SaturatedFat_g) by company.
select Company, avg(SaturatedFat_g) as AvgSaturated
from FastFoodNutrition
group by Company
--Top 10 items with the highest sodium content (Sodium_mg).
select top 10
    Item,
    Sodium_mg
from FastFoodNutrition
order by Sodium_mg desc
--Average Calories grouped by WeightWatchers_Points.
select WeightWatchers_Points, avg(Calories) AvgCalo
from FastFoodNutrition
group by WeightWatchers_Points
```

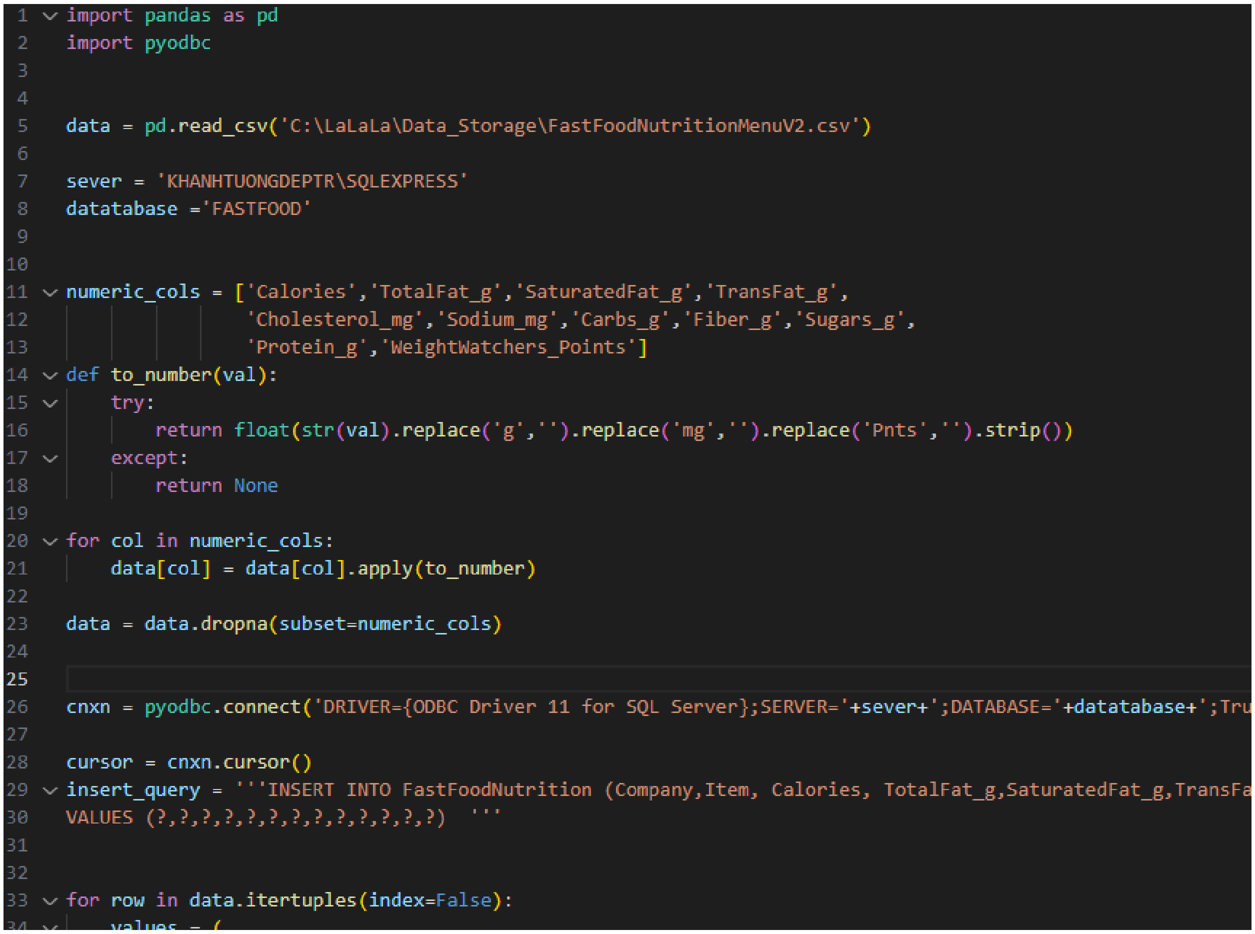


```sql
--Average ratio of Protein to Calories for each company.
select Company, avg(cast(Protein_g*1.0 / Calories as decimal(18,4))) as ProPerCalo
from FastFoodNutrition
where Calories > 0
group by Company
--Top 5 items with the highest protein content but the lowest calories.
select top 5
    Item,
    Protein_g,
    Calories,
    cast(Protein_g*1.0 / Calories as decimal(18,4)) as sth
from FastFoodNutrition
where Calories > 0
order by sth desc
```

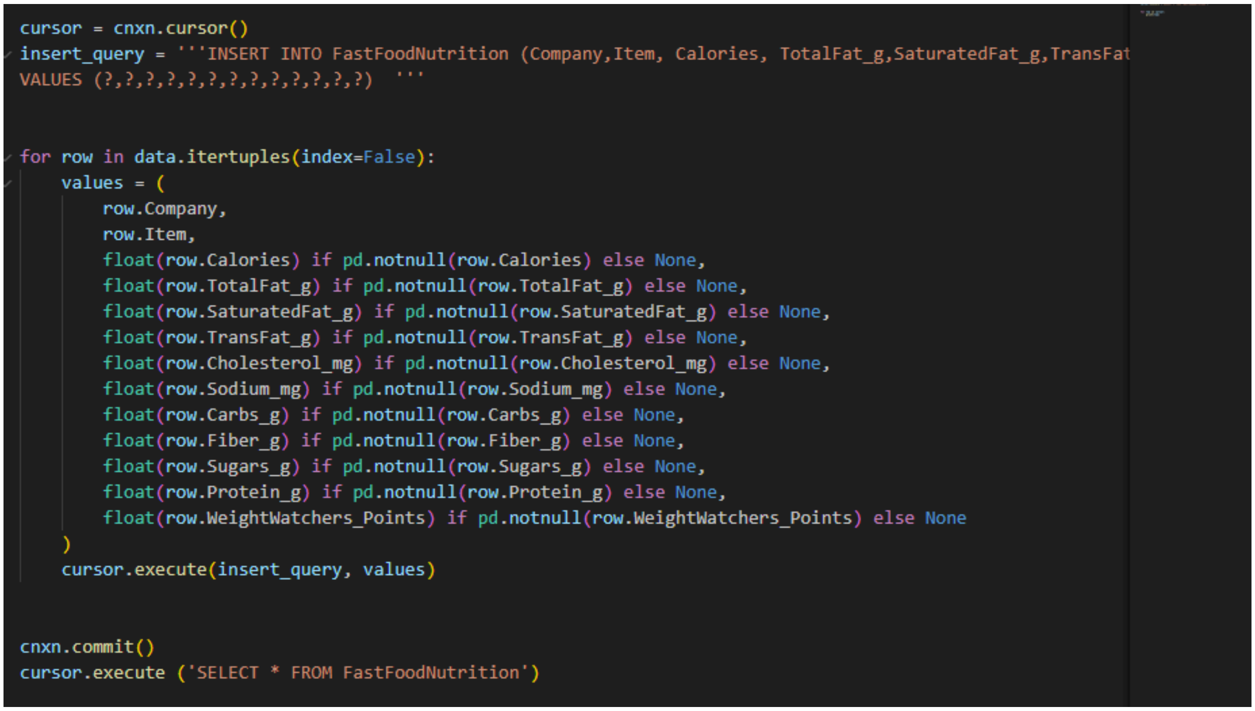
4. Analysis with Python

Expected analysis scope

- **Import dataset to SSMS via python code and clean data also.**



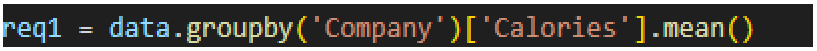
```python
import pandas as pd
import pyodbc


