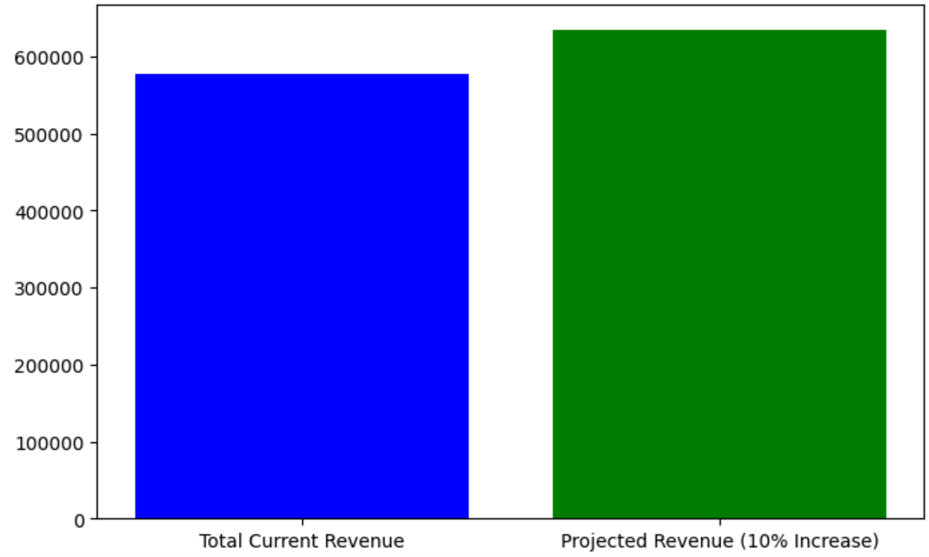
```
import matplotlib.pyplot as plt


import pandas as pd
from datetime import datetime, timedelta


df = pd.read_csv("/content/cleaned_supply_chain_data.csv")


# التحقق من وجود عمود الإيرادات في البيانات
if "Revenue generated" in df.columns:
    # حساب الإيرادات المتوقعة إذا زاد الطلب بنسبة 10%
    df["Predicted_Revenue_10_Percent_Increase"] = df["Revenue generated"] * 1.10


# حساب إجمالي الإيرادات الحالية
total_current_revenue = df["Revenue generated"].sum()


# حساب إجمالي الإيرادات المتوقعة بعد زيادة 10%
total_predicted_revenue = df["Predicted_Revenue_10_Percent_Increase"].sum()


# طباعة النتائج
print("Total current revenue:", round(total_current_revenue, 2))
print("Total projected revenue after 10% demand increase:", round(total_predicted_revenue, 2))
```

```
Total current revenue: 577604.82
Total projected revenue after 10% demand increase: 635365.3
```

```
# Create a bar chart to compare current and projected revenue
labels = ["Total Current Revenue", "Projected Revenue (10% Increase)"]
values = [total_current_revenue, total_predicted_revenue]
plt.figure(figsize=(8, 5))
plt.bar(labels, values, color=["blue", "green"])
```

```
<BarContainer object of 2 artists>
```

