In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df=pd.read_csv(r"C:\Data Science Power bi\traffic_accidents.csv")
df
```

Out[3]:

	Weather	Road_Type	Time_of_Day	Traffic_Density	Speed_Limit	Number_of_Vehicles	I
0	Rainy	City Road	Morning	1.0	100.0	5.0	
1	Clear	Rural Road	Night	NaN	120.0	3.0	
2	Rainy	Highway	Evening	1.0	60.0	4.0	
3	Clear	City Road	Afternoon	2.0	60.0	3.0	
4	Rainy	Highway	Morning	1.0	195.0	11.0	
...	...	...	...	...	...	...	
835	Clear	Highway	Night	2.0	30.0	4.0	
836	Rainy	Rural Road	Evening	2.0	60.0	4.0	
837	Foggy	Highway	Evening	NaN	30.0	4.0	
838	Foggy	Highway	Afternoon	2.0	60.0	3.0	

In [127]:
```
df.head()
```

Out[127]:

	Weather	Road_Type	Time_of_Day	Traffic_Density	Speed_Limit	Number_of_Vehicles	Driver_
0	Rainy	City Road	Morning	1.0	100.0	5.0	
1	Clear	Rural Road	Night	NaN	120.0	3.0	
2	Rainy	Highway	Evening	1.0	60.0	4.0	
3	Clear	City Road	Afternoon	2.0	60.0	3.0	
4	Rainy	Highway	Morning	1.0	195.0	11.0	

In [128]:
```
df.tail()
```

Out[128]:

	Weather	Road_Type	Time_of_Day	Traffic_Density	Speed_Limit	Number_of_Vehicles	Drive
835	Clear	Highway	Night	2.0	30.0	4.0	
836	Rainy	Rural Road	Evening	2.0	60.0	4.0	
837	Foggy	Highway	Evening	NaN	30.0	4.0	
838	Foggy	Highway	Afternoon	2.0	60.0	3.0	
839	Clear	Highway	Afternoon	1.0	60.0	4.0	

In [3]:
```
df.isnull().sum()
```

Out[3]:
```
Weather                 42
Road_Type               42
Time_of_Day             42
Traffic_Density         42
Speed_Limit             42
Number_of_Vehicles      42
Driver_Alcohol          42
Accident_Severity       42
Road_Condition          42
Vehicle_Type            42
Driver_Age              42
Driver_Experience       42
Road_Light_Condition    42
Accident                42
dtype: int64
```

To fill the missing value in given data

In [33]:
```
we=df["Weather"].mode()[0]
we
```

Out[33]:
```
'Clear'
```

In [43]:
```
df["Weather"].fillna(we,inplace=True)
```

In [44]:
```
rt=df["Road_Type"].mode()[0]
rt
```

Out[44]:
```
'Highway'
```

In [45]:
```
df["Road_Type"].fillna(rt,inplace=True)
```

In [46]:
```
tof=df["Time_of_Day"].mode()[0]
tof
```

Out[46]: 'Afternoon'

In [47]:
```
df["Time_of_Day"].fillna(tof,inplace=True)
```

In [48]:
```
accs=df["Accident_Severity"].mode()[0]
accs
```

Out[48]: 'Low'

In [49]:
```
df["Accident_Severity"].fillna(accs,inplace=True)
```

In [51]:
```
rc=df["Road_Condition"].mode()[0]
rc
```

Out[51]: 'Dry'

In [52]:
```
df["Road_Condition"].fillna(rc,inplace=True)
```

In [54]:
```
vt=df["Vehicle_Type"].mode()[0]
vt
```

Out[54]: 'Car'

In [55]:
```
df["Vehicle_Type"].fillna(vt,inplace=True)
```

In [57]:
```
rlc=df["Road_Light_Condition"].mode()[0]
rlc
```

Out[57]: 'Artificial Light'

In [58]:
```
df["Road_Light_Condition"].fillna(rlc,inplace=True)
```

In [59]:
```
td=round(df["Traffic_Density"].mean(),)
td
```

Out[59]: 1

In [60]:
```
df["Traffic_Density"].fillna(td,inplace=True)
```

In [61]:
```
sl=round(df["Speed_Limit"].mean(),)
sl
```

Out[61]: 71

In [62]:
```
df["Speed_Limit"].fillna(sl,inplace=True)
```

In [63]:
```
nov=round(df["Number_of_Vehicles"].median(),)
nov
```

Out[63]: 3

In [64]:
```
df["Number_of_Vehicles"].fillna(nov,inplace=True)
```

In [65]:
```
da=round(df["Driver_Alcohol"].mean(),)
da
```

Out[65]: 0

In [66]:
```
df["Driver_Alcohol"].fillna(da,inplace=True)
```

In [67]:
```
das=round(df["Driver_Age"].mean(),)
das
```

Out[67]: 43

In [68]:
```
df["Driver_Age"].fillna(das,inplace=True)
```

In [69]:
```
de=round(df["Driver_Experience"].mean(),)
de
```

Out[69]: 39

In [70]:
```
df["Driver_Experience"].fillna(de,inplace=True)
```

In [71]:
```
acc=round(df["Accident"].mean(),)
acc
```

Out[71]: 0

In [73]:
```
df["Accident"].fillna(acc,inplace=True)
```

In [75]:
```
df.isnull().sum()
```

Out[75]:
```
Weather                 0
Road_Type               0
Time_of_Day             0
Traffic_Density         0
Speed_Limit             0
Number_of_Vehicles      0
Driver_Alcohol          0
Accident_Severity       0
Road_Condition          0
Vehicle_Type            0
Driver_Age              0
Driver_Experience       0
Road_Light_Condition    0
Accident                0
dtype: int64
```

To Change the data types of fiven data

In [76]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 840 entries, 0 to 839
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Weather               840 non-null    object 
 1   Road_Type             840 non-null    object 
 2   Time_of_Day           840 non-null    object 
 3   Traffic_Density       840 non-null    float64
 4   Speed_Limit           840 non-null    float64
 5   Number_of_Vehicles    840 non-null    float64
 6   Driver_Alcohol        840 non-null    float64
 7   Accident_Severity     840 non-null    object 
 8   Road_Condition        840 non-null    object 
 9   Vehicle_Type          840 non-null    object 
 10  Driver_Age            840 non-null    float64
 11  Driver_Experience     840 non-null    float64
 12  Road_Light_Condition  840 non-null    object 
 13  Accident              840 non-null    float64
dtypes: float64(7), object(7)
memory usage: 92.0+ KB
```