data = pd.read_csv('C:\LaLaLa\Data_Storage\FastFoodNutritionMenuV2.csv')

sever = 'KHANHTUONGDEPTR\SQLEXPRESS'
datatabase ='FASTFOOD'


numeric_cols = ['Calories','TotalFat_g','SaturatedFat_g','TransFat_g',
                'Cholesterol_mg','Sodium_mg','Carbs_g','Fiber_g','Sugars_g',
                'Protein_g','WeightWatchers_Points']
def to_number(val):
    try:
        return float(str(val).replace('g','').replace('mg','').replace('Pnts','').strip())
    except:
        return None

for col in numeric_cols:
    data[col] = data[col].apply(to_number)

data = data.dropna(subset=numeric_cols)


cnxn = pyodbc.connect('DRIVER={ODBC Driver 11 for SQL Server};SERVER='+sever+';DATABASE='+datatabase+';Tru

cursor = cnxn.cursor()
insert_query = '''INSERT INTO FastFoodNutrition (Company,Item, Calories, TotalFat_g,SaturatedFat_g,TransFa
VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?)  '''


for row in data.itertuples(index=False):
    values = (
```



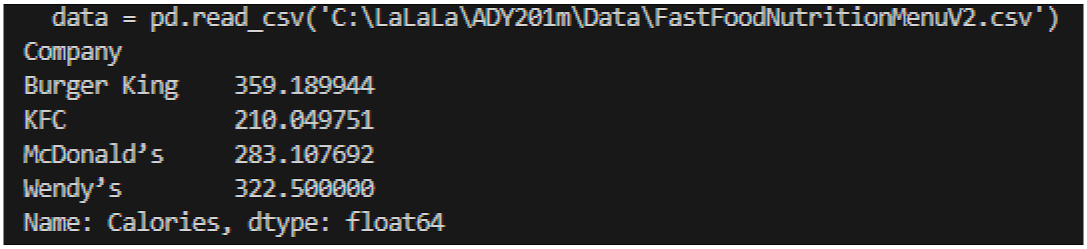
```python
cursor = cnxn.cursor()
insert_query = '''INSERT INTO FastFoodNutrition (Company,Item, Calories, TotalFat_g,SaturatedFat_g,TransFat
VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?)  '''


for row in data.itertuples(index=False):
    values = (
        row.Company,
        row.Item,
        float(row.Calories) if pd.notnull(row.Calories) else None,
        float(row.TotalFat_g) if pd.notnull(row.TotalFat_g) else None,
        float(row.SaturatedFat_g) if pd.notnull(row.SaturatedFat_g) else None,
        float(row.TransFat_g) if pd.notnull(row.TransFat_g) else None,
        float(row.Cholesterol_mg) if pd.notnull(row.Cholesterol_mg) else None,
        float(row.Sodium_mg) if pd.notnull(row.Sodium_mg) else None,
        float(row.Carbs_g) if pd.notnull(row.Carbs_g) else None,
        float(row.Fiber_g) if pd.notnull(row.Fiber_g) else None,
        float(row.Sugars_g) if pd.notnull(row.Sugars_g) else None,
        float(row.Protein_g) if pd.notnull(row.Protein_g) else None,
        float(row.WeightWatchers_Points) if pd.notnull(row.WeightWatchers_Points) else None
    )
    cursor.execute(insert_query, values)


cnxn.commit()
cursor.execute ('SELECT * FROM FastFoodNutrition')
```

- **Compute mean and correlation between nutrients.**

  -Average calorie content for each company

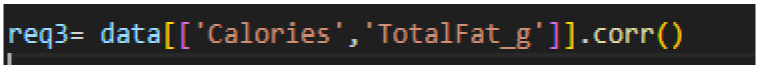


```
req1 = data.groupby('Company')['Calories'].mean()
```

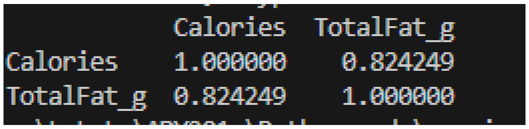


```
  data = pd.read_csv('C:\LaLaLa\ADY201m\Data\FastFoodNutritionMenuV2.csv')
Company
Burger King    359.189944
KFC            210.049751
McDonald’s     283.107692
Wendy’s        322.500000
Name: Calories, dtype: float64
```

  -Relationship between Calories and TotalFat_g



```
req3= data[['Calories','TotalFat_g']].corr()
```

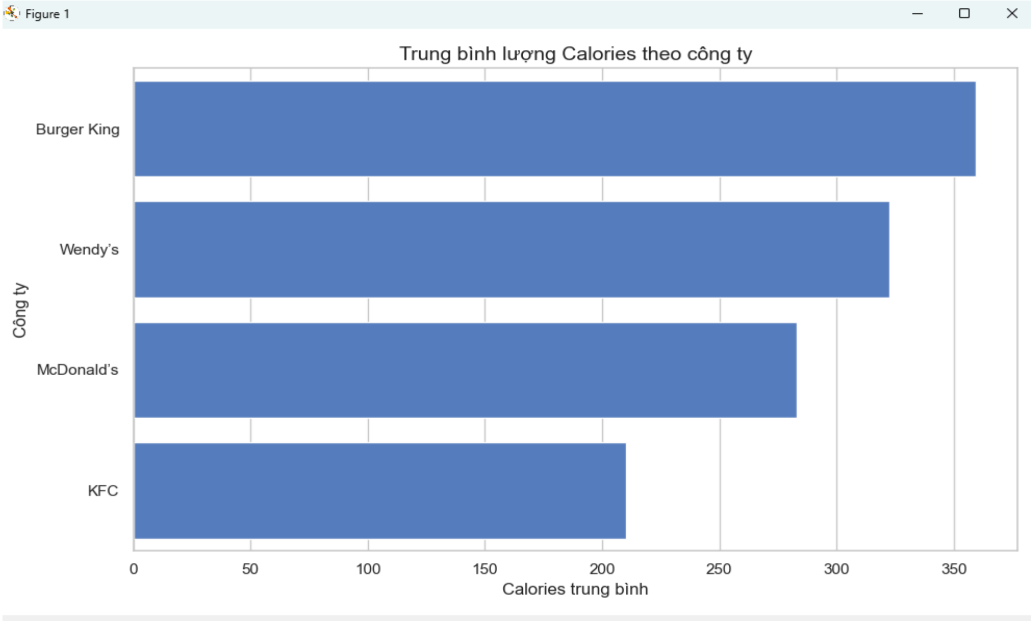


