```
pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.11/dist-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.11/dist-packages (from pyspark) (0.10.9.7)
```

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, count, avg, desc
```

```
spark = SparkSession.builder \
    .appName("Big Data Analysis Internship Task") \
    .getOrCreate()
```

```
df = spark.read.csv("/content/Walmart_Sales.csv", header=True, inferSchema=True)
df.printSchema()
df.show(5)
```

```
root
 |-- Store: integer (nullable = true)
 |-- Date: string (nullable = true)
 |-- Weekly_Sales: double (nullable = true)
 |-- Holiday_Flag: integer (nullable = true)
 |-- Temperature: double (nullable = true)
 |-- Fuel_Price: double (nullable = true)
 |-- CPI: double (nullable = true)
 |-- Unemployment: double (nullable = true)

+-----+----------+------------+------------+-----------+----------+-----------+------------+
|Store|      Date|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemployment|
+-----+----------+------------+------------+-----------+----------+-----------+------------+
|    1|05-02-2010|   1643690.9|           0|      42.31|     2.572|211.0963582|       8.106|
|    1|12-02-2010|  1641957.44|           1|      38.51|     2.548|211.2421698|       8.106|
|    1|19-02-2010|  1611968.17|           0|      39.93|     2.514|211.2891429|       8.106|
|    1|26-02-2010|  1409727.59|           0|      46.63|     2.561|211.3196429|       8.106|
|    1|05-03-2010|  1554806.68|           0|       46.5|     2.625|211.3501429|       8.106|
+-----+----------+------------+------------+-----------+----------+-----------+------------+
only showing top 5 rows
```

```
df.describe().show()  # summary statistics
df.columns            # list of columns
df.count()            # number of rows
df.dtypes             # data types
```

```
+-------+------------------+----------+------------------+-------------------+------------------+-------------------+---
|summary|             Store|      Date|      Weekly_Sales|       Holiday_Flag|       Temperature|         Fuel_Price|
+-------+------------------+----------+------------------+-------------------+------------------+-------------------+---
|  count|              6435|      6435|              6435|               6435|              6435|               6435|
|   mean|              23.0|      NULL|1046964.8775617732|0.06993006993006994| 60.66378243978229|  3.358606837606832|171
| stddev|12.988182381175454|      NULL| 564366.6220536977| 0.2550489443698279|18.444932875811585|0.45901970719285223|39.
|    min|                 1|01-04-2011|         209986.25|                  0|             -2.06|              2.472|
|    max|                45|31-12-2010|        3818686.45|                  1|            100.14|              4.468|
+-------+------------------+----------+------------------+-------------------+------------------+-------------------+---

[('Store', 'int'),
 ('Date', 'string'),
 ('Weekly_Sales', 'double'),
 ('Holiday_Flag', 'int'),
 ('Temperature', 'double'),
 ('Fuel_Price', 'double'),
 ('CPI', 'double'),
 ('Unemployment', 'double')]
```

```
from pyspark.sql.functions import isnan, when, count

df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()
```

```
+-----+----+------------+------------+-----------+----------+---+------------+
|Store|Date|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|CPI|Unemployment|
+-----+----+------------+------------+-----------+----------+---+------------+
|    0|   0|           0|           0|          0|         0|  0|           0|
+-----+----+------------+------------+-----------+----------+---+------------+
```

```
df.groupBy("Temperature").agg(
    count("*").alias("Total_Transactions"),
    avg("Weekly_Sales").alias("Average_Weekly_Sales")
).orderBy(desc("Total_Transactions")).show()
```

```
+-----------+------------------+--------------------+
|Temperature|Total_Transactions|Average_Weekly_Sales|
+-----------+------------------+--------------------+
|      50.43|               11|   937506.0818181819|
|      67.87|               10|   821978.0349999999|
|      76.67|                9|   1095250.281111111|
|      72.62|                9|  1209599.3055555553|
|      70.28|                9|    964847.051111111|
|      76.03|                9|  1157620.9699999997|
|      64.05|                8|       1172307.38625|
|      50.56|                8|  1169034.2912500002|
|      64.21|                8|   905388.9524999999|
|      62.62|                7|  1370573.1685714286|
|      49.96|                7|   988308.2628571427|
|      50.81|                7|  1295151.0457142857|
|      70.19|                7|   1003058.072857143|
|      78.47|                7|   1101103.775714286|
|      70.87|                7|   906468.0928571429|
|      44.42|                7|   956922.4485714287|
|      40.65|                7|   938682.9657142855|
|      52.27|                7|  1022438.3071428572|
|      58.97|                7|   902153.7571428573|
|      77.49|                6|  1189246.4716666667|
+-----------+-----------------+--------------------+
only showing top 20 rows
```