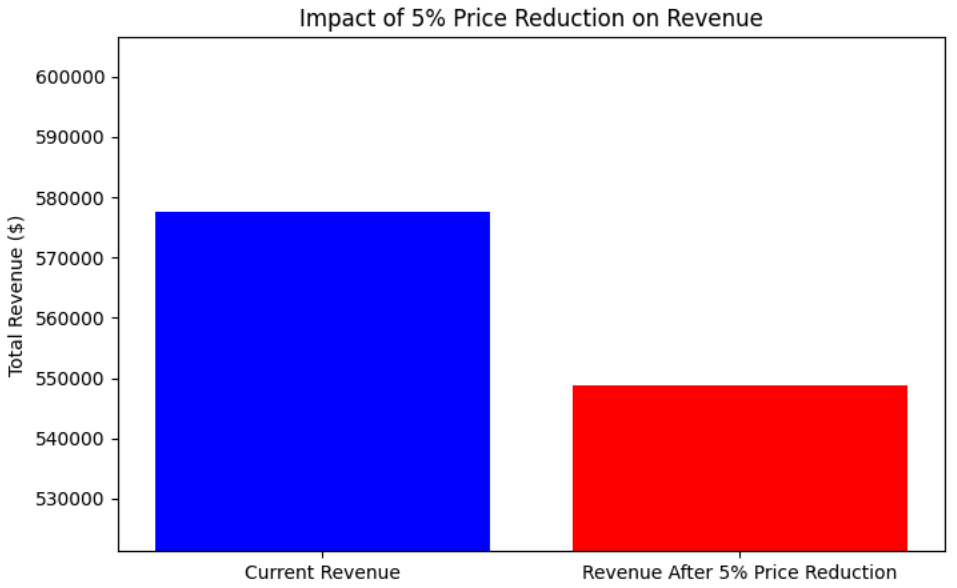


```
# Check if 'Revenue generated' and 'Price' columns exist
if 'Revenue generated' in df.columns and 'Price' in df.columns:
    # Calculate total current revenue
    total_current_revenue = df['Revenue generated'].sum()


# Calculate new revenue after 5% price reduction (assuming demand remains constant)
total_new_revenue = total_current_revenue * 0.95  # 5% reduction


# Calculate revenue difference
revenue_loss = total_current_revenue - total_new_revenue
```

```
# Print results
print(f"Total current revenue: ${total_current_revenue:,.2f}")
print(f"Total revenue after 5% price reduction: ${total_new_revenue:,.2f}")
print(f"Revenue decrease: ${revenue_loss:,.2f} ({(revenue_loss / total_current_revenue) * 100:.2f}%)")
```

```
Total current revenue: $577,604.82
Total revenue after 5% price reduction: $548,724.58
Revenue decrease: $28,880.24 (5.00%)
```

```
 # Plot results
labels = ["Current Revenue", "Revenue After 5% Price Reduction"]
values = [total_current_revenue, total_new_revenue]
plt.figure(figsize=(8, 5))
plt.bar(labels, values, color=['blue', 'red'])
plt.ylabel("Total Revenue ($)")
plt.title("Impact of 5% Price Reduction on Revenue")
plt.ylim(min(values) * 0.95, max(values) * 1.05)  # Adjust y-axis for clarity
plt.show()
```