```
            Calories  TotalFat_g
Calories    1.000000    0.824249
TotalFat_g  0.824249    1.000000
```

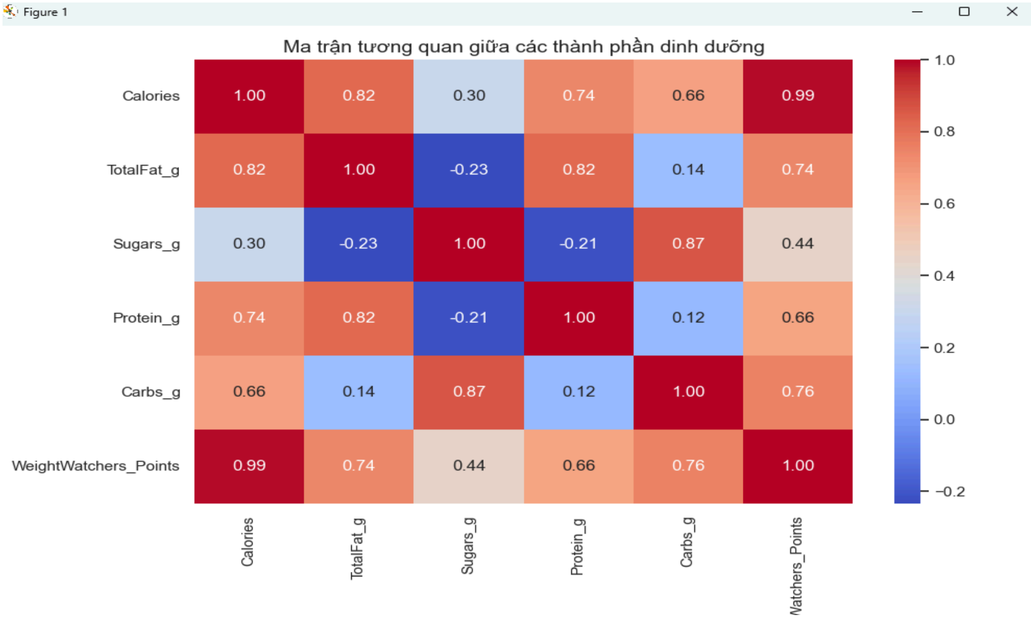
5. Visualization

- **Use seaborn and matplotlib.pyplot to display queries:**

Compare the average overall energy (calorie) levels across products from different fast-food companies.

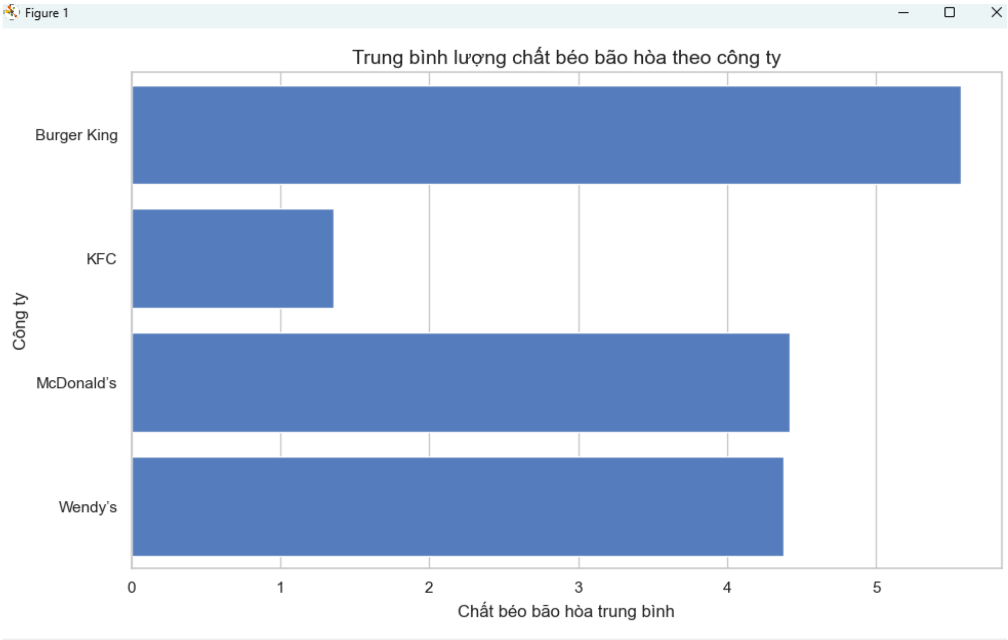


Examine the correlations between key nutritional components (e.g., fat, protein, carbohydrates, sodium, etc.).

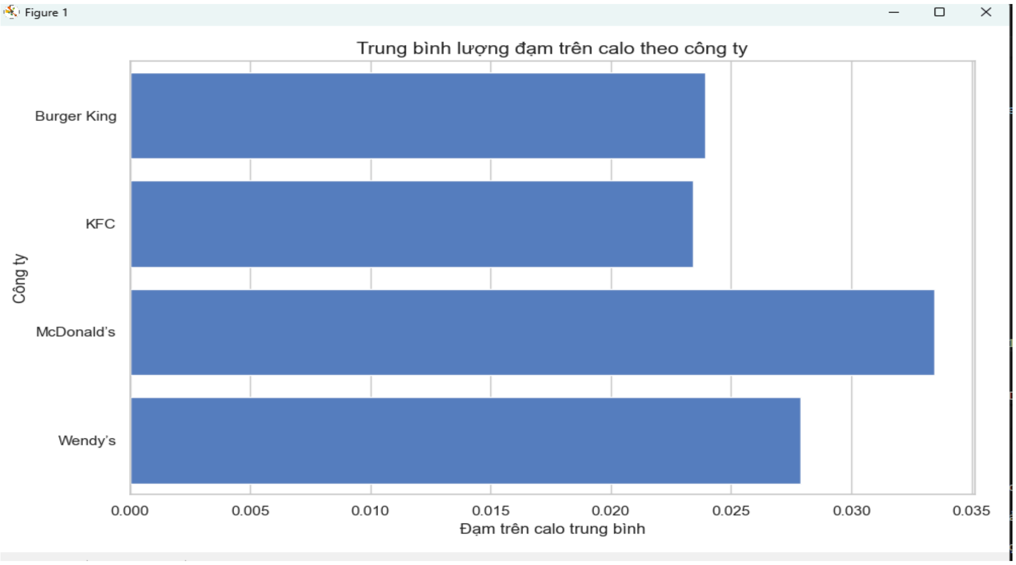


	Calories	TotalFat_g	Sugars_g	Protein_g	Carbs_g	Natchers_Points
Calories	1.00	0.82	0.30	0.74	0.66	0.99
TotalFat_g	0.82	1.00	-0.23	0.82	0.14	0.74
Sugars_g	0.30	-0.23	1.00	-0.21	0.87	0.44
