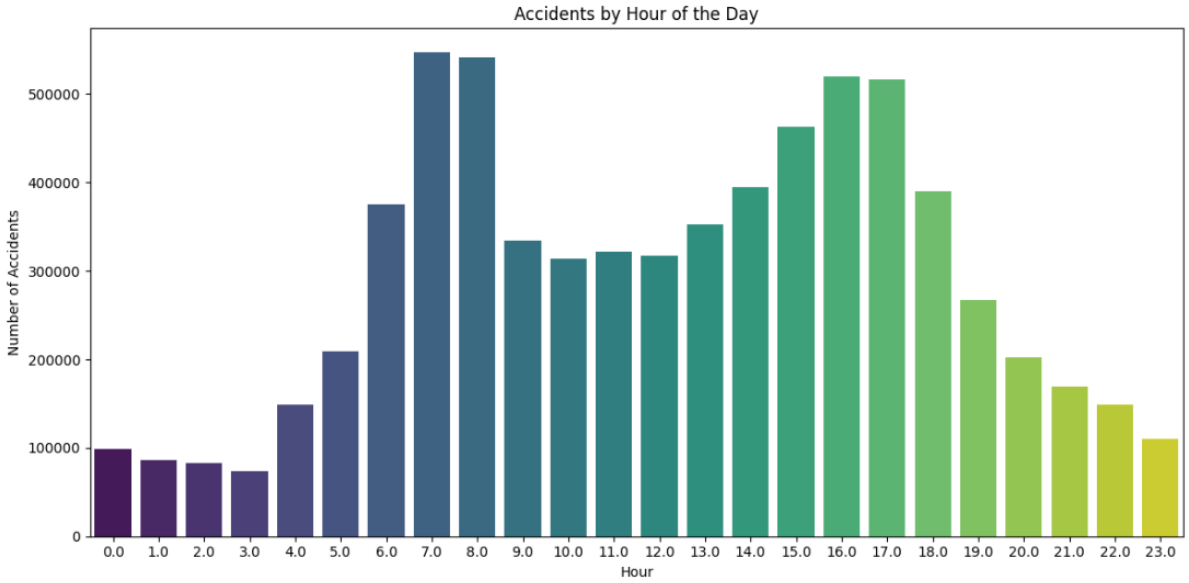
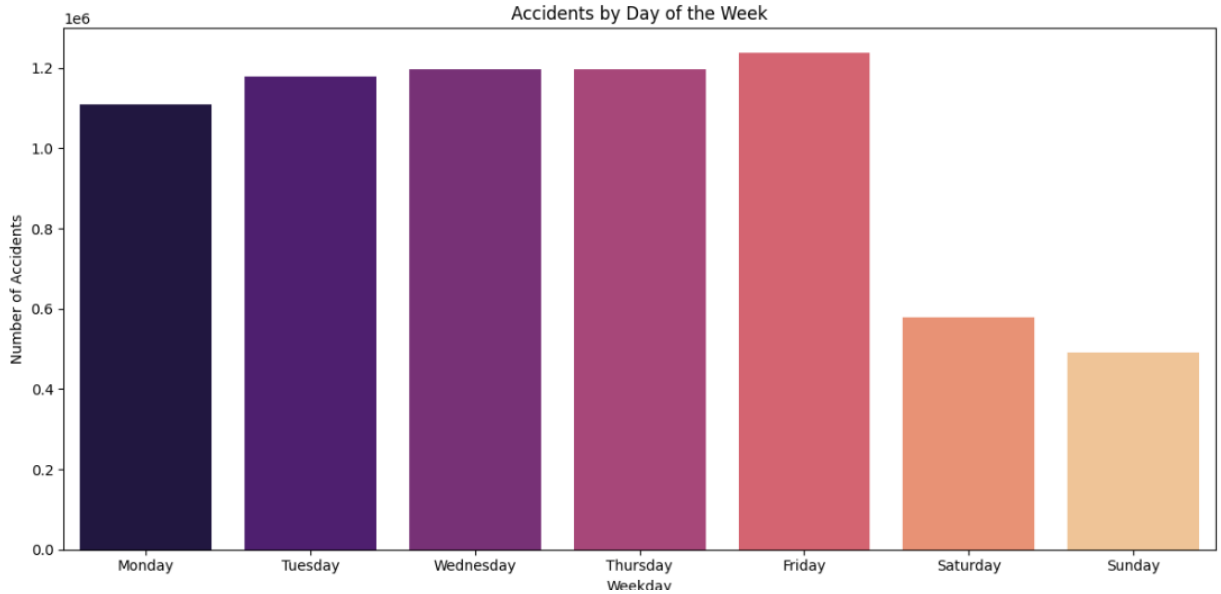
SKILLCRAFT TECHNOLOGY INTERNSHIP TASK-4