```
df.filter(col("Weekly_Sales") > 10000).show()
```

```
+-----+----------+------------+------------+-----------+----------+-----------+------------+
|Store|      Date|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemployment|
+-----+----------+------------+------------+-----------+----------+-----------+------------+
|    1|05-02-2010|   1643690.9|           0|      42.31|     2.572|211.0963582|       8.106|
|    1|12-02-2010|  1641957.44|           1|      38.51|     2.548|211.2421698|       8.106|
|    1|19-02-2010|  1611968.17|           0|      39.93|     2.514|211.2891429|       8.106|
|    1|26-02-2010|  1409727.59|           0|      46.63|     2.561|211.3196429|       8.106|
|    1|05-03-2010|  1554806.68|           0|       46.5|     2.625|211.3501429|       8.106|
|    1|12-03-2010|  1439541.59|           0|      57.79|     2.667|211.3806429|       8.106|
|    1|19-03-2010|  1472515.79|           0|      54.58|      2.72| 211.215635|       8.106|
|    1|26-03-2010|  1404429.92|           0|      51.45|     2.732|211.0180424|       8.106|
|    1|02-04-2010|  1594968.28|           0|      62.27|     2.719|210.8204499|       7.808|
|    1|09-04-2010|  1545418.53|           0|      65.86|      2.77|210.6228574|       7.808|
|    1|16-04-2010|  1466058.28|           0|      66.32|     2.808|   210.4887|       7.808|
|    1|23-04-2010|  1391256.12|           0|      64.84|     2.795|210.4391228|       7.808|
|    1|30-04-2010|  1425100.71|           0|      67.41|      2.78|210.3895456|       7.808|
|    1|07-05-2010|  1603955.12|           0|      72.55|     2.835|210.3399684|       7.808|
|    1|14-05-2010|   1494251.5|           0|      74.78|     2.854|210.3374261|       7.808|
|    1|21-05-2010|  1399662.07|           0|      76.44|     2.826|210.6170934|       7.808|
|    1|28-05-2010|  1432069.95|           0|      80.44|     2.759|210.8967606|       7.808|
|    1|04-06-2010|  1615524.71|           0|      80.69|     2.705|211.1764278|       7.808|
|    1|11-06-2010|  1542561.09|           0|      80.43|     2.668|211.4560951|       7.808|
|    1|18-06-2010|  1503284.06|           0|      84.11|     2.637|211.4537719|       7.808|
+-----+----------+------------+------------+-----------+----------+-----------+------------+
only showing top 20 rows
```

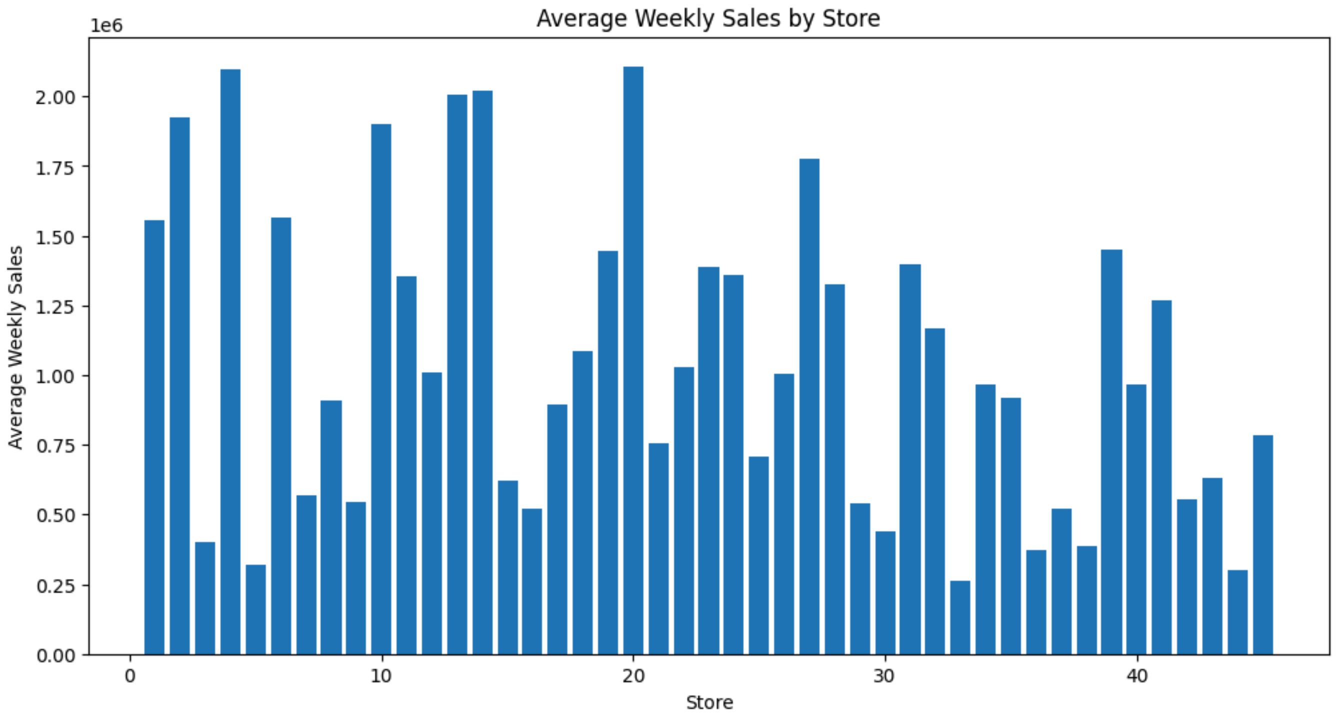
```
df.groupBy("Fuel_Price").agg(count("*").alias("Sales_Count")) \
  .orderBy(desc("Sales_Count")).show(10)
```