Protein_g	0.74	0.82	-0.21	1.00	0.12	0.66
Carbs_g	0.66	0.14	0.87	0.12	1.00	0.76
WeightWatchers_Points	0.99	0.74	0.44	0.66	0.76	1.00

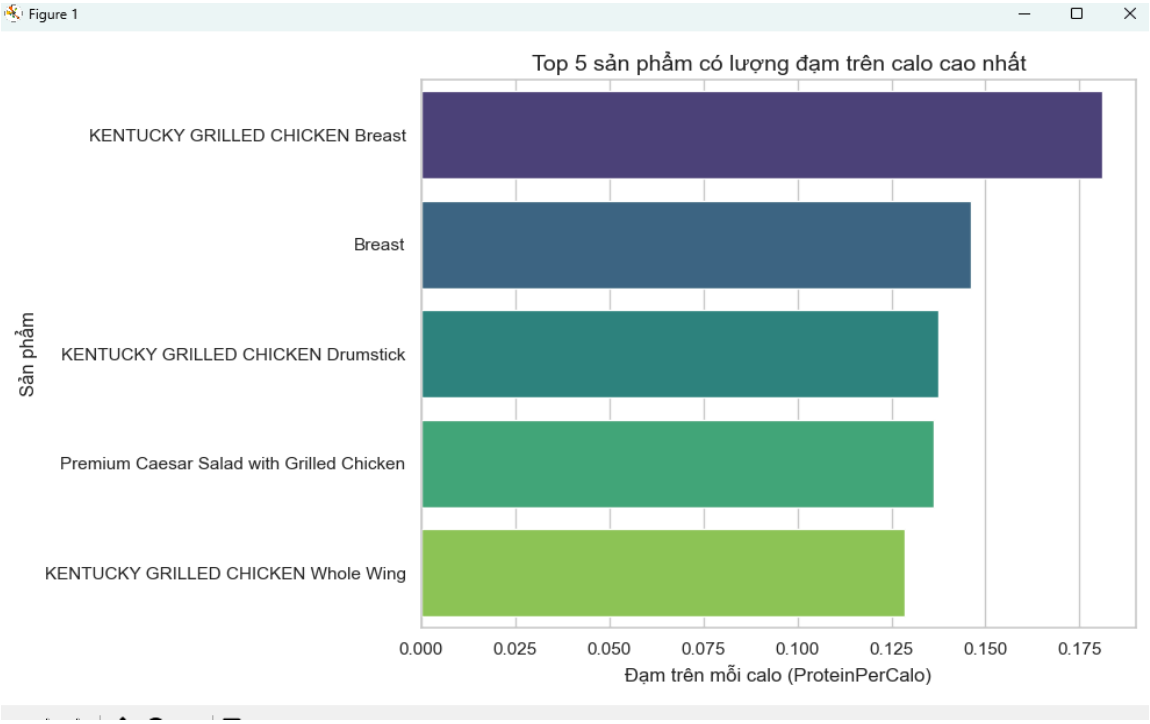
Evaluate which brand may pose a higher cardiovascular risk, based on nutrients related to heart health (e.g., saturated fat, trans fat, cholesterol, sodium).



Assess the "quality" of energy intake, with the assumption that higher protein relative to total calories indicates better nutritional quality.

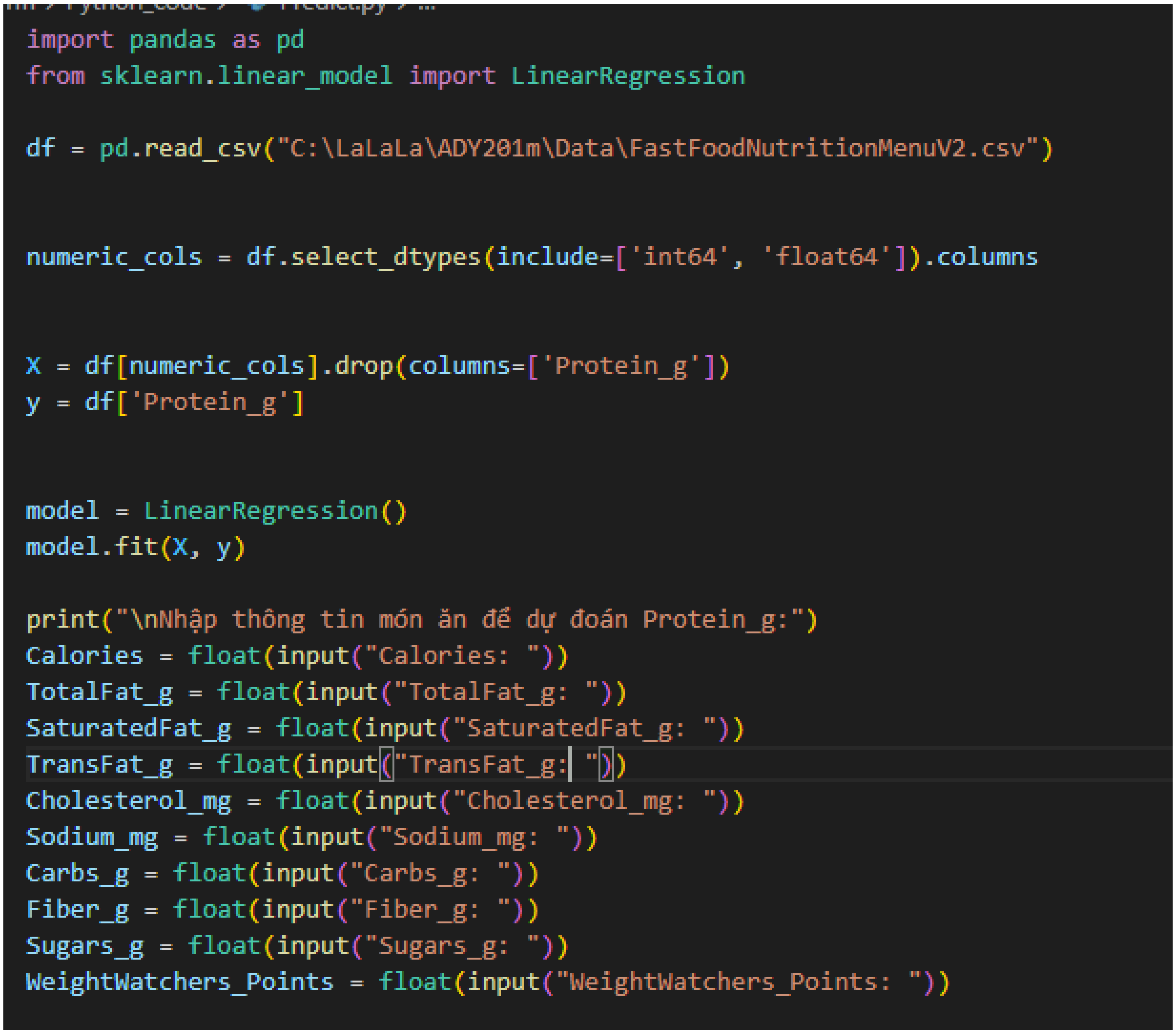


Identify the “best” menu item according to balanced and healthy nutritional criteria.

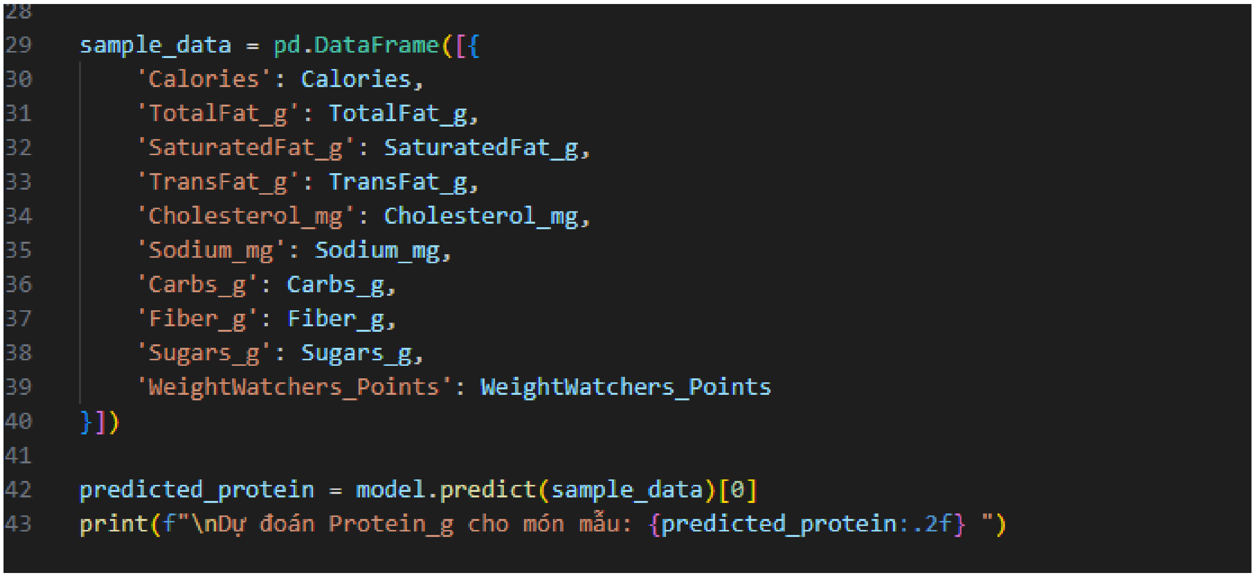


6. Regression

Use scikit-learn library to predict “Protein_g” via other nutrition attributes.