Analyze traffic accident data to identify patterns related to road conditions weather and time of day. Visualize accident hotspots and contributing factors

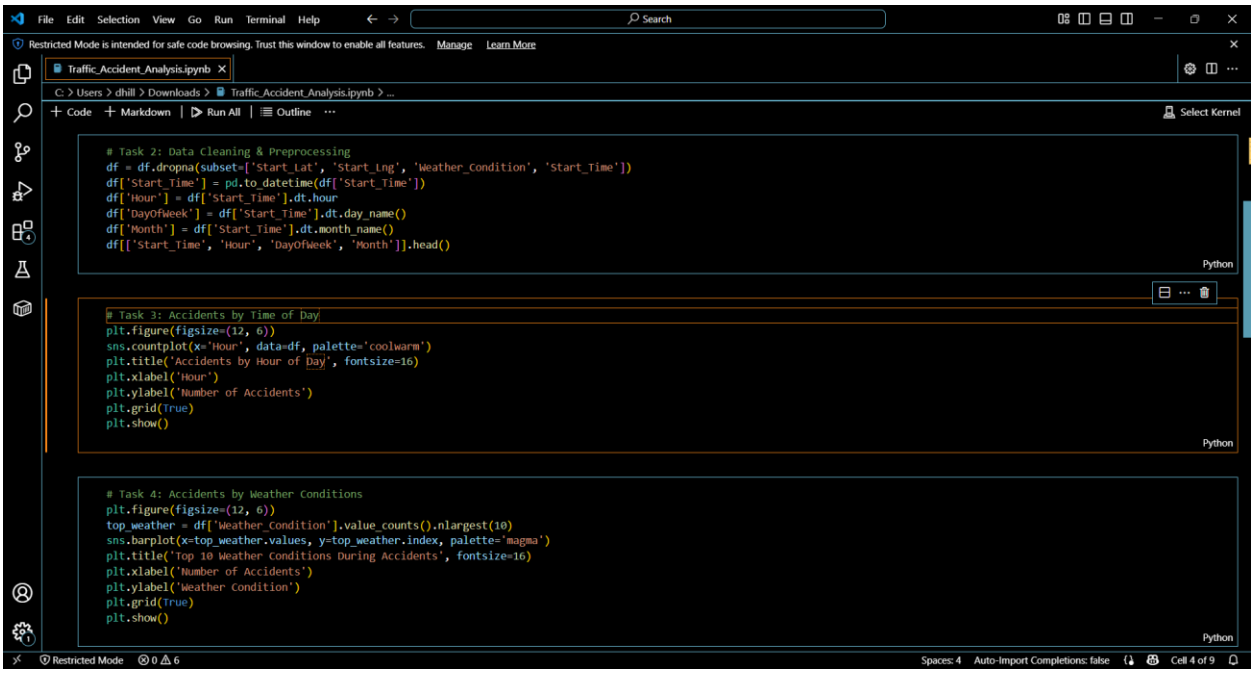
```
# STEP 1: Import required libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import os

# STEP 2: Locate the file inside the Kaggle input directory
folder_path = '/kaggle/input/us-accidents'
files = os.listdir(folder_path)
print("📁 Files found:", files)

# STEP 3: Load the dataset using the correct file name
csv_file = [f for f in files if f.endswith('.csv')][0]  # take the first CSV file
file_path = os.path.join(folder_path, csv_file)
print("✅ Loading file:", file_path)

df = pd.read_csv(file_path, low_memory=False)

# STEP 4: Parse datetime and extract hour and weekday
df['Start_Time'] = pd.to_datetime(df['Start_Time'], errors='coerce')
df['Hour'] = df['Start_Time'].dt.hour
df['Weekday'] = df['Start_Time'].dt.day_name()
```

