import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

data=pd.read\_csv("market.csv")

print(data.shape) //(1000, 17)

data.isnull().sum()

#gender count

plt.figure(figsize=(14,6))

plt.style.use('fivethirtyeight')

ax= sns.countplot('Gender', data=data , palette = 'copper')

ax.set\_xlabel(xlabel= "Gender",fontsize=18)

ax.set\_ylabel(ylabel = "Gender count", fontsize = 18)

ax.set\_title(label = "Gender count in supermarket", fontsize = 20)

plt.show()

A graph of a person's gender

Description automatically generated with medium confidence

#Customer type

plt.style.use('ggplot')

plt.figure(figsize= (14,6))

ax = sns.countplot(x = "Customer type", data = data, palette = "rocket\_r")

ax.set\_title("Type of customers", fontsize = 25)

ax.set\_xlabel("Customer type", fontsize = 16)

ax.set\_ylabel("Customer Count", fontsize = 16)

A screenshot of a graph

Description automatically generated

# type of customer in all branch combined

plt.figure(figsize=(14,6))

plt.style.use('classic')

ax = sns.countplot(x = "Customer type", hue = "Branch", data = data, palette= "rocket\_r")

ax.set\_title(label = "Customer type in different branch", fontsize = 25)

ax.set\_xlabel(xlabel = "Branches", fontsize = 16)

ax.set\_ylabel(ylabel = "Customer Count", fontsize = 16)

A chart of different colors

Description automatically generated with medium confidence

#the different payment methods used.

plt.figure(figsize = (14,6))

ax = sns.countplot(x = "Payment", data = data, palette = "tab20")

ax.set\_title(label = "Payment methods of customers ", fontsize= 25)

ax.set\_xlabel(xlabel = "Payment method", fontsize = 16)

ax.set\_ylabel(ylabel = " Customer Count", fontsize = 16)

A diagram of a payment method

Description automatically generated

#Payment method distribution in all branches

plt.figure(figsize = (14,6))

plt.style.use('classic')

ax = sns.countplot(x="Payment", hue = "Branch", data = data, palette= "tab20")

ax.set\_title(label = "Payment distribution in all branches", fontsize= 25)

ax.set\_xlabel(xlabel = "Payment method", fontsize = 16)

ax.set\_ylabel(ylabel = "Peple Count", fontsize = 16)

A graph of a payment method

Description automatically generated with medium confidence

#Max sales time

data["Time"]= pd.to\_datetime(data["Time"])

data["Hour"]= (data["Time"]).dt.hour

plt.figure(figsize=(14,6))

plt.style.use('classic')

SalesTime = sns.lineplot(x="Hour", y ="Quantity", data = data).set\_title("product sales per Hour")

A graph with a line

Description automatically generated

#Rating vs sales

plt.figure(figsize=(14,6))

plt.style.use('classic')

rating\_vs\_sales = sns.lineplot(x="Total", y= "Rating", data=data)

A blue lines on a white background

Description automatically generated

#Product sales on the basis of gender

plt.style.use('classic')

plt.figure(figsize = (14,6))

ax= sns.stripplot(y= "Product line", x = "Total", hue = "Gender", data = data)

ax.set\_title(label = "Product sales on the basis of gender")

ax.set\_xlabel(xlabel = " Total sales of products")

ax.set\_ylabel(ylabel = "Product Line")

A screen shot of a graph

Description automatically generated

import matplotlib.pyplot as plt

import seaborn as sns

color = sns.color\_palette()

%matplotlib inline

import plotly.offline as py

py.init\_notebook\_mode(connected=True)

import plotly.graph\_objs as go

import plotly.tools as tls

import plotly.express as px

# Product Scores

fig = px.histogram(df, x="Score")

fig.update\_traces(marker\_line\_width=0.5)

fig.update\_layout(title\_text='Product Score')

fig.show(renderer="colab")

df = df[df['Score'] != 3]

df['sentiment'] = df['Score'].apply(lambda rating : +1 if rating > 3 else -1)

positive = df[df['sentiment'] == 1]

negative = df[df['sentiment'] == -1]

df['sentimental Analysis'] = df['sentiment'].replace({-1 : 'negative'})

df['sentimental Analysis'] = df['sentimental Analysis'].replace({1 : 'positive'})

fig = px.histogram(df, x="sentimental Analysis")

fig.update\_traces(marker\_color="indianred",marker\_line\_color='rgb(8,48,107)',

                  marker\_line\_width=1.5)

fig.update\_layout(title\_text='Product Sentiment')

fig.show(renderer="colab")

A graph of a bar and a rectangular object

Description automatically generated with medium confidence