```
+----------+-----------+
|Fuel_Price|Sales_Count|
+----------+-----------+
|     3.638|         39|
|      3.63|         34|
|     2.771|         29|
|     3.891|         29|
|     3.594|         28|
|     3.524|         28|
|      2.72|         28|
|     3.666|         27|
|     3.523|         27|
|     3.129|         25|
+----------+-----------+
only showing top 10 rows
```

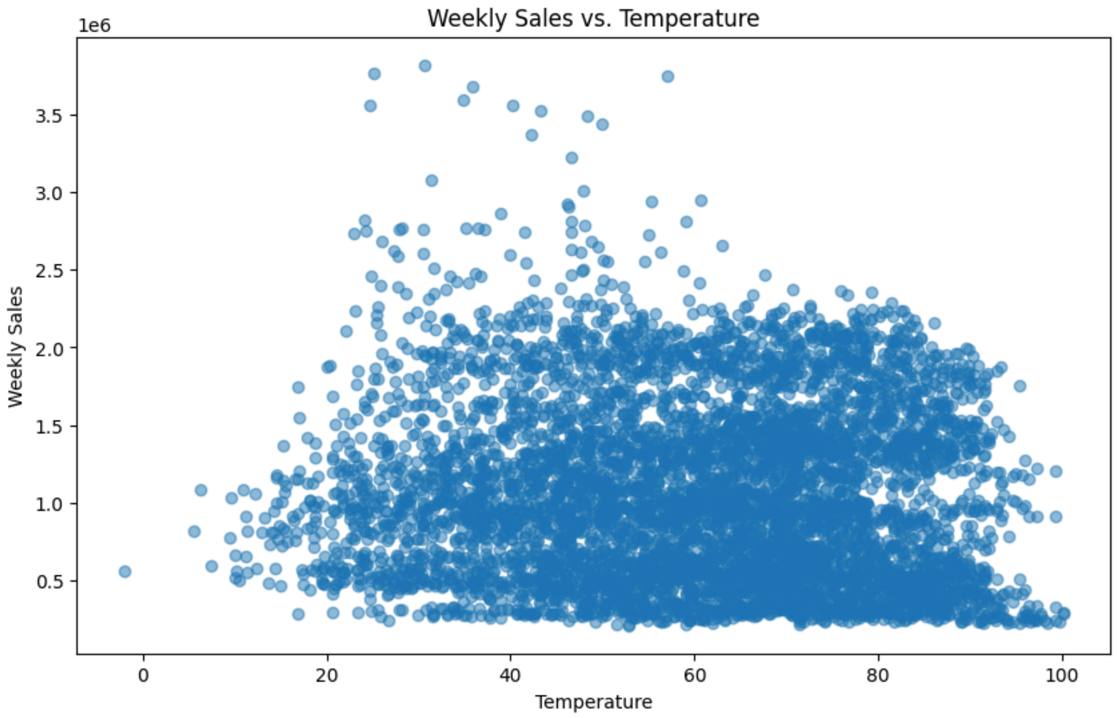
```
# draw a bar chart
import matplotlib.pyplot as plt
import pandas as pd
from pyspark.sql.functions import pandas_udf
from pyspark.sql.types import StringType