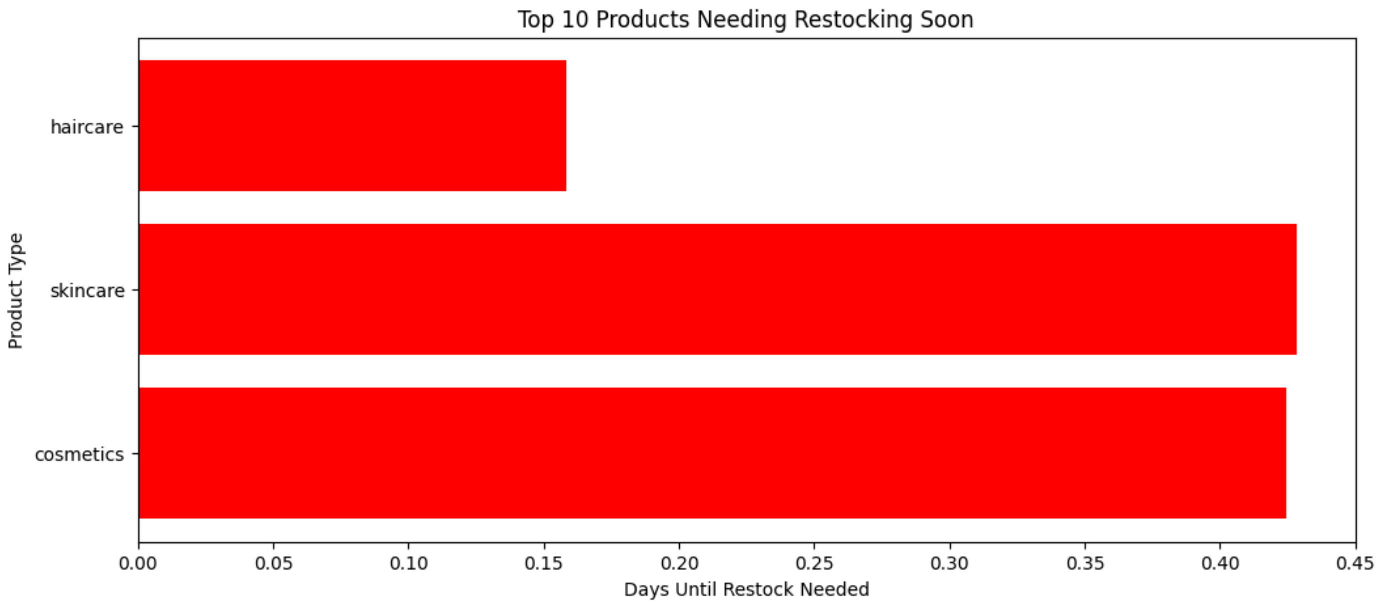


```
# Check if required columns exist
required_columns = ['Product type', 'SKU', 'Stock levels', 'Number of products sold']
if all(col in df.columns for col in required_columns):
    pass # Add your code here


# Assume sales data is from the last 30 days (if no time column is available)
days_observed = 30


# Calculate daily sales rate
df['Daily Sales Rate'] = df['Number of products sold'] / days_observed


# Avoid division by zero errors
df['Days Until Restock'] = df['Stock levels'] / df['Daily Sales Rate']
df.replace([float('inf'), -float('inf')], 9999, inplace=True)  # Mark products with no sales


# Sort products by earliest restock need
df_sorted = df[['Product type', 'SKU', 'Stock levels', 'Daily Sales Rate', 'Days Until Restock']].sort_values(by='Days Until Restock')


# Display top 10 products needing restock soon
print(df_sorted.head(10))
```

```
   Product type    SKU  Stock levels  Daily Sales Rate  Days Until Restock
68     haircare  SKU68             0          5.433333            0.000000
34     skincare  SKU34             1         20.066667            0.049834
47     skincare  SKU47             4         30.333333            0.131868
78     haircare  SKU78             5         31.533333            0.158562
4      skincare   SKU4             5         29.033333            0.172216
33    cosmetics  SKU33             4         20.533333            0.194805
16     skincare  SKU16             2          9.333333            0.214286
58     skincare  SKU58            10         29.866667            0.334821
```

```
    44    cosmetics  SKU44           13         30.633333            0.424374
    9      skincare   SKU9           14         32.666667            0.428571
```

```
# Visualization
top_products = df_sorted.head(10)


plt.figure(figsize=(12, 5))
plt.barh(top_products["Product type"], top_products["Days Until Restock"], color='red')
plt.xlabel("Days Until Restock Needed")
plt.ylabel("Product Type")
plt.title("Top 10 Products Needing Restocking Soon")
plt.gca().invert_yaxis()  # Invert y-axis so shortest time is at the top
plt.show()
```