In [77]:
```
df["Traffic_Density"]=df["Traffic_Density"].astype(int)
df["Traffic_Density"]
```

Out[77]:
```
0      1
1      1
2      1
3      2
4      1
      ..
835    2
836    2
837    1
838    2
839    1
Name: Traffic_Density, Length: 840, dtype: int32
```

In [79]:
```
df["Speed_Limit"]=df["Speed_Limit"].astype(int)
df["Speed_Limit"]
```

Out[79]:
```
0      100
1      120
2       60
3       60
4      195
      ...
835     30
836     60
837     30
838     60
839     60
Name: Speed_Limit, Length: 840, dtype: int32
```

In [80]:
```
df["Number_of_Vehicles"]=df["Number_of_Vehicles"].astype(int)
df["Number_of_Vehicles"]
```

Out[80]:
```
0       5
1       3
2       4
3       3
4      11
       ..
835     4
836     4
837     4
838     3
839     4
Name: Number_of_Vehicles, Length: 840, dtype: int32
```

In [81]:
```
df["Driver_Alcohol"]=df["Driver_Alcohol"].astype(int)
df["Driver_Alcohol"]
```

Out[81]:
```
0      0
1      0
2      0
3      0
4      0
      ..
835    0
836    0
837    1
838    0
839    0
Name: Driver_Alcohol, Length: 840, dtype: int32
```

In [82]:
```
df["Driver_Age"]=df["Driver_Age"].astype(int)
df["Driver_Age"]
```

Out[82]:
```
0      51
1      49
2      54
3      34
4      62
       ..
835    23
836    52
837    43
838    25
839    29
Name: Driver_Age, Length: 840, dtype: int32
```

In [83]:
```
df["Driver_Experience"]=df["Driver_Experience"].astype(int)
df["Driver_Experience"]
```

Out[83]:
```
0      48
1      43
2      52
3      31
4      55
       ..
835    15
836    46
837    34
838    19
839    21
Name: Driver_Experience, Length: 840, dtype: int32
```

In [84]:
```
df["Accident"]=df["Accident"].astype(int)
df["Accident"]
```

Out[84]:
```
0      0
1      0
2      0
3      0
4      1
      ..
835    0
836    1
837    0
838    0
839    0
Name: Accident, Length: 840, dtype: int32
```

In [85]:
```
df
```

Out[85]:

	Weather	Road_Type	Time_of_Day	Traffic_Density	Speed_Limit	Number_of_Vehicles	Drive
0	Rainy	City Road	Morning	1	100	5	
1	Clear	Rural Road	Night	1	120	3	
2	Rainy	Highway	Evening	1	60	4	
3	Clear	City Road	Afternoon	2	60	3	
4	Rainy	Highway	Morning	1	195	11	
...	...	...	...	...	...	...	
835	Clear	Highway	Night	2	30	4	
836	Rainy	Rural Road	Evening	2	60	4	
837	Foggy	Highway	Evening	1	30	4	
838	Foggy	Highway	Afternoon	2	60	3	
839	Clear	Highway	Afternoon	1	60	4	

840 rows × 14 columns

Create Pivot Table

In [87]:
```
pivot_table=df.pivot_table(values='Accident',index='Weather',columns='Road_Typ
pivot_table
```

Out[87]:

Road_Type	City Road	Highway	Mountain Road	Rural Road
Weather				
Clear	32	60	5	21
Foggy	9	14	1	4
Rainy	16	28	1	5
Snowy	5	14	1	8
Stormy	3	7	0	5

Find the coutn of road conditions

In [88]:
```
df["Road_Condition"].value_counts()
```

Out[88]:
```
Dry                  442
Icy                  154
Wet                  153
Under Construction    91
Name: Road_Condition, dtype: int64
```

Sort The Value In Ascending Order

In [89]:
```
df.sort_values(by="Accident_Severity")
```

Out[89]:

	Weather	Road_Type	Time_of_Day	Traffic_Density	Speed_Limit	Number_of_Vehicles	Drive
185	Clear	Highway	Morning	0	60	4	
665	Rainy	Highway	Morning	2	60	3	
496	Clear	City Road	Morning	1	120	1	
494	Stormy	Highway	Night	0	50	2	
271	Rainy	City Road	Morning	0	50	3	
...	...	...	...	...	...	...	...
503	Clear	Highway	Night	1	80	3	
679	Snowy	City Road	Evening	0	60	5	
680	Clear	Highway	Night	2	30	2	
683	Foggy	Mountain Road	Night	1	60	1	
419	Clear	Highway	Morning	1	100	4	

840 rows × 14 columns

Find the total count and total sum for Specific

In [53]:
```
df["Number_of_Vehicles"].value_counts()
```

Out[53]:
```
3.0     163
4.0     161
5.0     154
2.0     153
1.0     144
11.0      8
10.0      5
13.0      4
14.0      4
12.0      2
Name: Number_of_Vehicles, dtype: int64
```