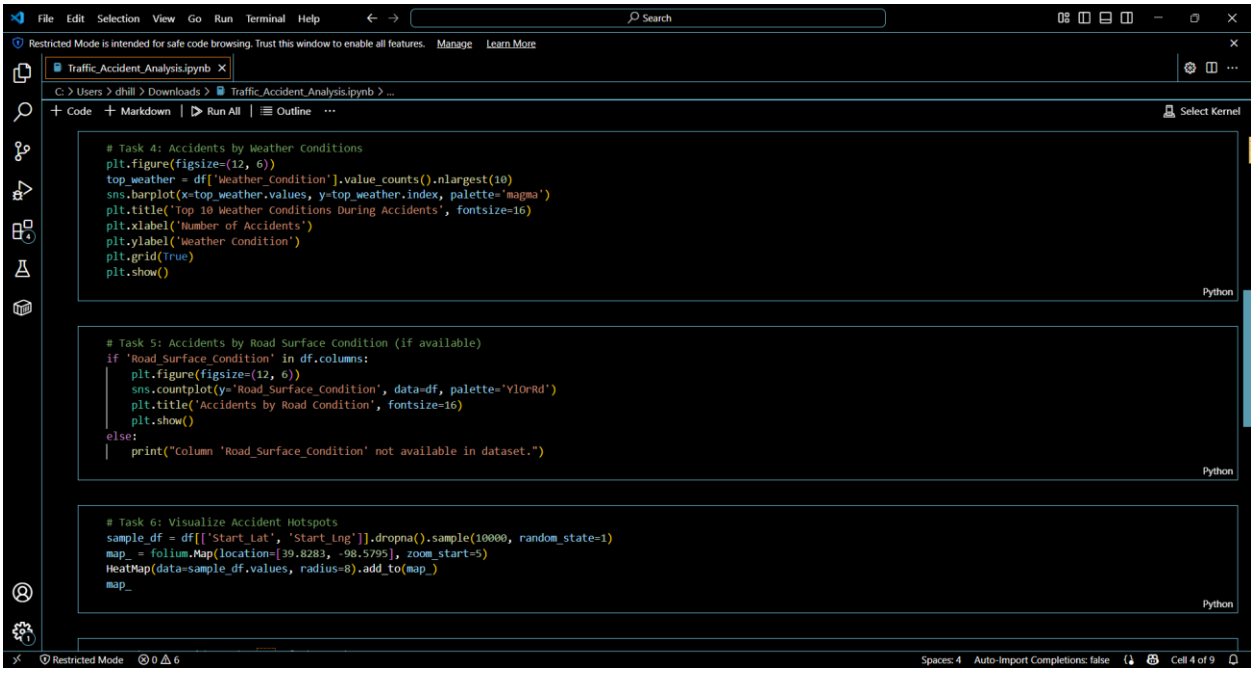
```
# STEP 5: Plot accidents by hour
plt.figure(figsize=(12,6))
sns.countplot(x='Hour', data=df, palette='viridis')
plt.title("Accidents by Hour of the Day")
plt.xlabel("Hour")
plt.ylabel("Number of Accidents")
plt.tight_layout()
plt.show()

# STEP 6: Plot accidents by weekday
plt.figure(figsize=(12,6))
weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
sns.countplot(x='Weekday', data=df, order=weekday_order, palette='magma')
plt.title("Accidents by Day of the Week")
plt.xlabel("Weekday")
plt.ylabel("Number of Accidents")
plt.tight_layout()
plt.show()
```


Accidents by Hour of the Day
Number of Accidents
0
100000
200000
300000
400000
500000
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
Hour

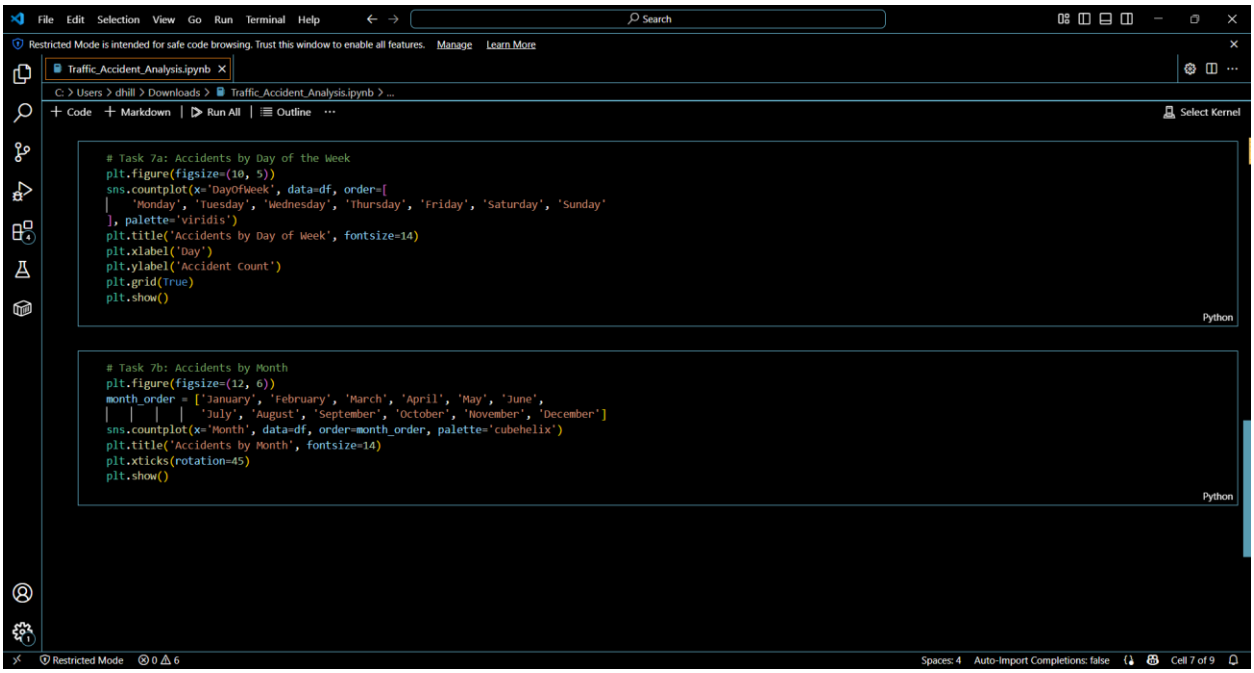

Accidents by Day of the Week
1e6
Number of Accidents
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday


File Edit Selection View Go Run Terminal Help
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Traffic_Accident_Analysis.ipynb
C: > Users > dhill > Downloads > Traffic_Accident_Analysis.ipynb > ...
+ Code + Markdown | Run All | Outline ···
Select Kernel
# Task 1: Import Libraries and Load Data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium.plugins import HeatMap
import warnings
warnings.filterwarnings("ignore")
df = pd.read_csv('US_Accidents_Dec21_updated.csv') # Ensure this file is in your directory
print(df.shape)
df.head()
Python
# Task 2: Data Cleaning & Preprocessing
df = df.dropna(subset=['Start_Lat', 'Start_Lng', 'Weather_Condition', 'Start_Time'])
df['Start_Time'] = pd.to_datetime(df['Start_Time'])
df['Hour'] = df['Start_Time'].dt.hour
df['DayOfWeek'] = df['Start_Time'].dt.day_name()
df['Month'] = df['Start_Time'].dt.month_name()
df[['Start_Time', 'Hour', 'DayOfWeek', 'Month']].head()
Python
# Task 3: Accidents by Time of Day
plt.figure(figsize=(12, 6))
sns.countplot(x='Hour', data=df, palette='coolwarm')
plt.title('Accidents by Hour of Day', fontsize=16)
plt.xlabel('Hour')
Restricted Mode 0 6
Cell 1 of 9


File Edit Selection View Go Run Terminal Help
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Traffic_Accident_Analysis.ipynb
C: > Users > dhill > Downloads > Traffic_Accident_Analysis.ipynb > ...
+ Code + Markdown | Run All | Outline ···
Select Kernel
# Task 2: Data Cleaning & Preprocessing
df = df.dropna(subset=['Start_Lat', 'Start_Lng', 'Weather_Condition', 'Start_Time'])
df['Start_Time'] = pd.to_datetime(df['Start_Time'])
df['Hour'] = df['Start_Time'].dt.hour
df['DayOfWeek'] = df['Start_Time'].dt.day_name()
df['Month'] = df['Start_Time'].dt.month_name()
df[['Start_Time', 'Hour', 'DayOfWeek', 'Month']].head()
Python
# Task 3: Accidents by Time of Day
plt.figure(figsize=(12, 6))
sns.countplot(x='Hour', data=df, palette='coolwarm')
plt.title('Accidents by Hour of Day', fontsize=16)
plt.xlabel('Hour')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
Python
# Task 4: Accidents by Weather Conditions
plt.figure(figsize=(12, 6))
top_weather = df['Weather_Condition'].value_counts().nlargest(10)
sns.barplot(x=top_weather.values, y=top_weather.index, palette='magma')
plt.title('Top 10 Weather Conditions During Accidents', fontsize=16)
plt.xlabel('Number of Accidents')
plt.ylabel('Weather Condition')
plt.grid(True)
plt.show()
Python
Restricted Mode 0 6
Spaces: 4 Auto-Import Completions: false
Cell 4 of 9


File Edit Selection View Go Run Terminal Help
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Traffic_Accident_Analysis.ipynb
C: > Users > dhill > Downloads > Traffic_Accident_Analysis.ipynb > ...
+ Code + Markdown | Run All | Outline ···
Select Kernel
# Task 4: Accidents by Weather Conditions
plt.figure(figsize=(12, 6))
top_weather = df['Weather_Condition'].value_counts().nlargest(10)
sns.barplot(x=top_weather.values, y=top_weather.index, palette='magma')
plt.title('Top 10 Weather Conditions During Accidents', fontsize=16)
plt.xlabel('Number of Accidents')
plt.ylabel('Weather Condition')
plt.grid(True)
plt.show()
Python
# Task 5: Accidents by Road Surface Condition (if available)
if 'Road_Surface_Condition' in df.columns:
plt.figure(figsize=(12, 6))
sns.countplot(y='Road_Surface_Condition', data=df, palette='YlOrRd')
plt.title('Accidents by Road Condition', fontsize=16)
plt.show()
else:
print("Column 'Road_Surface_Condition' not available in dataset.")
Python
# Task 6: Visualize Accident Hotspots
sample_df = df[['Start_Lat', 'Start_Lng']].dropna().sample(10000, random_state=1)
map_ = folium.Map(location=[39.8283, -98.5795], zoom_start=5)
HeatMap(data=sample_df.values, radius=8).add_to(map_)
map_
Python
Restricted Mode 0 6
Spaces: 4 Auto-Import Completions: false Cell 4 of 9


File Edit Selection View Go Run Terminal Help
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Traffic_Accident_Analysis.ipynb
C: > Users > dhill > Downloads > Traffic_Accident_Analysis.ipynb > ...
+ Code + Markdown | Run All | Outline ···
Select Kernel
# Task 6: Visualize Accident Hotspots
sample_df = df[['Start_Lat', 'Start_Lng']].dropna().sample(10000, random_state=1)
map_ = folium.Map(location=[39.8283, -98.5795], zoom_start=5)
HeatMap(data=sample_df.values, radius=8).add_to(map_)
map_
Python
# Task 7a: Accidents by Day of the Week
plt.figure(figsize=(10, 5))
sns.countplot(x='DayOfWeek', data=df, order=[
'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'
], palette='viridis')
plt.title('Accidents by Day of Week', fontsize=14)
plt.xlabel('Day')
plt.ylabel('Accident Count')
plt.grid(True)
plt.show()
Python
# Task 7b: Accidents by Month
plt.figure(figsize=(12, 6))
month_order = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December']
sns.countplot(x='Month', data=df, order=month_order, palette='cubehelix')
plt.title('Accidents by Month', fontsize=14)
plt.xticks(rotation=45)
plt.show()
Python
Restricted Mode 0 6
Spaces: 4 Auto-Import Completions: false Cell 7 of 9


File Edit Selection View Go Run Terminal Help
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Traffic_Accident_Analysis.ipynb
C: > Users > dhill > Downloads > Traffic_Accident_Analysis.ipynb > ...
Code Markdown Run All Outline
Select Kernel
# Task 7a: Accidents by Day of the Week
plt.figure(figsize=(10, 5))
sns.countplot(x='DayOfWeek', data=df, order=[
'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'
], palette='viridis')
plt.title('Accidents by Day of Week', fontsize=14)
plt.xlabel('Day')
plt.ylabel('Accident Count')
plt.grid(True)
plt.show()
Python
# Task 7b: Accidents by Month
plt.figure(figsize=(12, 6))
month_order = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December']
sns.countplot(x='Month', data=df, order=month_order, palette='cubehelix')
plt.title('Accidents by Month', fontsize=14)
plt.xticks(rotation=45)
plt.show()
Python
Restricted Mode 0 6
Spaces: 4 Auto-Import Completions: false Cell 7 of 9