```python
import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.read_csv("C:\LaLaLa\ADY201m\Data\FastFoodNutritionMenuV2.csv")


numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns


X = df[numeric_cols].drop(columns=['Protein_g'])
y = df['Protein_g']


model = LinearRegression()
model.fit(X, y)

print("\nNhập thông tin món ăn để dự đoán Protein_g:")
Calories = float(input("Calories: "))
TotalFat_g = float(input("TotalFat_g: "))
SaturatedFat_g = float(input("SaturatedFat_g: "))
TransFat_g = float(input("TransFat_g: "))
Cholesterol_mg = float(input("Cholesterol_mg: "))
Sodium_mg = float(input("Sodium_mg: "))
Carbs_g = float(input("Carbs_g: "))
Fiber_g = float(input("Fiber_g: "))
Sugars_g = float(input("Sugars_g: "))
WeightWatchers_Points = float(input("WeightWatchers_Points: "))
```

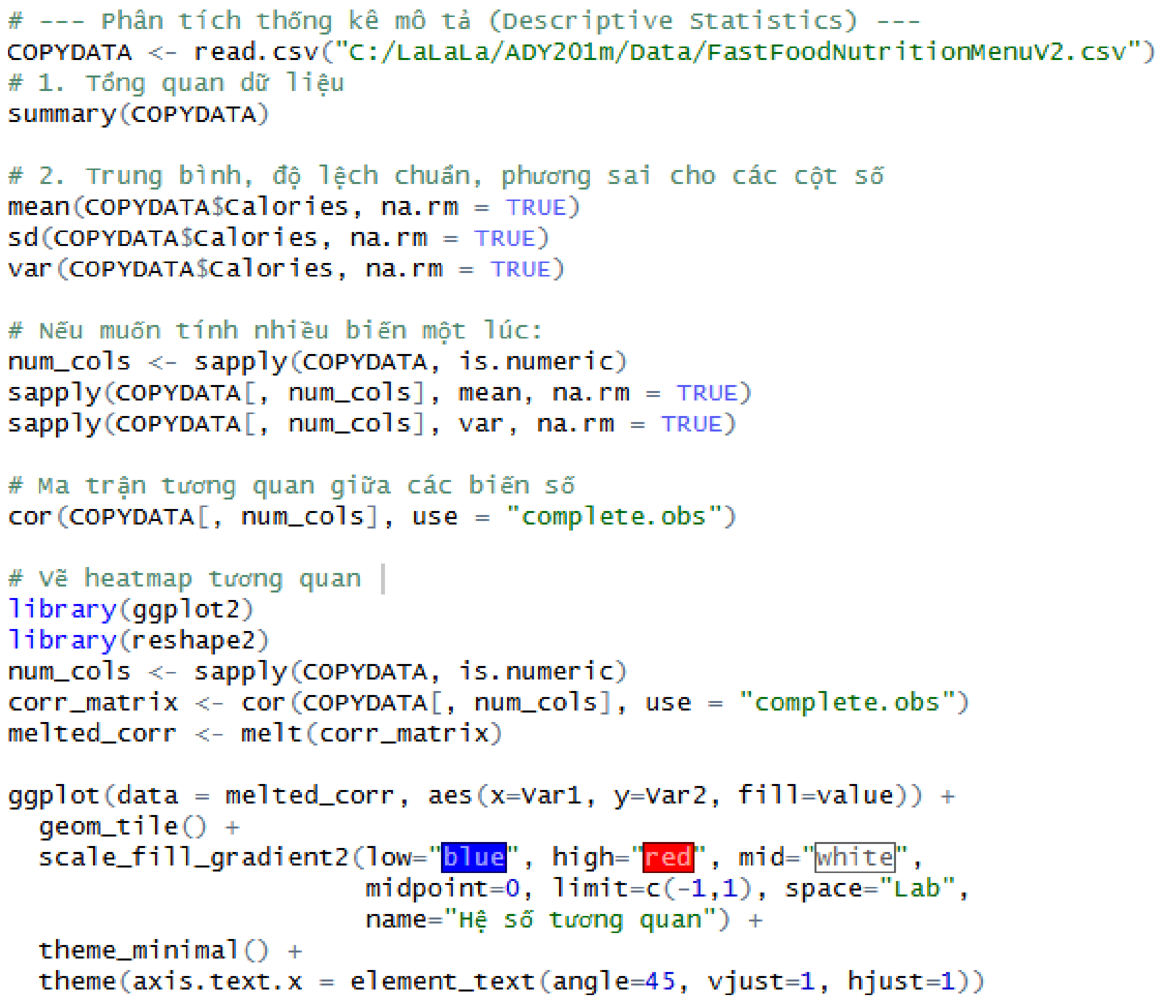


```python

sample_data = pd.DataFrame([{
    'Calories': Calories,
    'TotalFat_g': TotalFat_g,
    'SaturatedFat_g': SaturatedFat_g,
    'TransFat_g': TransFat_g,
    'Cholesterol_mg': Cholesterol_mg,
    'Sodium_mg': Sodium_mg,
    'Carbs_g': Carbs_g,
    'Fiber_g': Fiber_g,
    'Sugars_g': Sugars_g,
    'WeightWatchers_Points': WeightWatchers_Points
}])

predicted_protein = model.predict(sample_data)[0]
print(f"\nDự đoán Protein_g cho món mẫu: {predicted_protein:.2f} ")
```

7. Analysis tool

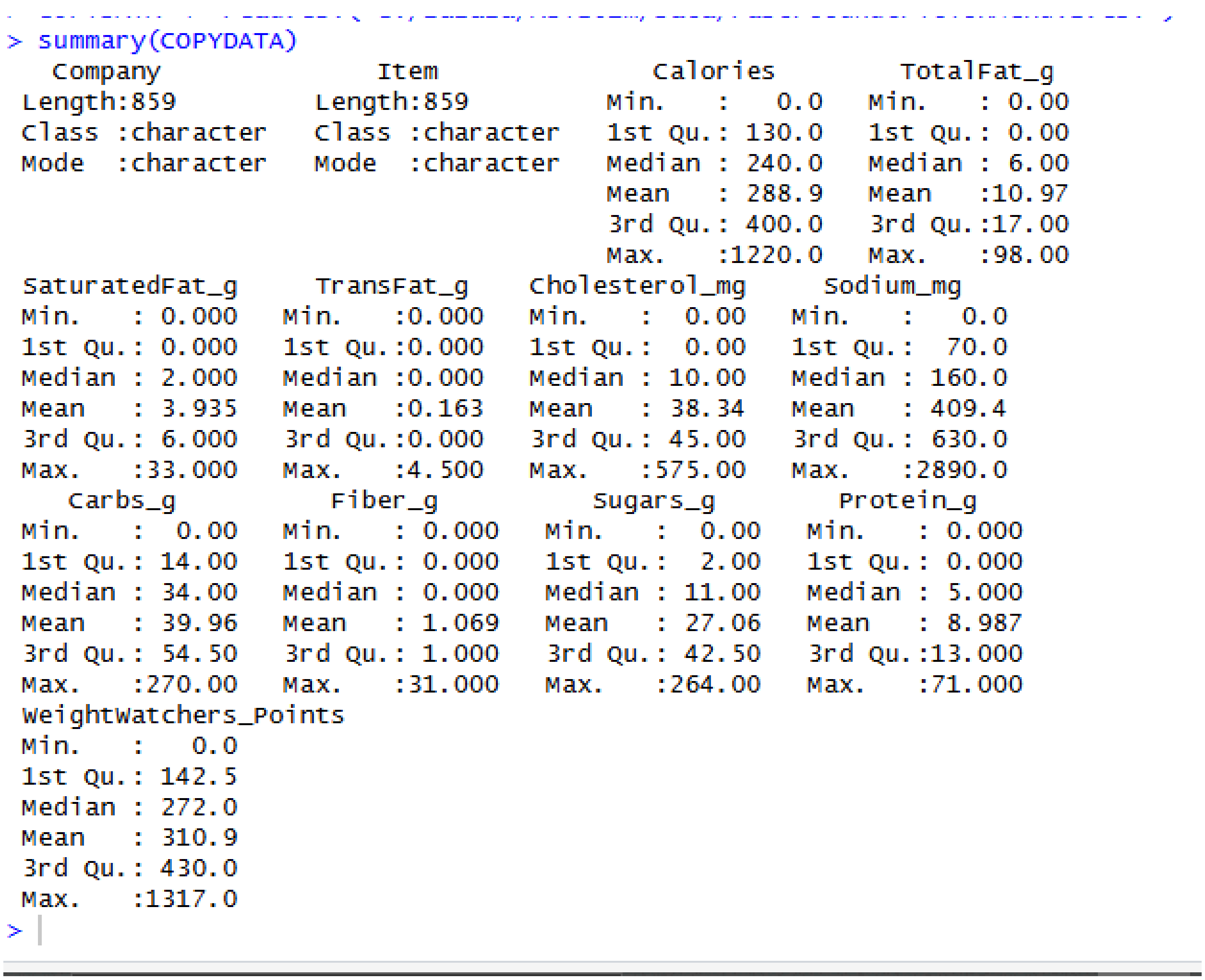
-R Studio was employed to validate the regression model and visualize relationships between dependent and independent variables. The *ggplot2* package was used to plot residuals and regression lines, confirming the linear relationship assumption. Additionally, summary statistics from R supported the findings obtained through Python, reinforcing the consistency of the analysis.

ANALYSIS CODE:

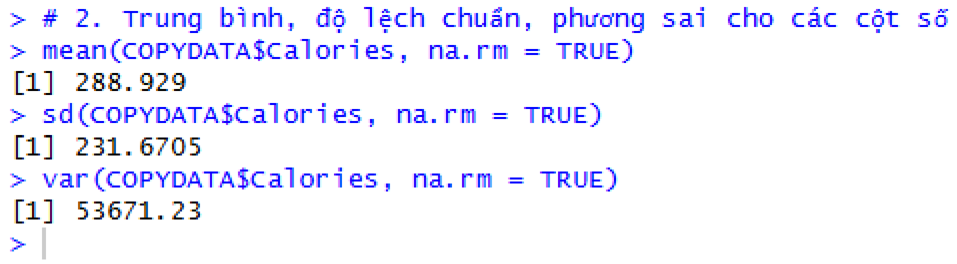