In [56]:
```
df["Accident"].sum()
```

Out[56]:
```
239.0
```

In [58]:
```
df["Speed_Limit"].sum()
```

Out[58]:
```
56698.0
```

In [59]:
```
df["Driver_Experience"].sum()
```

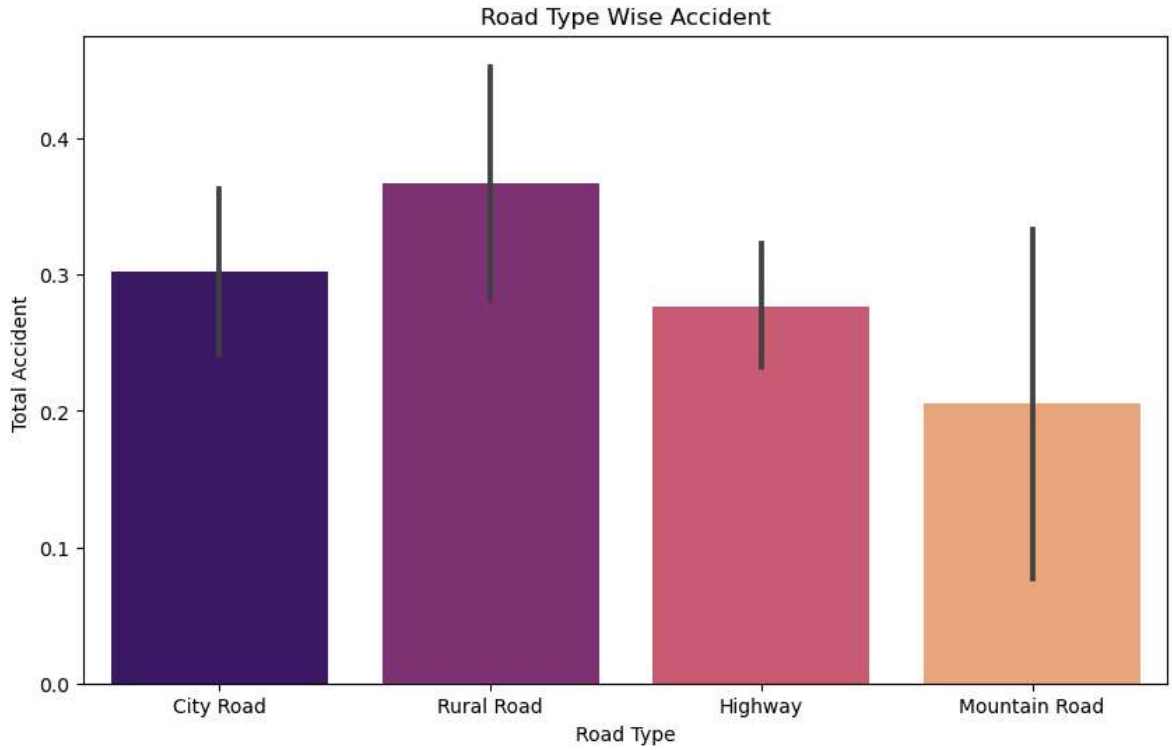
Out[59]:
```
31107.0
```

In [60]:
```
df["Traffic_Density"].sum()
```

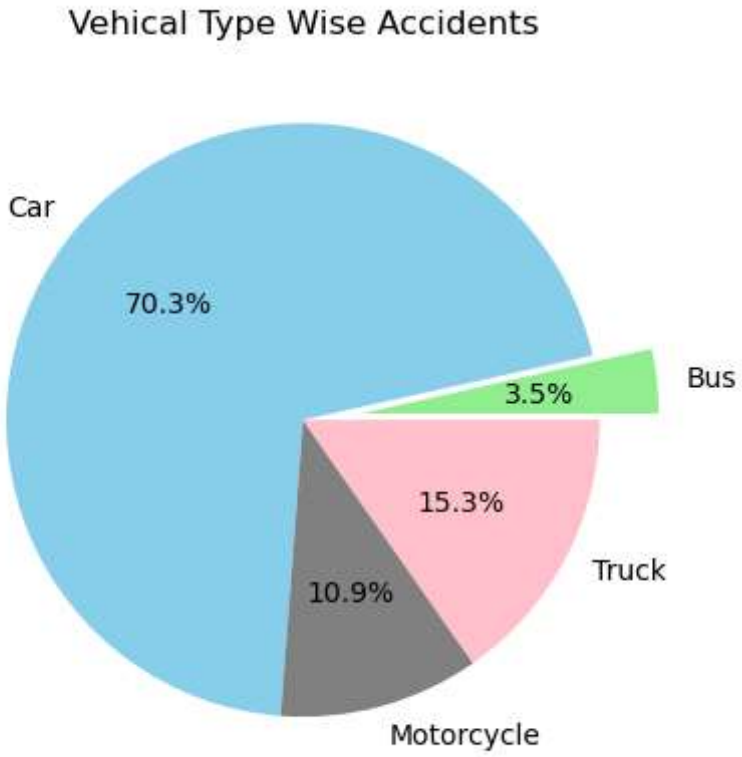
Out[60]:
```
799.0
```

Data Visualization

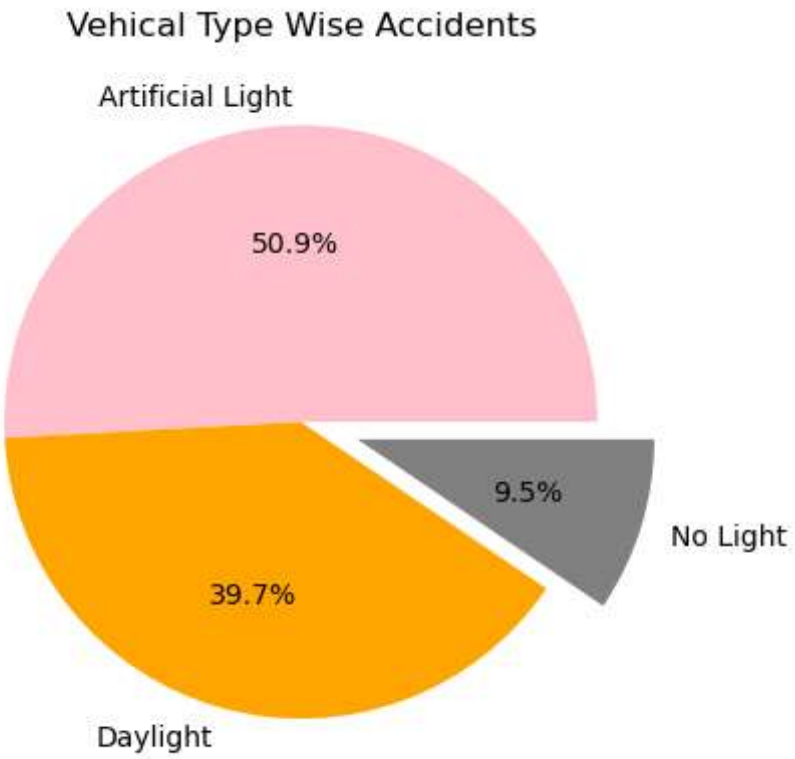
In [86]:
```
plt.figure(figsize=(10,6))
sns.barplot(data=df,x='Road_Type',y='Accident',palette='magma')
plt.title("Road Type Wise Accident")
plt.xlabel("Road Type")
plt.ylabel("Total Accident ")
plt.show()
```