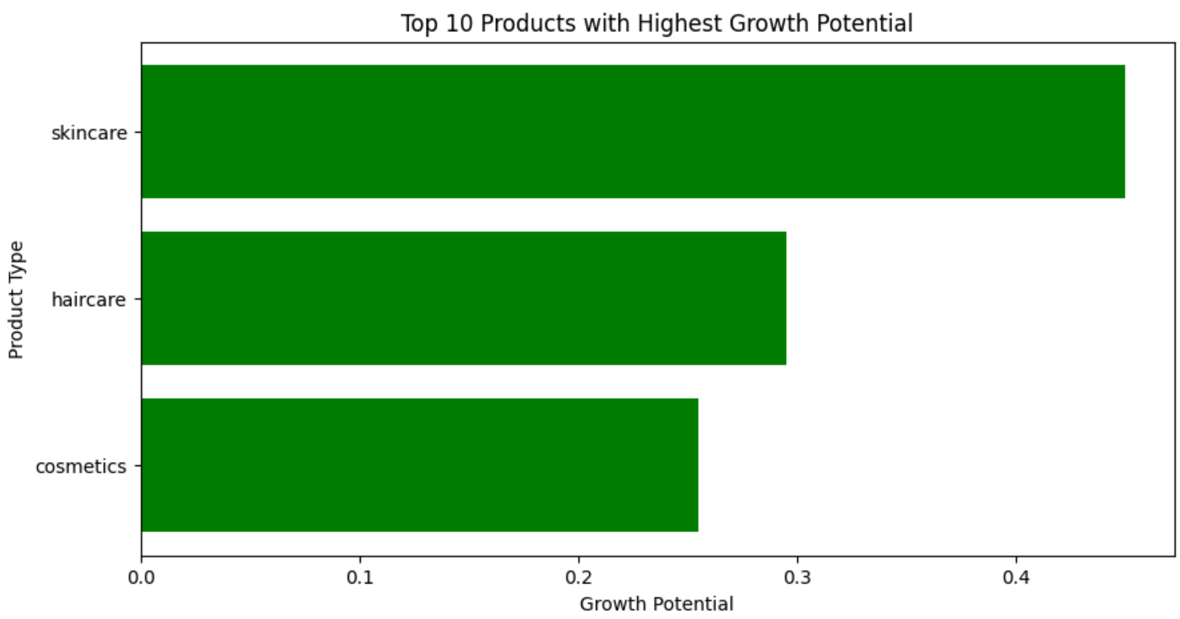


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression


file_path = "/content/cleaned_supply_chain_data.csv"
df = pd.read_csv(file_path)


print(df.columns)
```

```
Index(['Product type', 'SKU', 'Price', 'Availability',
       'Number of products sold', 'Revenue generated', 'Customer demographics',
       'Stock levels', 'Lead times', 'Order quantities', 'Shipping times',
       'Shipping carriers', 'Shipping costs', 'Supplier name', 'Location',
       'Lead time', 'Production volumes', 'Manufacturing lead time',
       'Manufacturing costs', 'Inspection results', 'Defect rates',
       'Transportation modes', 'Routes', 'Costs'],
      dtype='object')
```

```
required_columns = ['Product type', 'Number of products sold']
missing_columns = [col for col in required_columns if col not in df.columns]

if missing_columns:
    print(f"Missing columns in dataset: {missing_columns}")
else:
    # حساب إجمالي المبيعات لكل منتج
    product_sales = df.groupby("Product type")["Number of products sold"].sum().reset_index()


total_sales = product_sales["Number of products sold"].sum()
product_sales["Growth_Potential"] = product_sales["Number of products sold"] / total_sales
```

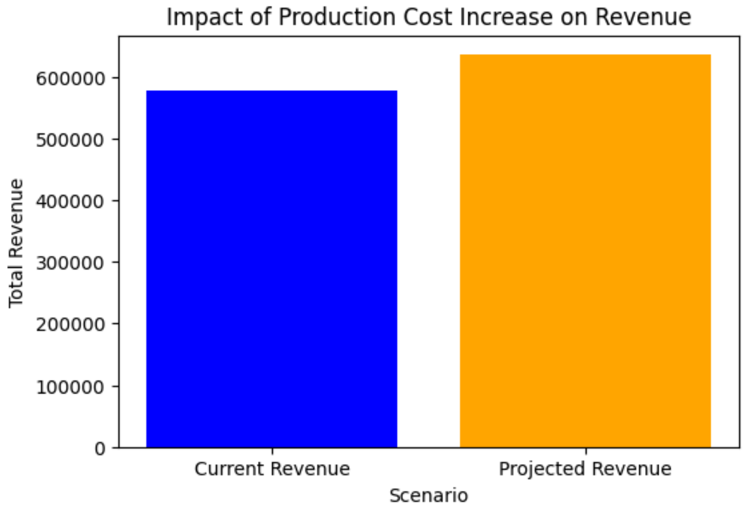
```
top_growing_products = product_sales.sort_values(by="Growth_Potential", ascending=False).head(10)



plt.figure(figsize=(10, 5))
plt.barh(top_growing_products["Product type"], top_growing_products["Growth_Potential"], color="green")
plt.xlabel("Growth Potential")
plt.ylabel("Product Type")
plt.title("Top 10 Products with Highest Growth Potential")
plt.gca().invert_yaxis()
plt.show()
```



```
print("Top 10 Products with Highest Growth Potential:")
print(top_growing_products[["Product type", "Growth_Potential"]])
```