```r
# --- Phân tích thống kê mô tả (Descriptive Statistics) ---
COPYDATA <- read.csv("C:/LaLaLa/ADY201m/Data/FastFoodNutritionMenuV2.csv")
# 1. Tổng quan dữ liệu
summary(COPYDATA)

# 2. Trung bình, độ lệch chuẩn, phương sai cho các cột số
mean(COPYDATA$Calories, na.rm = TRUE)
sd(COPYDATA$Calories, na.rm = TRUE)
var(COPYDATA$Calories, na.rm = TRUE)

# Nếu muốn tính nhiều biến một lúc:
num_cols <- sapply(COPYDATA, is.numeric)
sapply(COPYDATA[, num_cols], mean, na.rm = TRUE)
sapply(COPYDATA[, num_cols], var, na.rm = TRUE)

# Ma trận tương quan giữa các biến số
cor(COPYDATA[, num_cols], use = "complete.obs")

# Vẽ heatmap tương quan
library(ggplot2)
library(reshape2)
num_cols <- sapply(COPYDATA, is.numeric)
corr_matrix <- cor(COPYDATA[, num_cols], use = "complete.obs")
melted_corr <- melt(corr_matrix)

ggplot(data = melted_corr, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low="blue", high="red", mid="white",
                       midpoint=0, limit=c(-1,1), space="Lab",
                       name="Hệ số tương quan") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
```

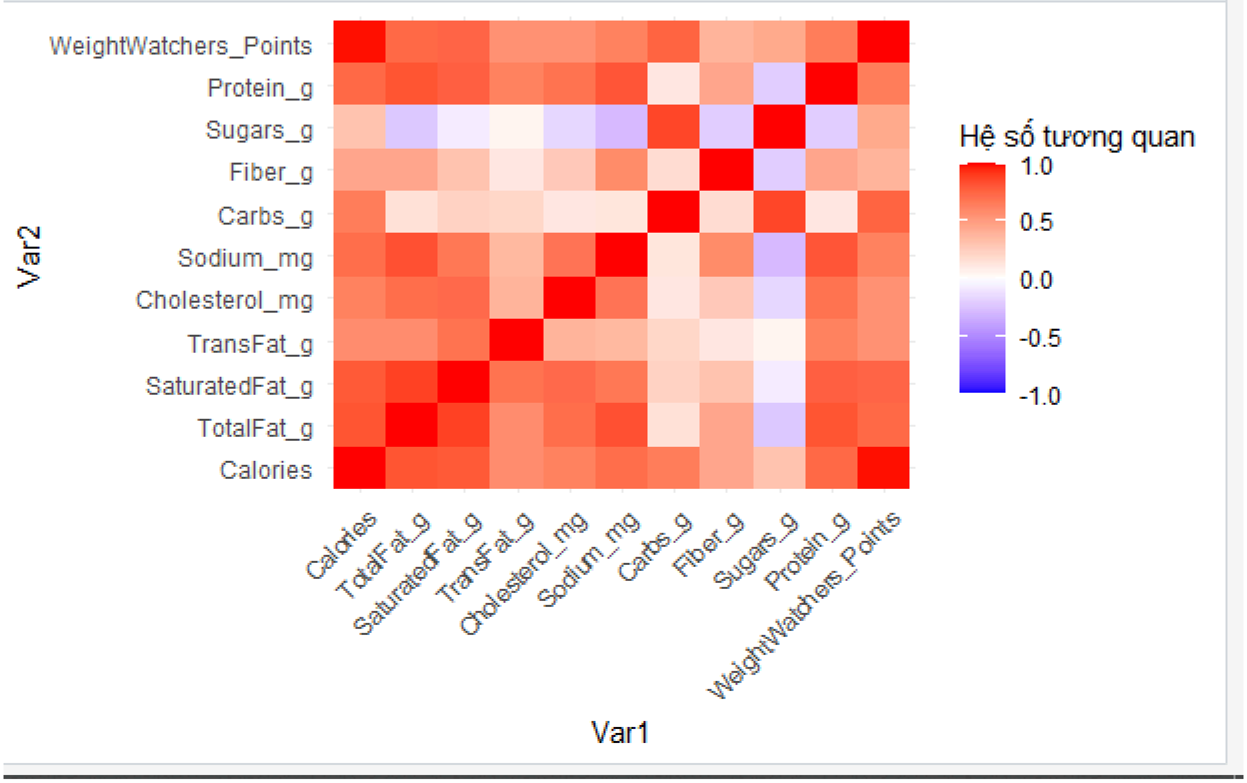
OUTPUT:



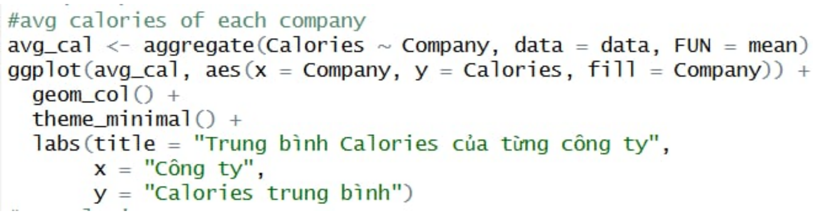
```
> summary(COPYDATA)
   Company               Item              Calories        TotalFat_g   
 Length:859         Length:859         Min.   :   0.0   Min.   : 0.00  
 Class :character   Class :character   1st Qu.: 130.0   1st Qu.: 0.00  
 Mode  :character   Mode  :character   Median : 240.0   Median : 6.00  
                                       Mean   : 288.9   Mean   :10.97  
                                       3rd Qu.: 400.0   3rd Qu.:17.00  
                                       Max.   :1220.0   Max.   :98.00  
 SaturatedFat_g     TransFat_g    Cholesterol_mg     Sodium_mg     
 Min.   : 0.000   Min.   :0.000   Min.   :  0.00   Min.   :   0.0  
 1st Qu.: 0.000   1st Qu.:0.000   1st Qu.:  0.00   1st Qu.:  70.0  
 Median : 2.000   Median :0.000   Median : 10.00   Median : 160.0  
 Mean   : 3.935   Mean   :0.163   Mean   : 38.34   Mean   : 409.4  
 3rd Qu.: 6.000   3rd Qu.:0.000   3rd Qu.: 45.00   3rd Qu.: 630.0  
 Max.   :33.000   Max.   :4.500   Max.   :575.00   Max.   :2890.0  
    Carbs_g          Fiber_g          Sugars_g        Protein_g     
 Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   : 0.000  
 1st Qu.: 14.00   1st Qu.: 0.000   1st Qu.:  2.00   1st Qu.: 0.000  
 Median : 34.00   Median : 0.000   Median : 11.00   Median : 5.000  
 Mean   : 39.96   Mean   : 1.069   Mean   : 27.06   Mean   : 8.987  
 3rd Qu.: 54.50   3rd Qu.: 1.000   3rd Qu.: 42.50   3rd Qu.:13.000  
 Max.   :270.00   Max.   :31.000   Max.   :264.00   Max.   :71.000  
 WeightWatchers_Points
 Min.   :   0.0       
 1st Qu.: 142.5       
 Median : 272.0       
 Mean   : 310.9       
 3rd Qu.: 430.0       
 Max.   :1317.0       
> 
```



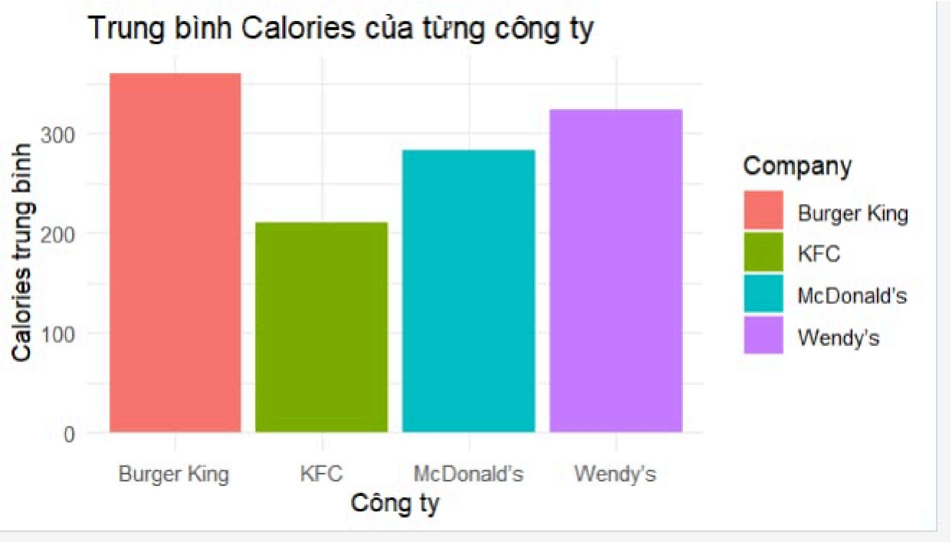
```
> # 2. Trung bình, độ lệch chuẩn, phương sai cho các cột số
> mean(COPYDATA$Calories, na.rm = TRUE)
[1] 288.929
> sd(COPYDATA$Calories, na.rm = TRUE)
[1] 231.6705
> var(COPYDATA$Calories, na.rm = TRUE)
[1] 53671.23
> 
```

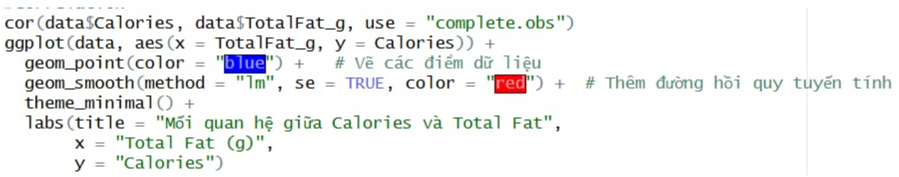


VISUALIZATION CODE:

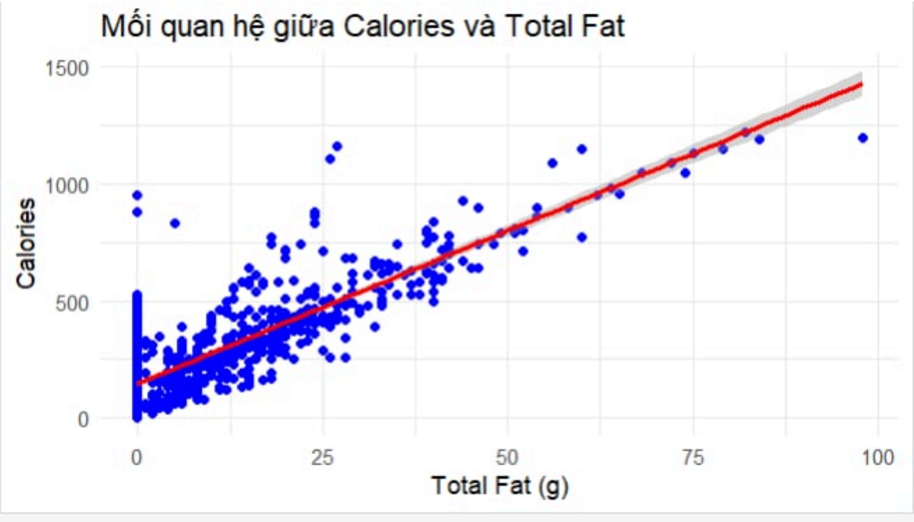


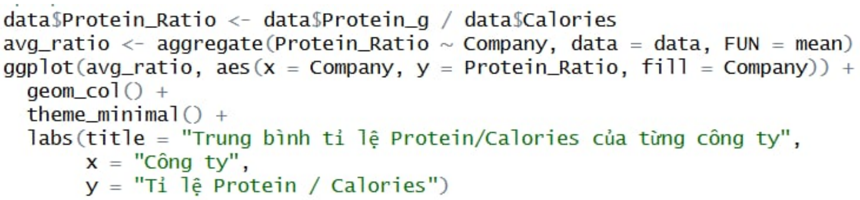
```
#avg calories of each company
avg_cal <- aggregate(Calories ~ Company, data = data, FUN = mean)
ggplot(avg_cal, aes(x = Company, y = Calories, fill = Company)) +
  geom_col() +
  theme_minimal() +
  labs(title = "Trung bình Calories của từng công ty",
       x = "Công ty",
       y = "Calories trung bình")
```



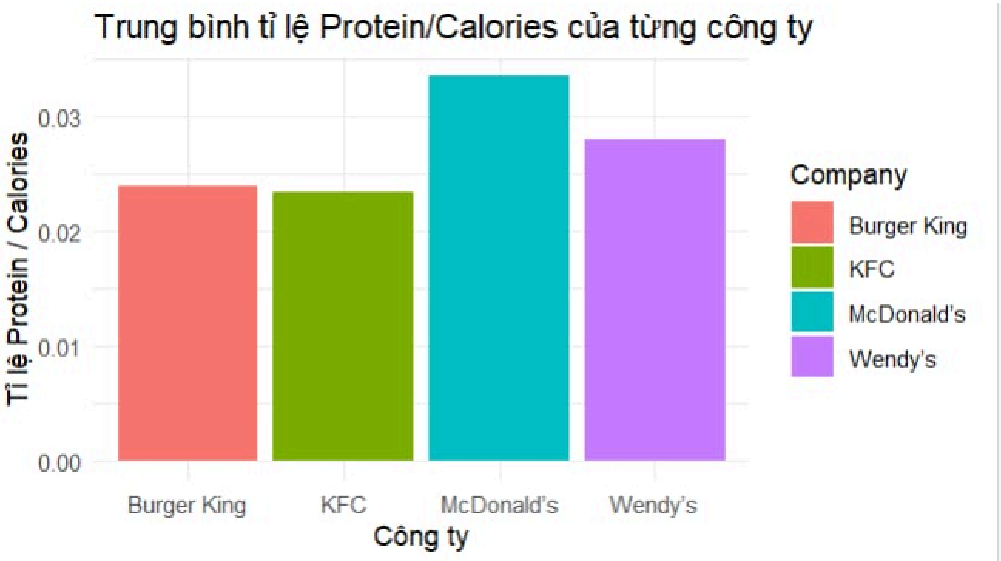


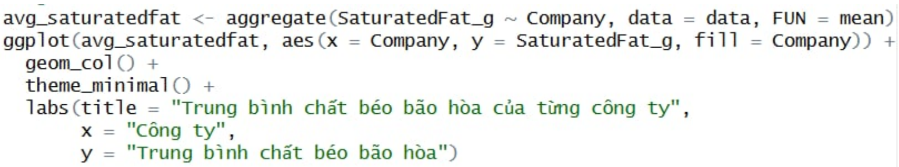
```
cor(data$Calories, data$TotalFat_g, use = "complete.obs")
ggplot(data, aes(x = TotalFat_g, y = Calories)) +
  geom_point(color = "blue") +   # Vẽ các điểm dữ liệu
  geom_smooth(method = "lm", se = TRUE, color = "red") +  # Thêm đường hồi quy tuyến tính
  theme_minimal() +
  labs(title = "Mối quan hệ giữa Calories và Total Fat",
       x = "Total Fat (g)",
       y = "Calories")
```



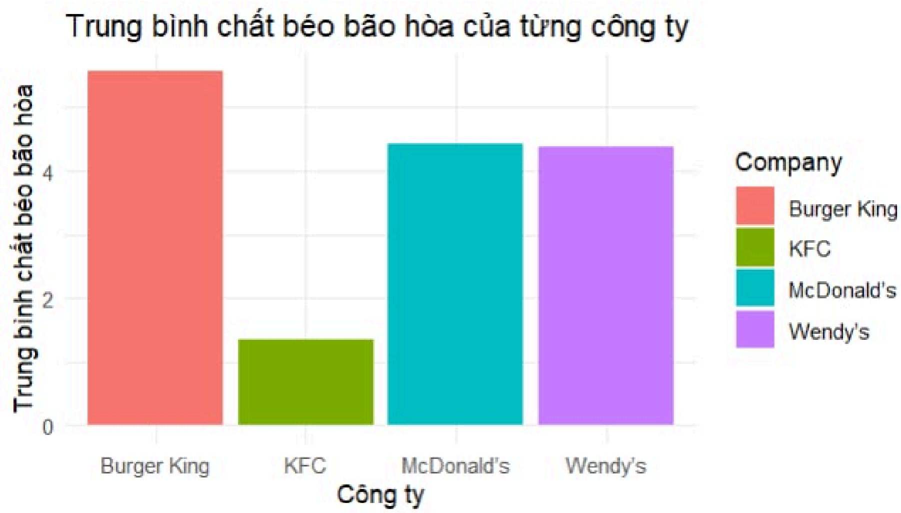


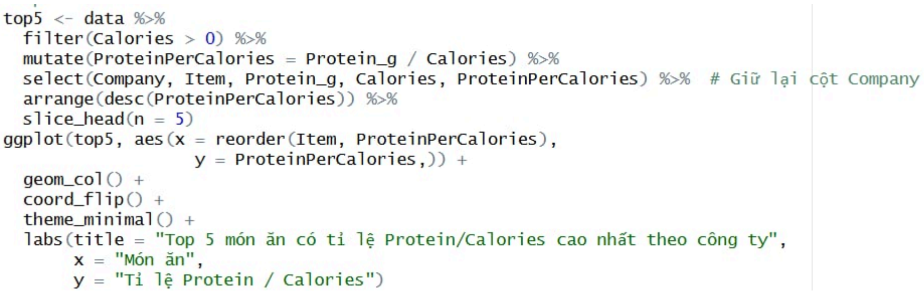
```
data$Protein_Ratio <- data$Protein_g / data$Calories
avg_ratio <- aggregate(Protein_Ratio ~ Company, data = data, FUN = mean)
ggplot(avg_ratio, aes(x = Company, y = Protein_Ratio, fill = Company)) +
  geom_col() +
  theme_minimal() +
  labs(title = "Trung bình tỉ lệ Protein/Calories của từng công ty",
       x = "Công ty",
       y = "Tỉ lệ Protein / Calories")
```



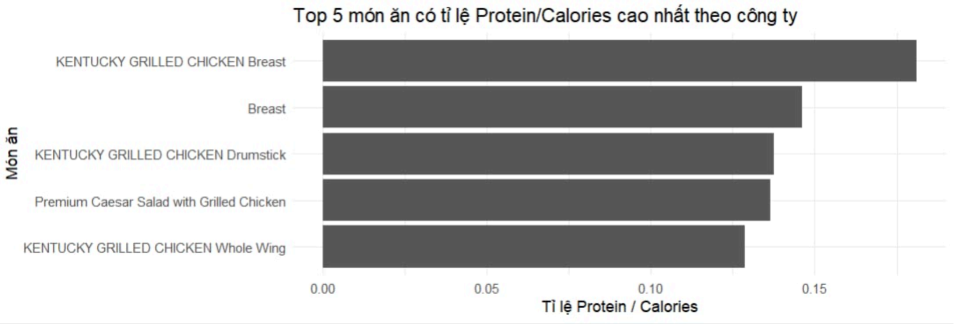