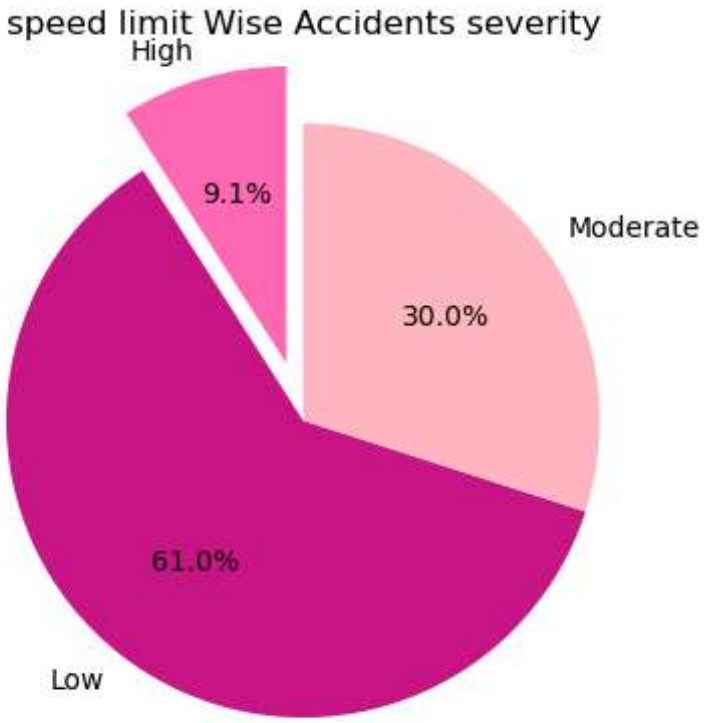
In [139]:
```
VT_wise_Aci=df.groupby("Vehicle_Type")["Accident"].sum()
plt.pie(VT_wise_Aci,labels=VT_wise_Aci.index,autopct='%1.1f%%',explode=[0.2,0,
plt.title("Vehical Type Wise Accidents")
plt.show()
```