# Convert Spark DataFrame to Pandas DataFrame for plotting
sales_by_store_df = df.groupBy("Store").agg({"Weekly_Sales": "avg"}).orderBy("Store").toPandas()

# Create a bar chart of average weekly sales by store
plt.figure(figsize=(12, 6))
plt.bar(sales_by_store_df["Store"], sales_by_store_df["avg(Weekly_Sales)"])
plt.xlabel("Store")
plt.ylabel("Average Weekly Sales")
plt.title("Average Weekly Sales by Store")
plt.show()
```



```
# Create a scatter plot of Weekly Sales vs. Temperature
plt.figure(figsize=(10, 6))
plt.scatter(df.select("Temperature").toPandas(), df.select("Weekly_Sales").toPandas(), alpha=0.5)
plt.xlabel("Temperature")
plt.ylabel("Weekly Sales")
plt.title("Weekly Sales vs. Temperature")
plt.show()
```

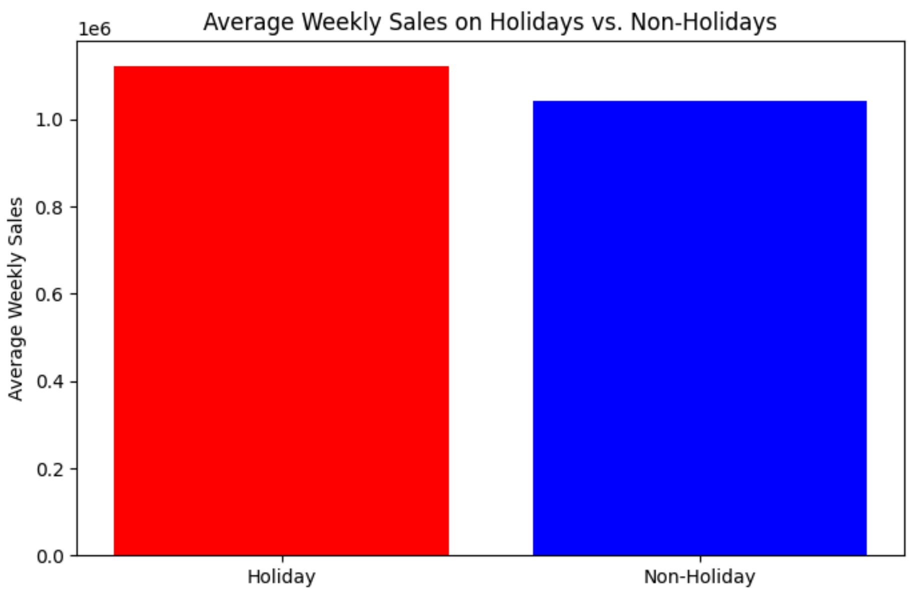


```
# Calculate average weekly sales for holidays and non-holidays
holiday_sales = df.filter(col("Holiday_Flag") == 1).agg(avg("Weekly_Sales")).collect()[0][0]
non_holiday_sales = df.filter(col("Holiday_Flag") == 0).agg(avg("Weekly_Sales")).collect()[0][0]

# Create a bar chart
labels = ['Holiday', 'Non-Holiday']
averages = [holiday_sales, non_holiday_sales]

plt.figure(figsize=(8, 5))
```

```
    plt.bar(labels, averages, color=['red', 'blue'])
    plt.ylabel('Average Weekly Sales')
    plt.title('Average Weekly Sales on Holidays vs. Non-Holidays')
    plt.show()
```


Average Weekly Sales on Holidays vs. Non-Holidays
1e6
Average Weekly Sales
Holiday
Non-Holiday