```
avg_saturatedfat <- aggregate(SaturatedFat_g ~ Company, data = data, FUN = mean)
ggplot(avg_saturatedfat, aes(x = Company, y = SaturatedFat_g, fill = Company)) +
  geom_col() +
  theme_minimal() +
  labs(title = "Trung bình chất béo bão hòa của từng công ty",
       x = "Công ty",
       y = "Trung bình chất béo bão hòa")
```





```
top5 <- data %>%
  filter(Calories > 0) %>%
  mutate(ProteinPerCalories = Protein_g / Calories) %>%
  select(Company, Item, Protein_g, Calories, ProteinPerCalories) %>%  # Giữ lại cột Company
  arrange(desc(ProteinPerCalories)) %>%
  slice_head(n = 5)
ggplot(top5, aes(x = reorder(Item, ProteinPerCalories),
                 y = ProteinPerCalories,)) +
  geom_col() +
  coord_flip() +
  theme_minimal() +
  labs(title = "Top 5 món ăn có tỉ lệ Protein/Calories cao nhất theo công ty",
       x = "Món ăn",
       y = "Tỉ lệ Protein / Calories")
```


Top 5 món ăn có tỉ lệ Protein/Calories cao nhất theo công ty
Món ăn
KENTUCKY GRILLED CHICKEN Breast
Breast
KENTUCKY GRILLED CHICKEN Drumstick
Premium Caesar Salad with Grilled Chicken
KENTUCKY GRILLED CHICKEN Whole Wing
Tỉ lệ Protein / Calories