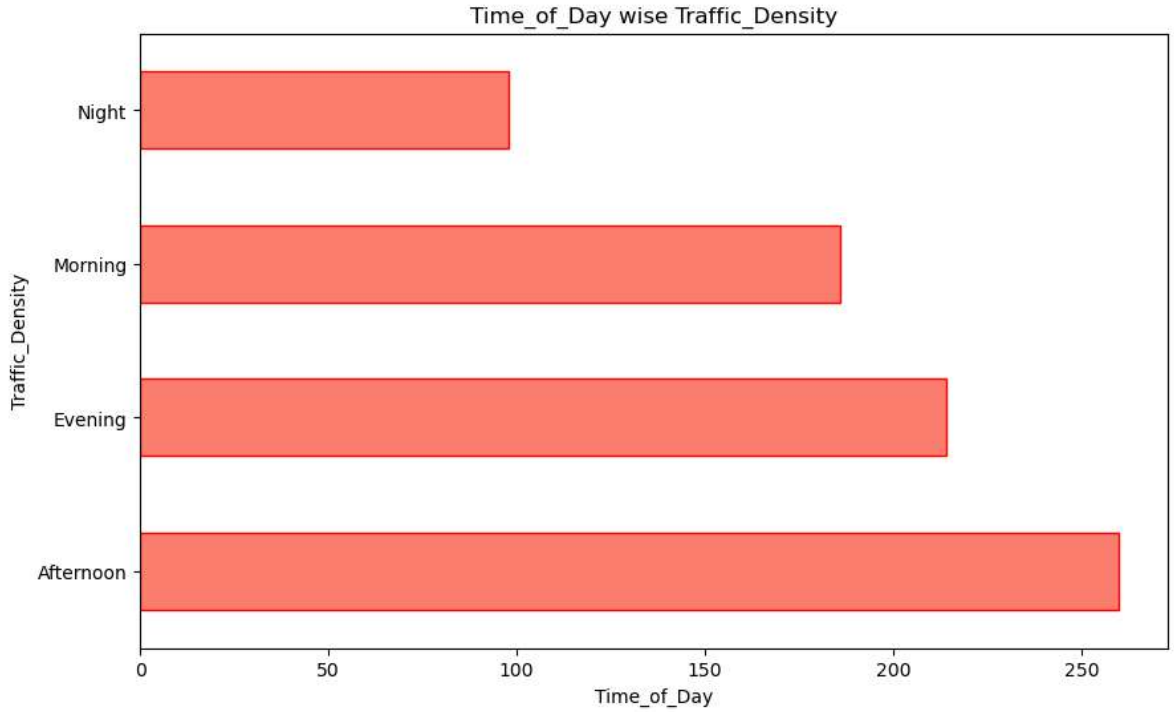
In [138]:
```
VT_wise_Aci=df.groupby("Road_Light_Condition")["Accident"].sum()
plt.pie(VT_wise_Aci,labels=VT_wise_Aci.index,autopct='%1.1f%%',colors=["Pink",
plt.title("Vehical Type Wise Accidents")
plt.show()
```



In [137]:
```
sl_wise_Acis=df.groupby("Accident_Severity")["Speed_Limit"].sum()
plt.pie(sl_wise_Acis,labels=sl_wise_Acis.index,autopct='%1.1f%%',startangle=90
        ,explode=[0.2,0,0])
plt.title("speed limit Wise Accidents severity")
plt.show()
```



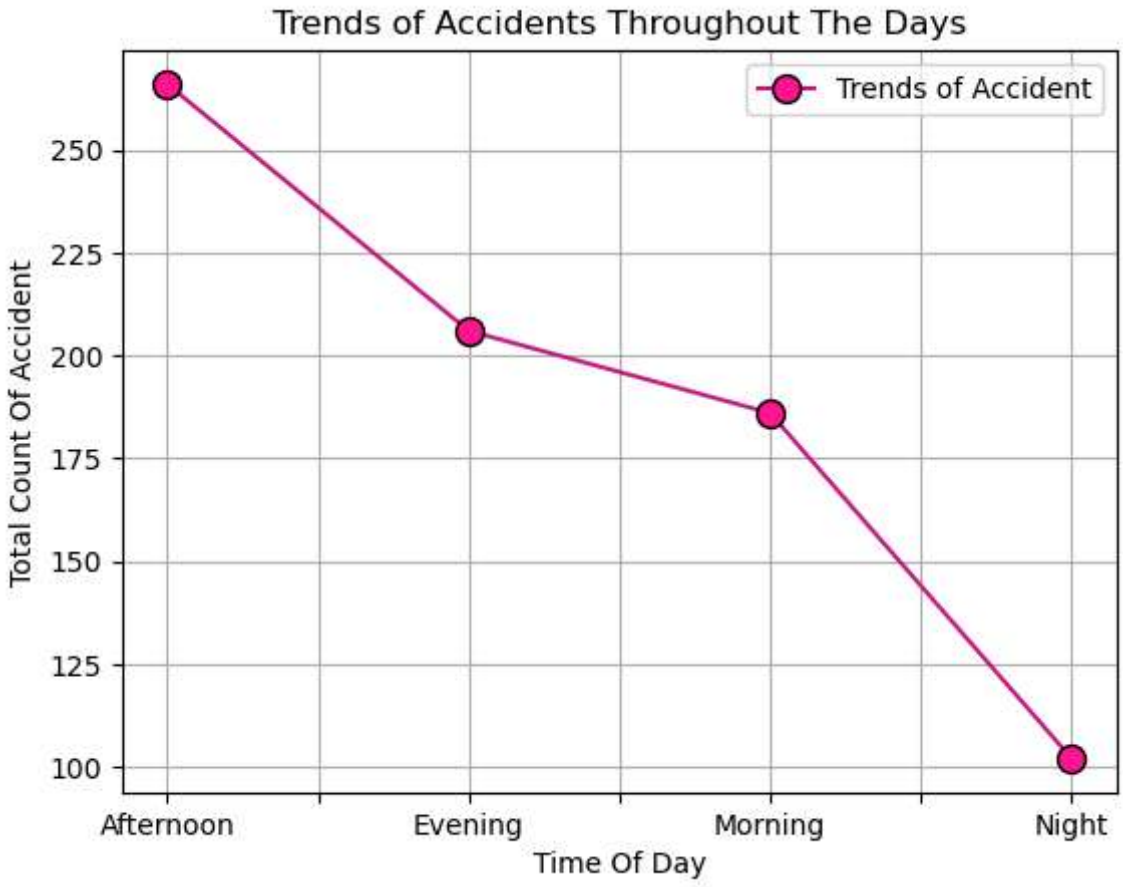
In [113]:
```
TD =df.groupby("Time_of_Day")["Traffic_Density"].sum()
plt.figure(figsize=(10,6))
TD.plot(kind='barh',color="Salmon",edgecolor="Red",label="Traffic Density")
plt.title("Time_of_Day wise Traffic_Density")
plt.xlabel("Time_of_Day")
plt.ylabel("Traffic_Density")
plt.show()
```



