import pandas as pd

import matplotlib.pyplot as plt

# CSV yükle

df = pd.read\_csv("starbucks.csv")

# Konsolda sütunların kesilmemesi için ayarlar

pd.set\_option("display.max\_columns", None)

pd.set\_option("display.width", None)

# 1) Genel bilgi

print("==== INFO ====")

print(df.info())

# 2) Sayısal sütunların istatistik özeti

print("\n==== NUMERICAL DESCRIBE ====")

print(df.describe())

# 3) Kategorik sütunların özet istatistikleri

print("\n==== CATEGORICAL DESCRIBE ====")

print(df.describe(include="object"))

# 4) Tüm sütun isimlerini yazdır

print("\n==== COLUMNS ====")

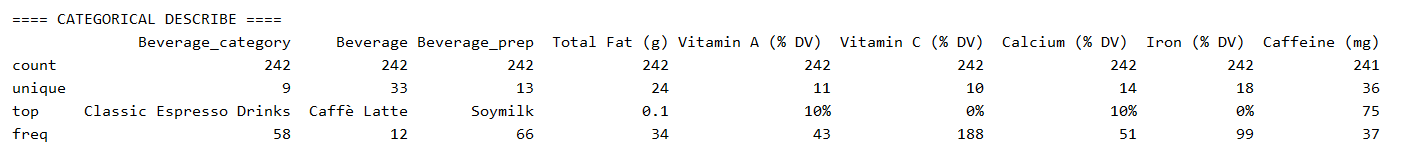
print(df.columns.tolist())

metin, menü, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.



*# Visualization*plt.figure(figsize=(10,6))  
avg\_calories.plot(kind="bar", color="darkorange", edgecolor="black")  
plt.title("Average Calories per Beverage Category", fontsize=14)  
plt.ylabel("Average Calories", fontsize=12)  
plt.xlabel("Beverage Category", fontsize=12)  
plt.xticks(rotation=45, ha="right")  
plt.tight\_layout()  
plt.show()

metin, ekran görüntüsü, yazı tipi, diyagram içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
*# Load CSV*df = pd.read\_csv("starbucks.csv")  
  
*# Pandas display settings*pd.set\_option("display.max\_columns", None)  
pd.set\_option("display.width", None)  
  
*# Calculate mean sugar per category for ordering*order = (  
 df.groupby("Beverage\_category")[" Sugars (g)"]  
 .mean()  
 .sort\_values(ascending=False)  
 .index  
)  
  
*# ==== ANALYSIS: Sugar distribution per category (Boxplot with ordered categories) ====*plt.figure(figsize=(12,6))  
sns.boxplot(data=df, x="Beverage\_category", y=" Sugars (g)", order=order, palette="Set2")  
  
plt.title("Sugar Distribution per Beverage Category (Ordered by Mean)", fontsize=14)  
plt.ylabel("Sugar (g)", fontsize=12)  
plt.xlabel("Beverage Category", fontsize=12)  
plt.xticks(rotation=45, ha="right")  
plt.tight\_layout()  
plt.show()

metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
*# Load CSV*df = pd.read\_csv("starbucks.csv")  
  
*# Clean caffeine column*df["Caffeine (mg)"] = pd.to\_numeric(df["Caffeine (mg)"], errors="coerce")  
  
*# Calculate average caffeine per product (Beverage)*avg\_caffeine\_product = (  
 df.groupby("Beverage")["Caffeine (mg)"]  
 .mean()  
 .sort\_values(ascending=False)  
)  
  
*# Select top 10*top10 = avg\_caffeine\_product.head(100).reset\_index()  
  
*# ==== Visualization ====*plt.figure(figsize=(10,6))  
ax = sns.barplot(data=top10, x="Caffeine (mg)", y="Beverage", palette="coolwarm")  
  
*# Add labels to bars*for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", fontsize=10, padding=2)  
  
plt.title("Top 10 Beverages by Average Caffeine", fontsize=14)  
plt.xlabel("Average Caffeine (mg)", fontsize=12)  
plt.ylabel("Beverage", fontsize=12)  
plt.tight\_layout()  
plt.show()

metin, ekran görüntüsü, yazı tipi, diyagram içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
*# Load CSV*df = pd.read\_csv("starbucks.csv")  
  
*# Convert caffeine to numeric*df["Caffeine (mg)"] = pd.to\_numeric(df["Caffeine (mg)"], errors="coerce")  
  
*# --- Define thresholds (25th / 75th percentile) ---*calorie\_low = df["Calories"].quantile(0.25)  
calorie\_high = df["Calories"].quantile(0.75)  
  
sugar\_low = df[" Sugars (g)"].quantile(0.25)  
sugar\_high = df[" Sugars (g)"].quantile(0.75)  
  
caffeine\_high = df["Caffeine (mg)"].quantile(0.75)  
  
