#Load and inspect the data:

import pandas as pd

data=pd.read\_csv(r"C:\Users\kkhus*\D*ownloads\UnifiedMentor\Coffee-Sales\index\_1*.*csv")

print(data.head)

 # Check for missing values

print(data.isnull().sum())

# Convert Date to datetime type

data['date'] = pd.to\_datetime(data['date'], *errors*='coerce')

print(data['date'].head())

print(data['date'].isna().sum())

# Extract month and year from the Date

data['Month'] = data['date'].dt.month

data['Year'] = data['date'].dt.year

 # Drop the original Date column

import matplotlib.pyplot as plt

import seaborn as sns

# Set a clean style

sns.set\_theme(*style*="whitegrid")

plt.figure(*figsize*=(12, 7))

# Lineplot with markers and better palette

sns.lineplot(

*data*=data,

*x*='Month',

*y*='Sales',

*hue*='Year',

*marker*="o",

*palette*="tab10",

*linewidth*=2.5

)

# Titles and labels

plt.title("📈 Monthly Sales Over Years", *fontsize*=18, *weight*='bold', *pad*=20)

plt.xlabel("Month", *fontsize*=14)

plt.ylabel("Sales", *fontsize*=14)

# Move legend outside

plt.legend(*title*="Year", *fontsize*=12, *title\_fontsize*=13, *bbox\_to\_anchor*=(1.05, 1), *loc*='upper left')

# Improve ticks

plt.xticks(range(1, 13),

           ["Jan", "Feb", "Mar", "Apr", "May", "Jun",

            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"],

*fontsize*=12)

plt.yticks(*fontsize*=12)

# Add light grid

plt.grid(True, *linestyle*="--", *alpha*=0.6)

plt.tight\_layout()

plt.show()

# Sales by store

# Set theme

sns.set\_theme(*style*="whitegrid")

plt.figure(*figsize*=(12, 7))

# Sort stores by sales for cleaner display

sorted\_data = data.groupby("Store", *as\_index*=False)["Sales"].sum().sort\_values("Sales", *ascending*=False)

# Barplot

ax = sns.barplot(

*data*=sorted\_data,

*x*="Store",

*y*="Sales",

*palette*="Set2"

)

# Add value labels on bars

for p in ax.patches:

    ax.annotate(f'{p.get\_height():,.0f}',

                (p.get\_x() + p.get\_width() / 2., p.get\_height()),

*ha*='center', *va*='bottom',

*fontsize*=11, *fontweight*='bold', *color*='black')

# Title & labels

plt.title("🏬 Sales by Store", *fontsize*=18, *weight*="bold", *pad*=20)

plt.xlabel("Store", *fontsize*=14)

plt.ylabel("Sales", *fontsize*=14)

# Rotate store labels if needed

plt.xticks(*rotation*=30, *ha*="right", *fontsize*=12)

plt.yticks(*fontsize*=12)

plt.tight\_layout()

plt.show()

 # Sales by product

# Set theme

sns.set\_theme(*style*="whitegrid")

plt.figure(*figsize*=(12, 7))

# Sort products by sales for clarity

sorted\_data = (

    data.groupby("coffee\_name", *as\_index*=False)["Sales"]

    .sum()

    .sort\_values("Sales", *ascending*=False)

)

# Horizontal barplot

ax = sns.barplot(

*data*=sorted\_data,

*y*="coffee\_name",

*x*="Sales",

*palette*="crest"

)

# Add data labels

for p in ax.patches:

    ax.annotate(

        f'{p.get\_width():,.0f}',  # format with commas

        (p.get\_width(), p.get\_y() + p.get\_height() / 2),

*ha*="left", *va*="center",

*fontsize*=11, *fontweight*="bold",

*color*="black", *xytext*=(5, 0),

*textcoords*="offset points"

    )

# Titles and labels

plt.title("☕ Sales by Coffee Product", *fontsize*=18, *weight*="bold", *pad*=20)

plt.xlabel("Total Sales", *fontsize*=14)

plt.ylabel("Coffee Name", *fontsize*=14)

# Ticks styling

plt.xticks(*fontsize*=12)

plt.yticks(*fontsize*=12)

plt.tight\_layout()

plt.show()

#monthly sales

monthly\_sales = (

    data.groupby(['coffee\_name', 'Month'])

        .count()['Sales']  # use 'Sales' instead of 'date'