```
Top 10 Products with Highest Growth Potential:
  Product type  Growth_Potential
2     skincare          0.449706
1     haircare          0.295256
0    cosmetics          0.255038
```

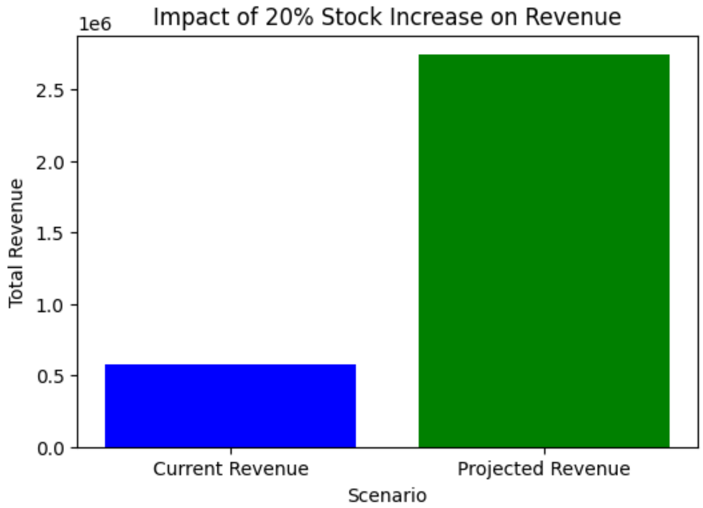
```
required_columns = ['Product type', 'Price', 'Manufacturing costs', 'Revenue generated']
missing_columns = [col for col in required_columns if col not in df.columns]

if missing_columns:
    print(f"Missing columns: {missing_columns}")
else:
    # حساب الهامش الربحي الحالي لكل منتج
    df["Profit Margin"] = (df["Price"] - df["Manufacturing costs"]) / df["Price"]

    # زيادة تكاليف الإنتاج بنسبة 10%
    df["New Manufacturing Costs"] = df["Manufacturing costs"] * 1.10

    # حساب الأسعار الجديدة مع الحفاظ على نفس نسبة الهامش الربحي
    df["New Price"] = df["New Manufacturing Costs"] / (1 - df["Profit Margin"])

    # حساب الأرباح الجديدة بعد تعديل الأسعار
    df["New Revenue"] = df["New Price"] * (df["Revenue generated"] / df["Price"])



  # مقارنة الأرباح قبل وبعد الزيادة
total_revenue_before = df["Revenue generated"].sum()
total_revenue_after = df["New Revenue"].sum()


 # رسم بياني لمقارنة الأرباح قبل وبعد
plt.figure(figsize=(6, 4))
plt.bar(["Current Revenue", "Projected Revenue"], [total_revenue_before, total_revenue_after], color=["blue", "orange"])
plt.xlabel("Scenario")
plt.ylabel("Total Revenue")
plt.title("Impact of Production Cost Increase on Revenue")
plt.show()
```



```
# طباعة النتائج
print(f"Total Current Revenue: {total_revenue_before:.2f}")
print(f"Total Projected Revenue after 10% Cost Increase: {total_revenue_after:.2f}")
```

```
Total Current Revenue: 577604.82
Total Projected Revenue after 10% Cost Increase: 635365.30
```

```
# التحقق من الأعمدة المطلوبة
required_columns = ['Product type', 'Stock levels', 'Number of products sold', 'Price', 'Revenue generated']
missing_columns = [col for col in required_columns if col not in df.columns]

if missing_columns:
    print(f"Missing columns: {missing_columns}")
else:
    # زيادة المخزون بنسبة 20%
    df["New Stock Levels"] = df["Stock levels"] * 1.20

    # افتراض أن زيادة المخزون تؤدي إلى زيادة المبيعات بنفس النسبة
    df["New Sales"] = df["Number of products sold"] * 1.20

    # حساب الإيرادات الجديدة بعد زيادة المبيعات
    df["New Revenue"] = df["New Sales"] * df["Price"]

    # حساب إجمالي الإيرادات قبل وبعد الزيادة
    total_revenue_before = df["Revenue generated"].sum()
    total_revenue_after = df["New Revenue"].sum()



plt.figure(figsize=(6, 4))
plt.bar(["Current Revenue", "Projected Revenue"], [total_revenue_before, total_revenue_after], color=["blue", "green"])
plt.xlabel("Scenario")
plt.ylabel("Total Revenue")
plt.title("Impact of 20% Stock Increase on Revenue")
plt.show()
```



Start coding or generate with AI.