*# --- Healthiest drinks ---*healthiest = df[  
 (df["Calories"] <= calorie\_low) &  
 (df[" Sugars (g)"] <= sugar\_low) &  
 (df["Caffeine (mg)"] >= caffeine\_high)  
][["Beverage", "Calories", " Sugars (g)", "Caffeine (mg)"]]  
  
*# --- Riskiest drinks ---*riskiest = df[  
 (df["Calories"] >= calorie\_high) &  
 (df[" Sugars (g)"] >= sugar\_high)  
][["Beverage", "Calories", " Sugars (g)", "Caffeine (mg)"]]  
  
print("==== HEALTHIEST DRINKS ====")  
print(healthiest.head(10))  
  
print("\n==== RISKIEST DRINKS ====")  
print(riskiest.head(10))  
  
*# --- Visualization ---*fig, axes = plt.subplots(1, 2, figsize=(16,6))  
  
*# Healthiest chart*sns.barplot(  
 data=healthiest.head(10).sort\_values("Caffeine (mg)", ascending=False),  
 x="Caffeine (mg)", y="Beverage", ax=axes[0], palette="Greens"  
)  
axes[0].set\_title("Top 10 Healthiest Drinks", fontsize=14)  
axes[0].set\_xlabel("Caffeine (mg)")  
axes[0].set\_ylabel("Beverage")  
  
*# Add values*for container in axes[0].containers:  
 axes[0].bar\_label(container, fmt="%.0f", label\_type="edge", fontsize=9)  
  
*# Riskiest chart*sns.barplot(  
 data=riskiest.head(10).sort\_values("Calories", ascending=False),  
 x="Calories", y="Beverage", ax=axes[1], palette="Reds"  
)  
axes[1].set\_title("Top 10 Riskiest Drinks", fontsize=14)  
axes[1].set\_xlabel("Calories")  
axes[1].set\_ylabel("Beverage")  
  
*# Add values*for container in axes[1].containers:  
 axes[1].bar\_label(container, fmt="%.0f", label\_type="edge", fontsize=9)  
  
plt.tight\_layout()  
plt.show()

ekran görüntüsü, diyagram, metin, çizgi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
*# Load CSV*df = pd.read\_csv("starbucks.csv")  
  
*# Select Top 10 lowest calorie drinks*lowest\_cal = df.nsmallest(50, "Calories")[["Beverage", "Calories", " Sugars (g)", "Caffeine (mg)"]]  
  
print("==== TOP 10 LOWEST CALORIE DRINKS ====")  
print(lowest\_cal)  
  
*# Visualization*plt.figure(figsize=(10,6))  
ax = sns.barplot(data=lowest\_cal, x="Calories", y="Beverage", palette="Greens\_r")  
  
plt.title("Top 10 Lowest Calorie Beverages", fontsize=14)  
plt.xlabel("Calories", fontsize=12)  
plt.ylabel("Beverage", fontsize=12)  
  
*# Add values on bars*for container in ax.containers:  
 ax.bar\_label(container, fmt="%.0f", label\_type="edge", fontsize=9)  
  
plt.tight\_layout()  
plt.show()

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
*# Load CSV*df = pd.read\_csv("starbucks.csv")  
  
*# Convert caffeine to numeric and replace NaN with 0*df["Caffeine (mg)"] = pd.to\_numeric(df["Caffeine (mg)"], errors="coerce").fillna(0)  
  
*# --- Top 5 Sugar ---*top5\_sugar = df.nlargest(5, " Sugars (g)")[["Beverage", " Sugars (g)"]].sort\_values(" Sugars (g)")  
  
*# --- Top 5 Caffeine ---*top5\_caffeine = df.nlargest(5, "Caffeine (mg)")[["Beverage", "Caffeine (mg)"]].sort\_values("Caffeine (mg)")  
  
*# --- Top 5 Calories ---*top5\_calories = df.nlargest(5, "Calories")[["Beverage", "Calories"]].sort\_values("Calories")  
  
*# --- Visualization ---*fig, axes = plt.subplots(1, 3, figsize=(18,6))  
  
*# Sugar*sns.barplot(data=top5\_sugar, x=" Sugars (g)", y="Beverage", ax=axes[0], palette="Reds\_r")  
axes[0].set\_title("Top 5 Sugar (g)")  
for container in axes[0].containers:  
 axes[0].bar\_label(container, fmt="%.0f", fontsize=8)  
  
*# Caffeine*sns.barplot(data=top5\_caffeine, x="Caffeine (mg)", y="Beverage", ax=axes[1], palette="Blues\_r")  
axes[1].set\_title("Top 5 Caffeine (mg)")  
for container in axes[1].containers:  
 axes[1].bar\_label(container, fmt="%.0f", fontsize=8)  
  
*# Calories*sns.barplot(data=top5\_calories, x="Calories", y="Beverage", ax=axes[2], palette="Greens\_r")  
axes[2].set\_title("Top 5 Calories")  
for container in axes[2].containers:  
 axes[2].bar\_label(container, fmt="%.0f", fontsize=8)  
  
plt.tight\_layout()  
plt.show()

ekran görüntüsü, diyagram, metin, çizgi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.