```
from google.colab import files
uploaded = files.upload()
```

Choose files menu.csv

- **menu.csv**(text/csv) - 29988 bytes, last modified: 16/03/2025 - 100% done

```
Saving menu.csv to menu.csv
```

```
import pandas as pd

# File name (ensure the correct file name is used)
file_path = "menu.csv"

# Load the CSV file
df = pd.read_csv(file_path)

# Display the first 5 rows
df.head()
```

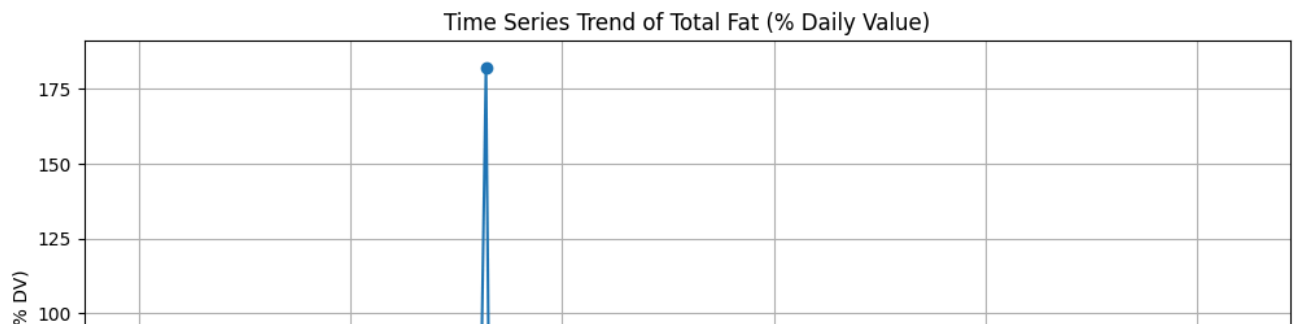
	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Sat
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	

5 rows × 24 columns

```
df.describe(include='all')
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
df["Total Fat (% Daily Value)"].plot(marker="o", linestyle="-")  # Replace with relevant
plt.title("Time Series Trend of Total Fat (% Daily Value)")
plt.xlabel("Date")
plt.ylabel("Total Fat (% DV)")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```




```
import pandas as pd

# Load dataset
file_path = "menu.csv"  # Ensure the correct filename
df = pd.read_csv(file_path)

# Group by Category and calculate average calories, fat, and protein
category_analysis = df.groupby("Category")[["Calories", "Total Fat", "Protein"]].mean()
```

```
# Display the results
print(category_analysis)
```

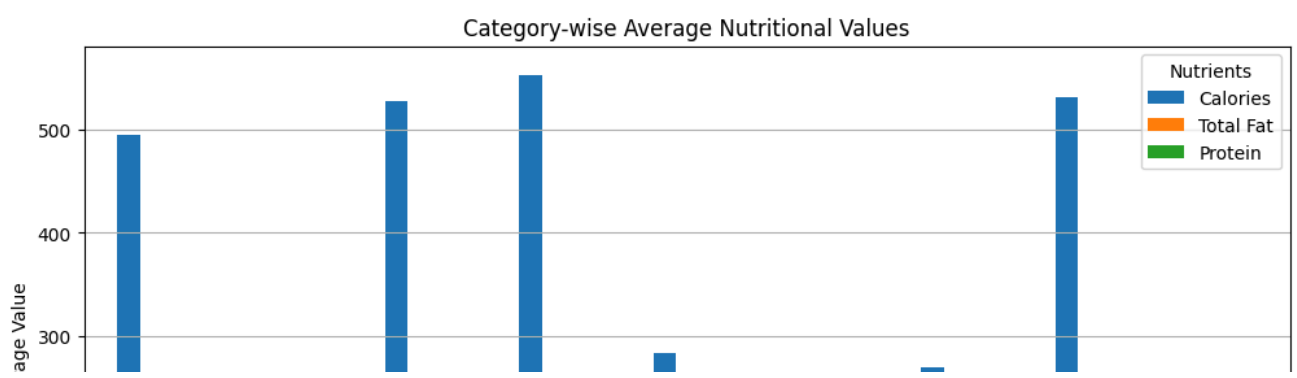
```
                      Calories  Total Fat    Protein
Category
Beef & Pork         494.000000  24.866667  27.333333
Beverages           113.703704   0.092593   1.333333
Breakfast           526.666667  27.690476  19.857143
Chicken & Fish      552.962963  26.962963  29.111111
Coffee & Tea        283.894737   8.021053   8.863158
Desserts            222.142857   7.357143   4.000000
Salads              270.000000  11.750000  19.833333
Smoothies & Shakes  531.428571  14.125000  10.857143
Snacks & Sides      245.769231  10.538462   8.384615
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
file_path = "menu.csv"  # Ensure the correct filename
df = pd.read_csv(file_path)

# Group by Category and calculate average Calories, Fat, and Protein
category_analysis = df.groupby("Category")[["Calories", "Total Fat", "Protein"]].mean()