        .reset\_index()

        .rename(*columns*={'Sales':'count'})

        .pivot(*index*='Month', *columns*='coffee\_name', *values*='count')

        .reset\_index()

)

monthly\_sales

monthly\_sales.describe().T.loc[:,['min','max']]

plt.figure(*figsize*=(12,6))

sns.lineplot(*data*=monthly\_sales)

plt.legend(*loc*='upper left')

plt.xticks(range(len(monthly\_sales['Month'])),monthly\_sales['Month'],*size*='small')

# Extract weekday name

data['weekday'] = data['date'].dt.day\_name()

#weekly sales

weekday\_sales = (

    data.groupby('weekday')

        .count()['Sales']  # use a column that exists instead of 'date'

        .reset\_index()

        .rename(*columns*={'Sales':'count'})

)

# Optionally, order weekdays

weekday\_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

weekday\_sales['weekday'] = pd.Categorical(weekday\_sales['weekday'], *categories*=weekday\_order, *ordered*=True)

weekday\_sales = weekday\_sales.sort\_values('weekday')

weekday\_sales

sns.set\_theme(*style*="whitegrid")

plt.figure(*figsize*=(12,6))

# Barplot

ax = sns.barplot(

*data*=weekday\_sales,

*x*='weekday',

*y*='count',

*color*='steelblue'

)

# Add data labels

for p in ax.patches:

    ax.annotate(

        f'{p.get\_height():,}',

        (p.get\_x() + p.get\_width() / 2., p.get\_height()),

*ha*='center', *va*='bottom',

*fontsize*=11, *fontweight*='bold'

    )

# Titles and labels

plt.title("📊 Sales by Weekday", *fontsize*=16, *weight*='bold', *pad*=15)

plt.xlabel("Weekday", *fontsize*=12)

plt.ylabel("Sales Count", *fontsize*=12)

# Rotate x-ticks if needed

plt.xticks(*fontsize*=12)

plt.tight\_layout()

plt.show()

#Cash or Card

data['cash\_type'].hist()

#coffee type proportions in sales hourly

def extract\_hour(*dt\_str*):

    try:

        mins, secs = map(float, *dt\_str*.split(':'))

        total\_hours = int(mins // 60)  # convert cumulative minutes to hours

        return total\_hours

    except:

        return None

# Ensure Sales is numeric

# Drop invalid rows

hourly\_data = data.dropna(*subset*=['Sales', 'hour'])

hourly\_data['hour'] = hourly\_data['hour'].astype(int)

# Group by hour and sum Sales

hourly\_sales = hourly\_data.groupby('hour')[['Sales']].sum().reset\_index()

hourly\_sales

#show hourly sales chart

plt.figure(*figsize*=(12,6))

# Barplot

ax = sns.barplot(*data*=hourly\_sales, *x*='hour', *y*='Sales', *color*='steelblue')

# Add data labels

for p in ax.patches:

    ax.annotate(

        f'{p.get\_height():.0f}',

        (p.get\_x() + p.get\_width() / 2., p.get\_height()),

*ha*='center', *va*='bottom',

*fontsize*=11, *fontweight*='bold'

    )

plt.title("🕒 Hourly Sales", *fontsize*=16, *weight*='bold')

plt.xlabel("Hour of Day")

plt.ylabel("Sales")

plt.xticks(*rotation*=0)

plt.tight\_layout()

plt.show()

#Hourly Sales by Coffee Type

hourly\_sales\_by\_coffee=data.groupby(['hour','coffee\_name']).count()['date'].reset\_index().rename(*columns*={'date':'count'}).pivot(*index*='hour',

*columns*='coffee\_name',*values*='count').fillna(0).reset\_index()

hourly\_sales\_by\_coffee

fig, axs = plt.subplots(2, 4, *figsize*=(20, 10))

 # Flatten the array of subplots for easy iteration

axs = axs.flatten()

 # Loop through each column in the DataFrame, skipping the'Index' column

for i, column in enumerate(hourly\_sales\_by\_coffee.columns[1:]):

 # Skip the first column ('Index')

    axs[i].bar(hourly\_sales\_by\_coffee['hour'],

    hourly\_sales\_by\_coffee[column])

    axs[i].set\_title(f'{column}')

    axs[i].set\_xlabel('Hour')

 #axs[i].set\_ylabel('Sales')

plt.tight\_layout()

 # Show the plot

plt.show()