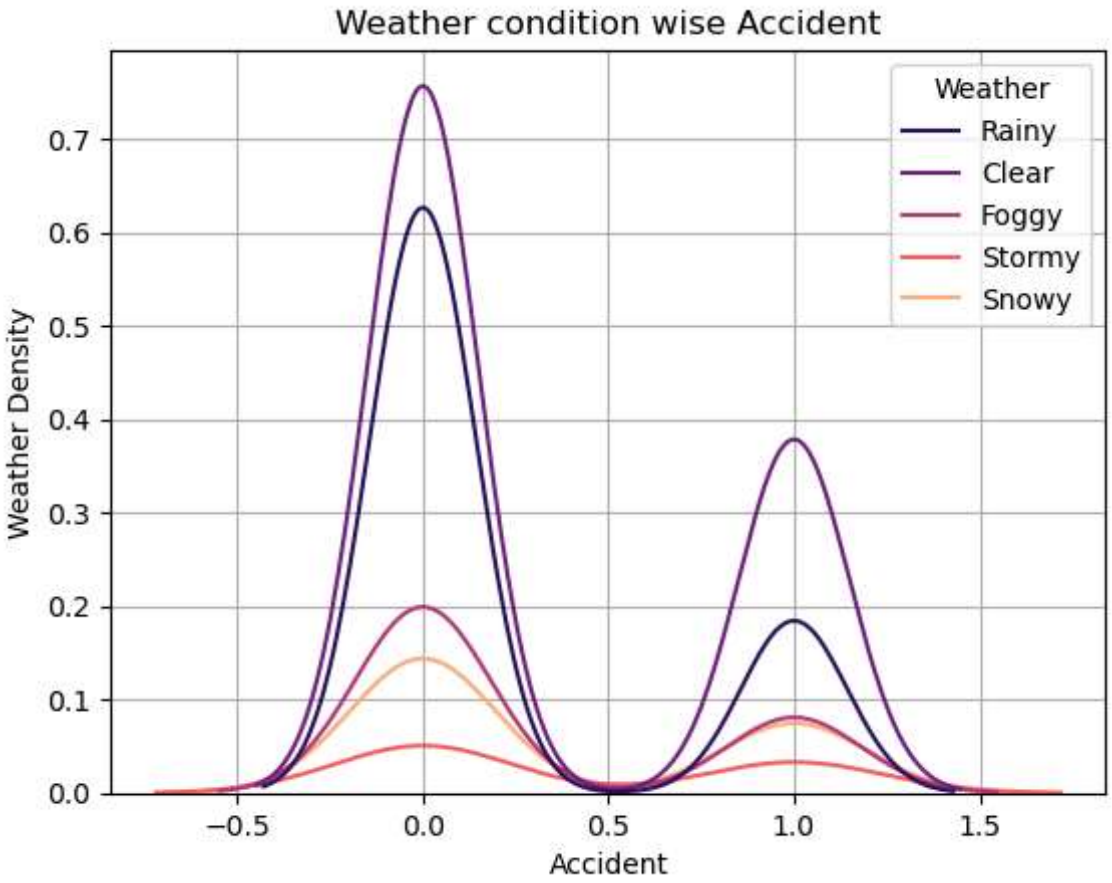
In [41]:
```
Trend =df.groupby("Time_of_Day")["Accident"].count()
Trend
```