# Plot bar chart
category_analysis.plot(kind="bar", figsize=(12, 6))
plt.title("Category-wise Average Nutritional Values")
plt.xlabel("Category")
plt.ylabel("Average Value")
plt.xticks(rotation=45)
plt.grid(axis="y")
plt.legend(title="Nutrients")
plt.show()
```




```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
file_path = "menu.csv"  # Ensure correct filename
df = pd.read_csv(file_path)

# Set plot style
sns.set_style("whitegrid")

# 1. **Calories Distribution**
plt.figure(figsize=(10, 5))
sns.histplot(df["Calories"], bins=20, kde=True, color="blue")
plt.title("Distribution of Calories")
```

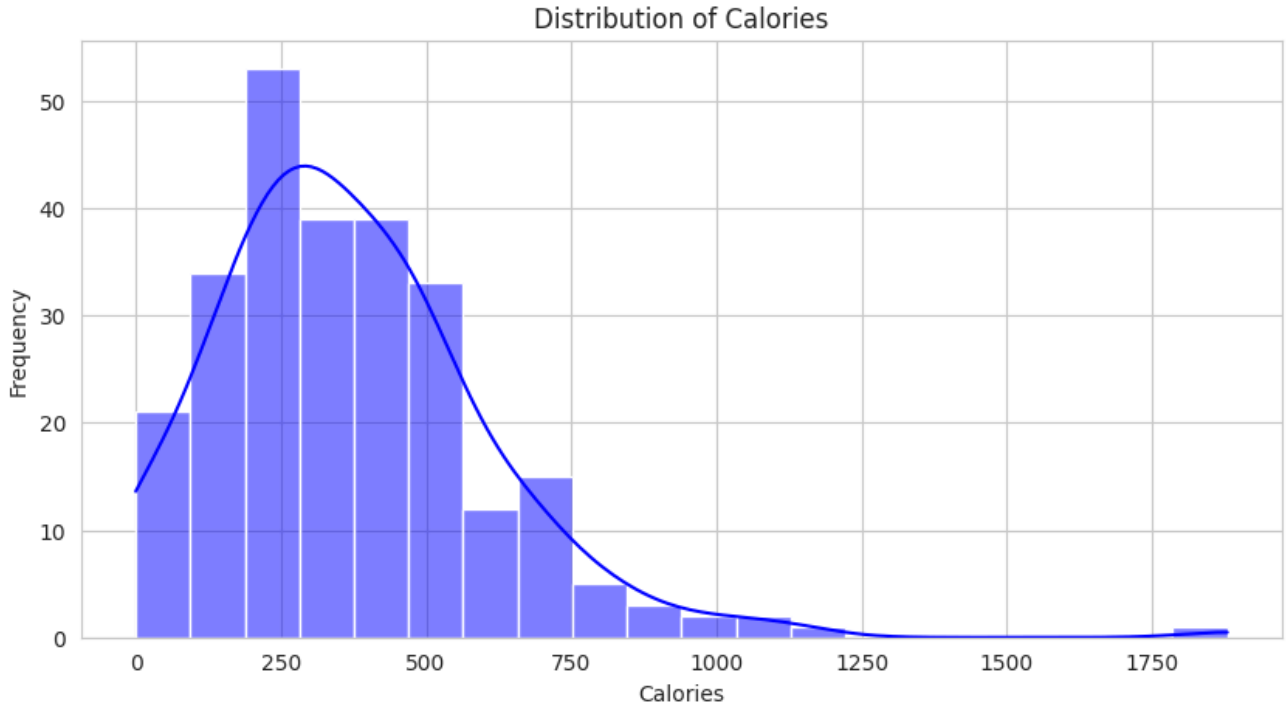
```
plt.xlabel("Calories")
plt.ylabel("Frequency")
plt.show()

# 2. **Category-wise Calories**
plt.figure(figsize=(12, 6))
sns.boxplot(x="Category", y="Calories", data=df, palette="Set2")
plt.title("Calories Distribution Across Categories")
plt.xticks(rotation=45)
plt.show()

# 3. **Correlation Heatmap**
plt.figure(figsize=(10, 6))
sns.heatmap(df[["Calories", "Total Fat", "Saturated Fat", "Cholesterol", "Sodium", "Carbo
plt.title("Correlation Heatmap of Nutritional Values")
plt.show()

# 4. **Total Fat vs Calories**
plt.figure(figsize=(8, 5))
sns.scatterplot(x="Total Fat", y="Calories", data=df, color="red")
plt.title("Total Fat vs Calories")
plt.xlabel("Total Fat (g)")
plt.ylabel("Calories")
plt.show()

# 5. **Category Count Plot**
plt.figure(figsize=(10, 5))
sns.countplot(y="Category", data=df, palette="viridis")
plt.title("Count of Items in Each Category")
plt.xlabel("Count")
plt.ylabel("Category")
plt.show()
```




```
<ipython-input-54-076b3f276d65>:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed ir

  sns.boxplot(x="Category", y="Calories", data=df, palette="Set2")
```