Out[41]:
```
Time_of_Day
Afternoon    266
Evening      206
Morning      186
Night        102
Name: Accident, dtype: int64
```

In [118]:
```
Trend.plot(kind="line",color="MediumVioletRed",marker="o",ms='10',mfc="DeepPi
plt.title("Trends of Accidents Throughout The Days")
plt.xlabel("Time Of Day")
plt.ylabel("Total Count Of Accident")
plt.legend()
plt.grid()
plt.show()
```

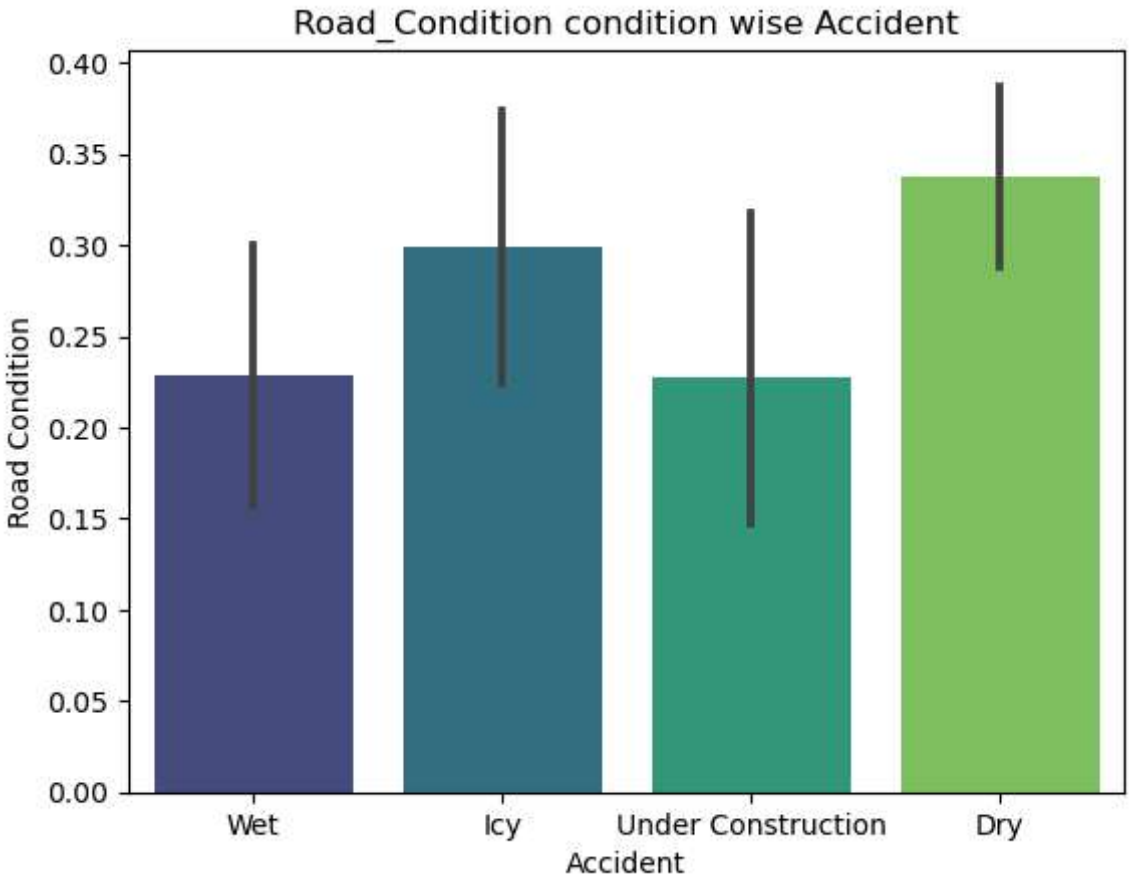


In [126]:
```
sns.kdeplot(data=df,x = "Accident",hue = "Weather",palette="magma")
plt.title("Weather condition wise Accident")
plt.xlabel("Accident")
plt.ylabel("Weather Density")
plt.grid()
plt.show()
```



In [81]:
```
sns.barplot(data=df,x= "Road_Condition",y = "Accident",palette="viridis")
plt.title("Road_Condition condition wise Accident")
plt.xlabel("Accident")
plt.ylabel("Road Condition")

plt.show()
```



In [ ]: