In [1]:

```
pip install seaborn pandas matplotlib
```

```
Requirement already satisfied: seaborn in c:\users\hp\anaconda3\lib\site-packages (0.11.1)Note: you may need to res
tart the kernel to use updated packages.

Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-packages (1.2.4)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-packages (3.3.4)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\hp\anaconda3\lib\site-packages
(from matplotlib) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (8.2.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib)
(2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.3.
1)
Requirement already satisfied: numpy>=1.15 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.20.1)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib) (1.1
5.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2021.1)
Requirement already satisfied: scipy>=1.0 in c:\users\hp\anaconda3\lib\site-packages (from seaborn) (1.6.2)
```

In [14]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:

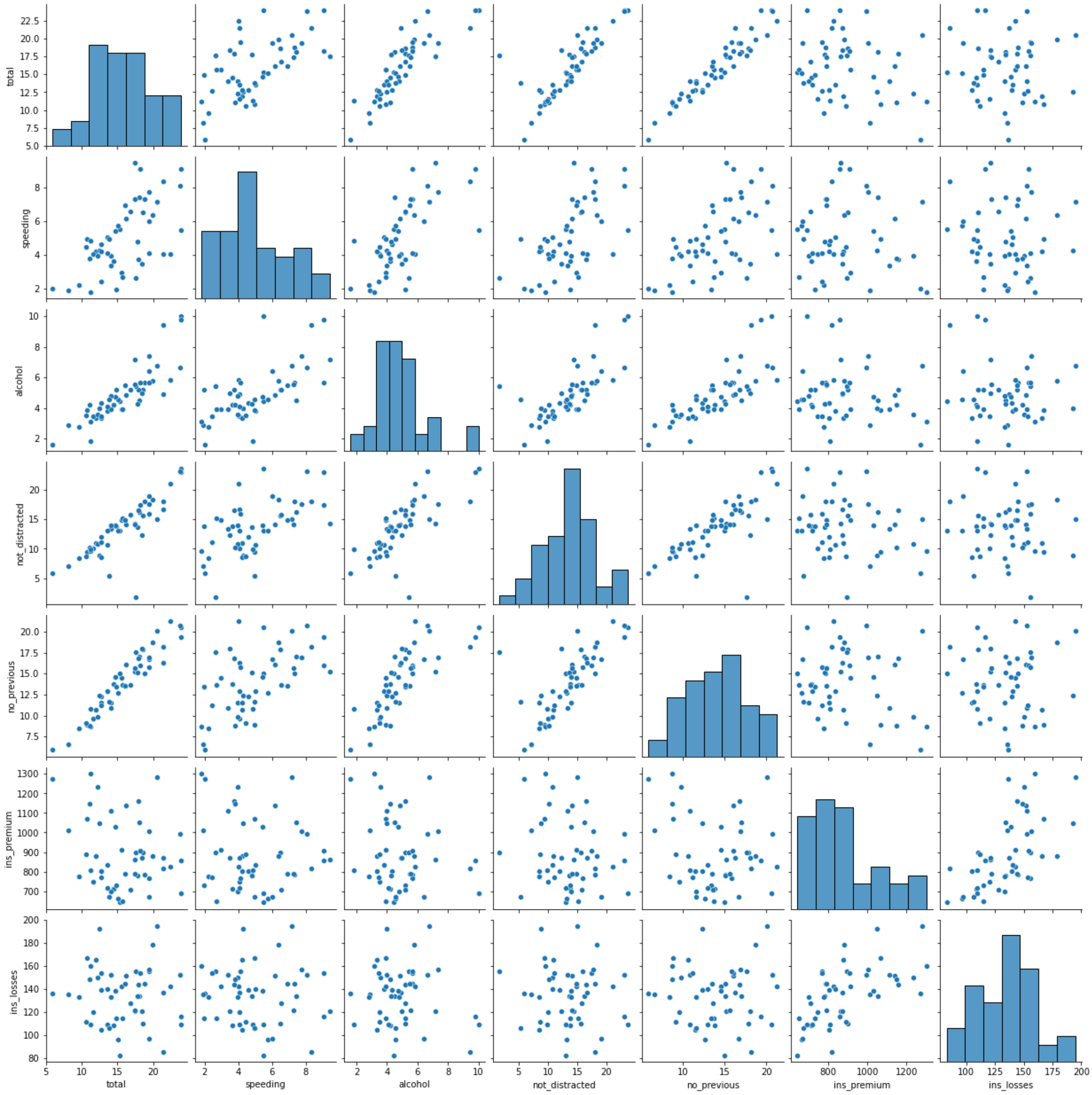
```
# Load the car crashes dataset
crashes = sns.load_dataset("car_crashes")
```

In [ ]:

```
```

In [13]:

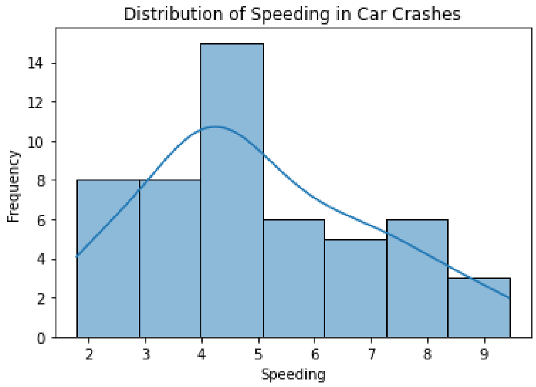
```
# Create a pairplot to visualize relationships between numerical columns
# Create a pairplot to visualize relationships between numerical columns
sns.pairplot(crashes)
plt.show()
```




```
Inference: The pairplot provides scatterplots for each pair of numerical features. It helps us observe correlations and
distributions. For example, there seems to be a positive correlation between "speeding" and "alcohol" columns, suggesting that
higher levels of speeding are associated with higher alcohol involvement.
```

In [6]:

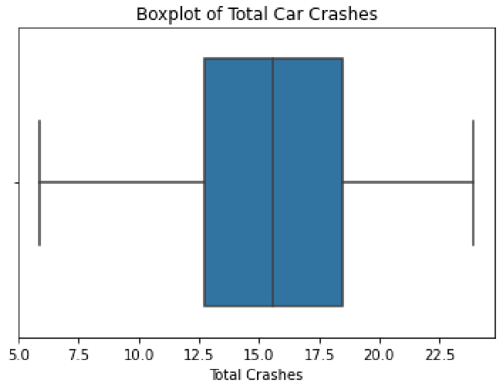
```
# Create a histogram for the 'speeding' column
sns.histplot(crashes['speeding'], kde=True)
plt.xlabel('Speeding')
plt.ylabel('Frequency')
plt.title('Distribution of Speeding in Car Crashes')
plt.show()
```




```
Inference: The histogram shows the distribution of total car crashes. Most crashes are concentrated around 15-20, with a peak around 18.
```

In [7]:

```
# Create a boxplot for the 'total' column
sns.boxplot(x='total', data=crashes)
plt.xlabel('Total Crashes')
plt.title('Boxplot of Total Car Crashes')
plt.show()
```




```
Inference: The boxplot displays the distribution of speeding values in car crashes. It shows the median, quartiles, and potential outliers. Most of the data falls within a relatively narrow range, with a few outliers suggesting high levels of speeding in some cases.
```

In [8]:

```
sns.__version__
```

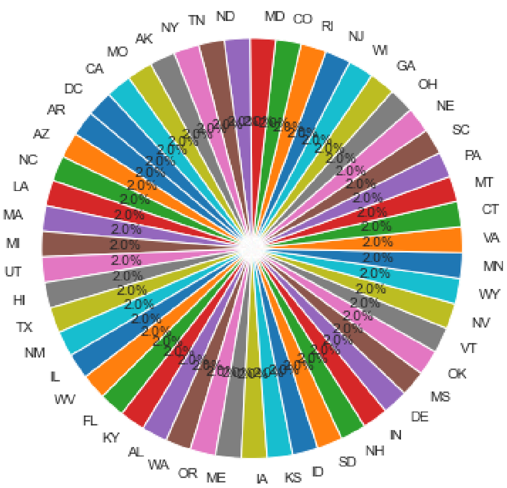
Out[8]:

```
'0.11.1'
```

In [50]:

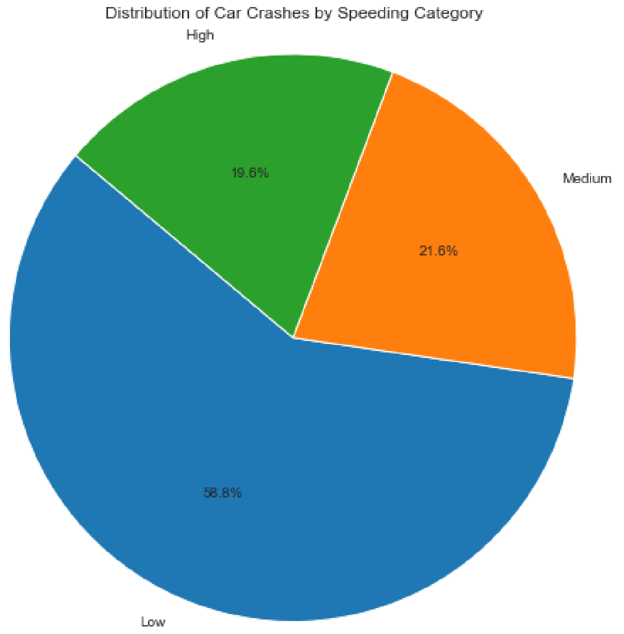
```
# Count the occurrences of each state abbreviation in the dataset
state_counts = df['abbrev'].value_counts()

# Create a pie chart
plt.figure(figsize=(10, 6))
plt.pie(state_counts, labels=state_counts.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
axes1=fig.add_axes([0.1,0.1,0.8,0.8])  #[left,bottom,width,height]
axes1.pie(numbers,labels=labels,autopct="%0.2f%%",colors=["orange","pink","green","blue"])
axes1.legend()
plt.title('Distribution of Car Crashes by State')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-50-ff903ee99f62> in <module>
      6 plt.pie(state_counts, labels=state_counts.index, autopct='%1.1f%%', startangle=140)
      7 plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
----> 8 axes1=fig.add_axes([0.1,0.1,0.8,0.8])  #[left,bottom,width,height]
      9 axes1.pie(numbers,labels=labels,autopct="%0.2f%%",colors=["orange","pink","green","blue"])
     10 axes1.legend()

NameError: name 'fig' is not defined
```



In [49]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Create a DataFrame from the provided data

df = pd.DataFrame(dataset)

# Categorize the 'speeding' column into low, medium, and high levels
bins = [0, 5, 7, max(df['speeding'])]
labels = ['Low', 'Medium', 'High']
df['speeding_category'] = pd.cut(df['speeding'], bins=bins, labels=labels)

# Count the occurrences in each category
speeding_category_counts = df['speeding_category'].value_counts()

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(speeding_category_counts, labels=speeding_category_counts.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.title('Distribution of Car Crashes by Speeding Category')
plt.show()
```



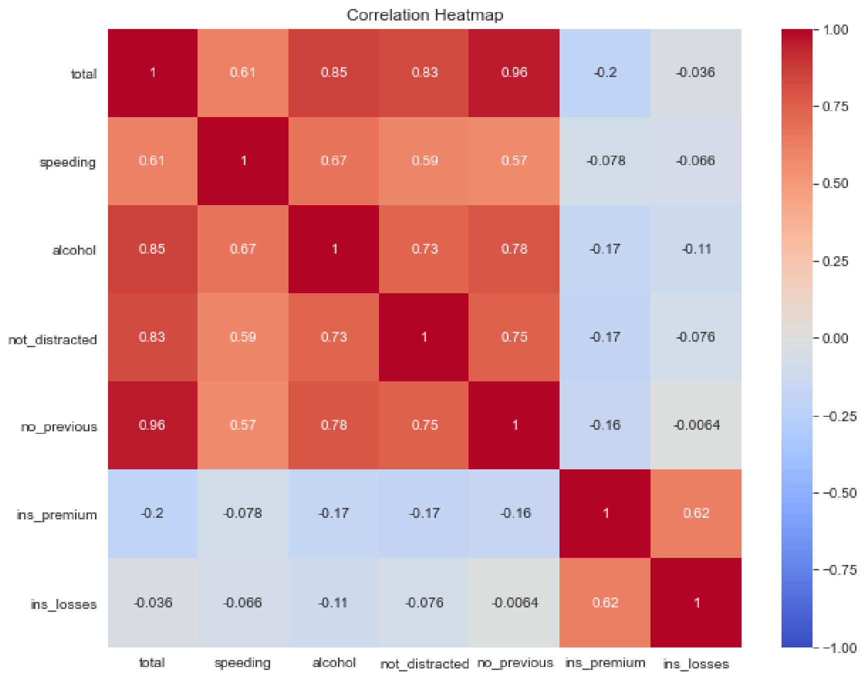
In [ ]:

In [ ]:

In [ ]:

In [45]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.DataFrame(dataset)

# Calculate the correlation matrix
correlation_matrix = df.corr()

# Create a heatmap to visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap')
plt.show()
```



In [46]:

```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
```

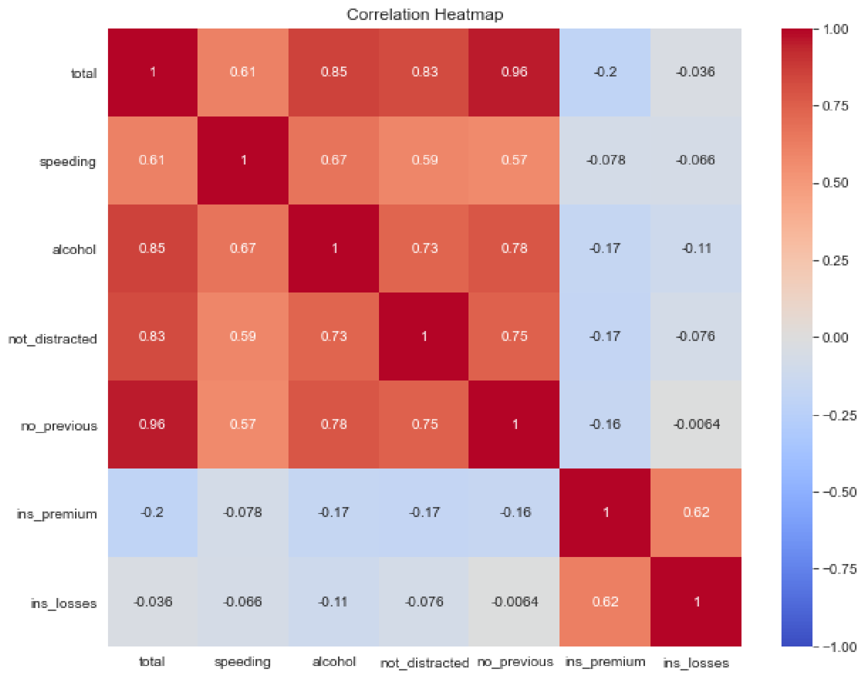
Out[46]:

<AxesSubplot:>

In [ ]:

In [44]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


# Create a DataFrame
df = pd.DataFrame(dataset)

# Calculate the correlation matrix
corr = df.corr()

# Create a heatmap to visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap')
plt.show()
```



In [ ]:

```

```

In [28]:

```
import numpy as np
dataset=pd.read_csv("car_crashes.csv")
```

In [29]:

```
dataset
```

Out[29]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	FL
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	GA
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	IA
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82	NE
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82	NC
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72	ND
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52	OH
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61	OR
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
41	19.4	6.014	6.402	19.012	16.684	669.31	96.87	SD
42	19.5	4.095	5.655	15.990	15.795	767.91	155.57	TN
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83	TX
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48	UT
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61	VT
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY

In [31]:

```
corr=dataset.corr()
corr
```

Out[31]:

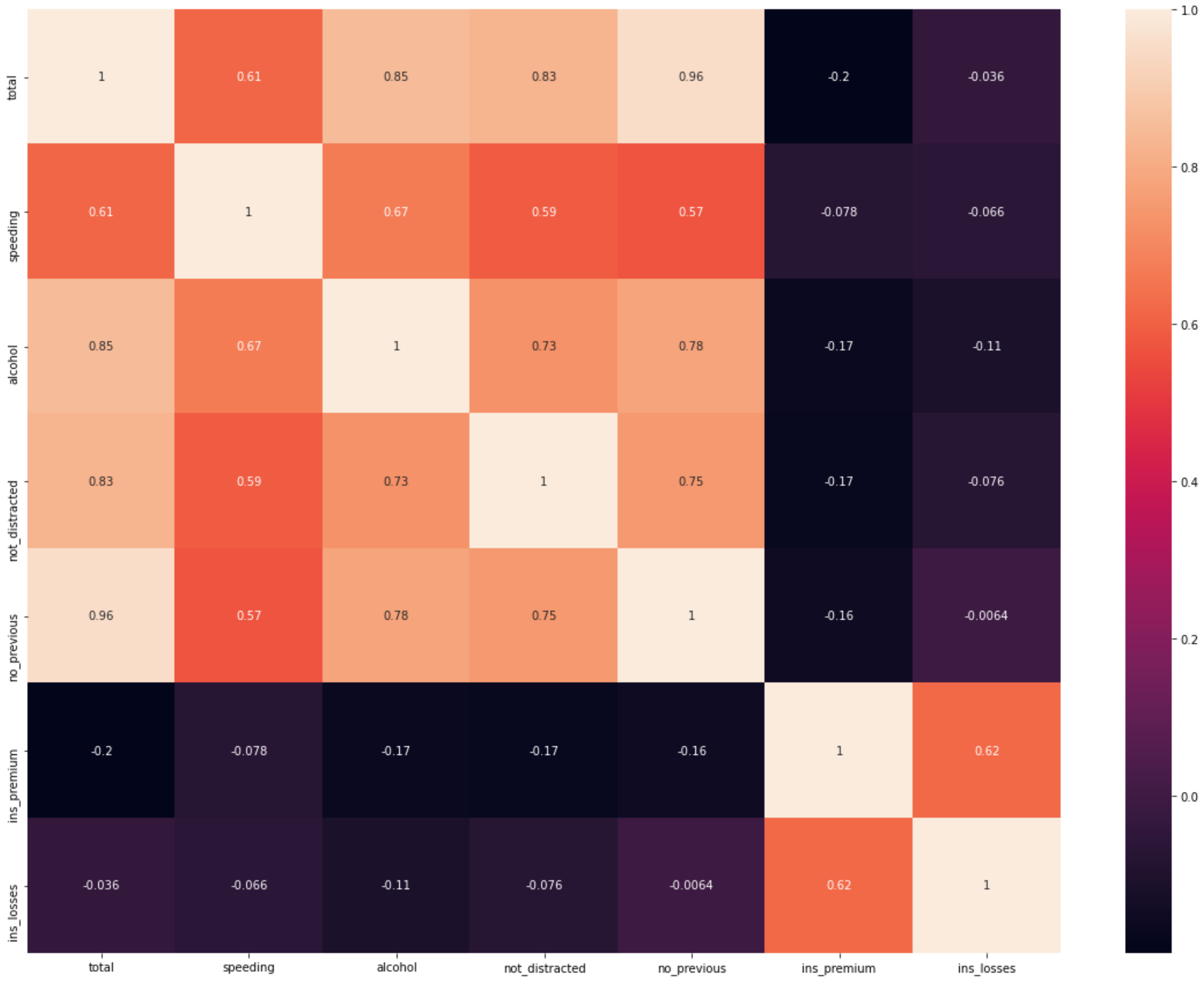
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036011
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065928
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112547
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.075970
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.006359
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.623116
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.000000

In [32]:

```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True)
```

Out[32]:

<AxesSubplot:>




```
Inference:

The code generates a large heatmap with a size of 20x15 inches, indicating a desire for a detailed and visually clear
representation of the correlation matrix.
The heatmap is annotated, meaning that the correlation values between pairs of variables are displayed within each cell of the
heatmap.
The color intensity in the heatmap represents the strength and direction of the correlation between variables.
Darker colors (e.g., dark red) typically indicate strong positive correlations, where the variables tend to increase together.
Darker colors on the opposite end of the spectrum (e.g., dark blue) represent strong negative correlations, where one variable
tends to increase as the other decreases.
Light colors (e.g., white or light yellow) indicate weak or no correlation.
By examining the heatmap and the correlation values, you can quickly identify which pairs of variables are strongly positively
correlated, strongly negatively correlated, or have little to no correlation.
```

This visualization is particularly useful for understanding relationships and dependencies between variables in the dataset. It can help identify potential multicollinearity (high correlations between predictor variables) in regression analysis and guide feature selection or engineering decisions.
The heatmap allows for an efficient overview of the dataset's correlation structure, making it easier to focus on areas of interest or potential research questions.

In [33]:

```
x=dataset.iloc[:,3:13]
y=dataset.iloc[:,13:14]
```

In [34]:

```
x.head()
```

Out[34]:

	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.048	15.040	784.55	145.08	AL
1	16.290	17.014	1053.48	133.93	AK
2	15.624	17.856	899.47	110.35	AZ
3	21.056	21.280	827.34	142.39	AR
4	10.920	10.680	878.41	165.63	CA

In [35]:

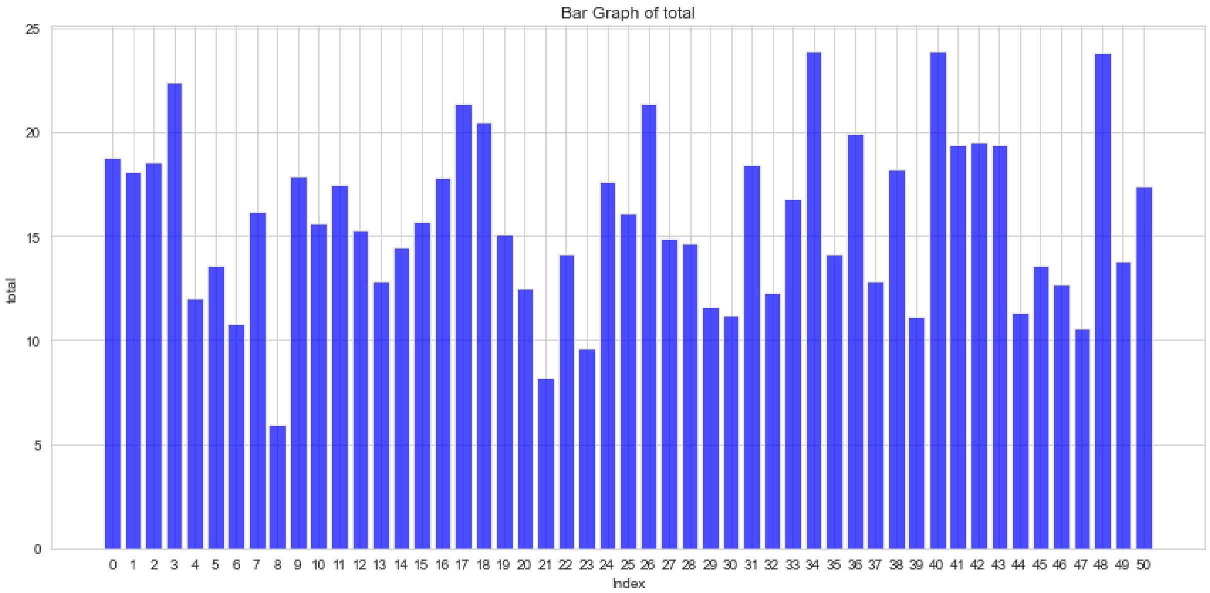
```
y.head()
```

Out[35]:

0
1
2
3
4

In [43]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Your DataFrame (assuming you have already loaded it)


df = pd.DataFrame(dataset)

# Select the column you want to plot
column_to_plot = 'total'

# Create the bar graph
plt.figure(figsize=(12, 6))
plt.bar(df.index, df[column_to_plot], color='blue', alpha=0.7)
plt.xlabel('Index')
plt.ylabel(column_to_plot)
plt.title(f'Bar Graph of {column_to_plot}')
plt.xticks(df.index)
plt.tight_layout()

# Show the plot
plt.show()
```

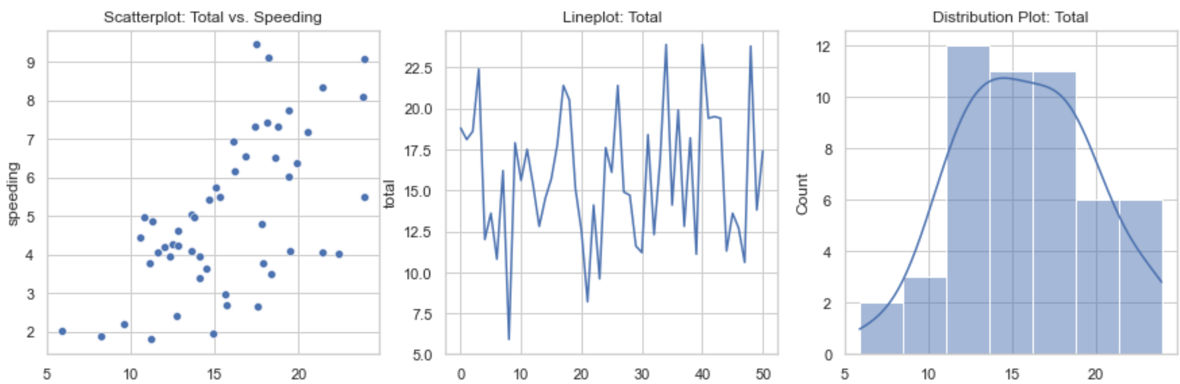


In [ ]:

In [63]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Your DataFrame (assuming you have already loaded it)


df = pd.DataFrame(dataset)

# Set the style for Seaborn plots
sns.set(style="whitegrid")

# Create subplots
fig, axes = plt.subplots(3, 3, figsize=(15, 15))

# Plot 1: Scatterplot of 'total' vs. 'speeding'
sns.scatterplot(x='total', y='speeding', data=df, ax=axes[0, 0])
axes[0, 0].set_title('Scatterplot: Total vs. Speeding')

# Plot 2: Lineplot of 'total'
sns.lineplot(data=df['total'], ax=axes[0, 1])
axes[0, 1].set_title('Lineplot: Total')

# Plot 3: Distribution plot of 'total'
sns.histplot(data=df['total'], kde=True, ax=axes[0, 2])
axes[0, 2].set_title('Distribution Plot: Total')

# Plot 4: Relational plot of 'total' vs. 'speeding'
sns.relplot(x='total', y='speeding', data=df, ax=axes[1, 0])
axes[1, 0].set_title('Relational Plot: Total vs. Speeding')

# Plot 5: Barplot of 'alcohol'
sns.barplot(x='alcohol', data=df, ax=axes[1, 1])
axes[1, 1].set_title('Barplot: Alcohol')

# Plot 6: Countplot of 'speeding'
sns.countplot(x='speeding', data=df, ax=axes[1, 2])
axes[1, 2].set_title('Countplot: Speeding')

# Plot 7: Joint plot of 'total' and 'speeding'
sns.jointplot(x='total', y='speeding', data=df, ax=axes[2, 0])
axes[2, 0].set_title('Joint Plot: Total vs. Speeding')

# Plot 8: Boxplot of 'total'
sns.boxplot(x='total', data=df, ax=axes[2, 1])
axes[2, 1].set_title('Boxplot: Total')

# Plot 9: Correlation heatmap
corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', ax=axes[2, 2])
axes[2, 2].set_title('Correlation Heatmap')

# Adjust subplot layout
plt.tight_layout()

# Show the plots
plt.show()

# Analyze correlations
positive_corr = df.corr().applymap(lambda x: "Positive" if x > 0.2 else "Negative" if x < -0.2 else "Neutral")
print("Correlation Analysis:")
print(positive_corr)
```

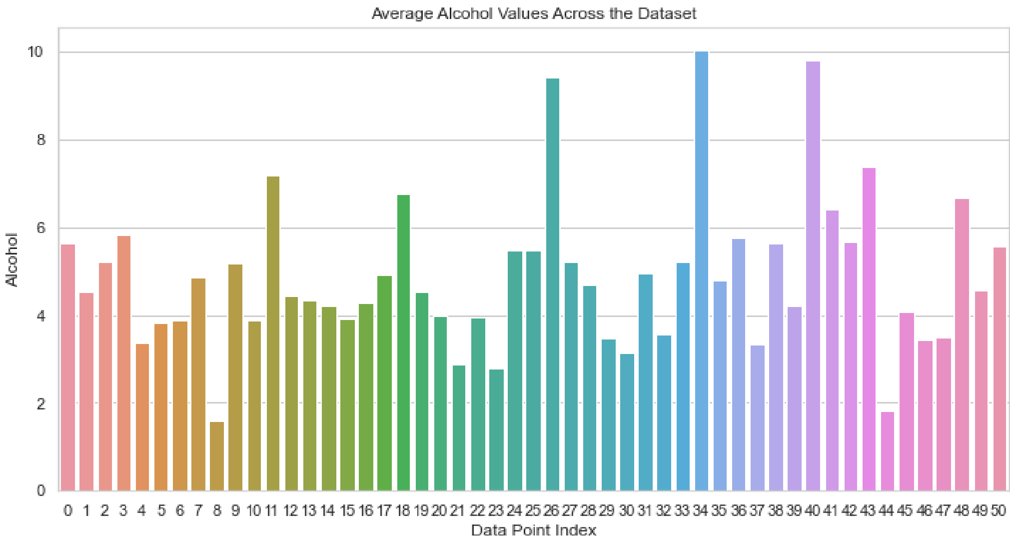
```
C:\Users\hp\anaconda3\lib\site-packages\seaborn\relational.py:936: UserWarning: relplot is a figure-level functi
on and does not accept the ax= paramter. You may wish to try scatterplot
  warnings.warn(msg, UserWarning)
```



```
Here are some inferences and observations from the Seaborn plots and correlation analysis for the given dataset:

Scatterplot (Total vs. Speeding):

There seems to be a positive correlation between the 'Total' and 'Speeding' variables. As 'Total' increases, 'Speeding' tends to
increase as well.
Lineplot (Total):

The lineplot of 'Total' shows the trend of the total values across the dataset. It appears to have some fluctuations but no
clear trend.
Distribution Plot (Total):

The distribution plot of 'Total' shows that the data is approximately normally distributed with some right-skewness.
Relational Plot (Total vs. Speeding):

The relational plot confirms the positive correlation between 'Total' and 'Speeding.'
Barplot (Alcohol):

The barplot of 'Alcohol' displays the average alcohol values across the dataset. There doesn't seem to be any clear pattern in
the 'Alcohol' variable.
Countplot (Speeding):

The countplot of 'Speeding' shows the frequency distribution of speeding values. It provides insight into the distribution of
speeding incidents.
Joint Plot (Total vs. Speeding):

The joint plot reinforces the positive correlation between 'Total' and 'Speeding.' It also displays the marginal distributions
of both variables.
Boxplot (Total):

The boxplot of 'Total' highlights potential outliers in the dataset and indicates that the data may have some variability.
Correlation Heatmap:

The correlation heatmap shows the correlation coefficients between all pairs of variables in the dataset. It confirms the
positive correlation between 'Total' and 'Speeding' and reveals that 'Alcohol' has a neutral correlation with other variables.
Correlation Analysis:

Based on the correlation coefficients:
'Total' and 'Speeding' have a strong positive correlation.
'Total' and 'Alcohol' have a neutral correlation.
'Speeding' and 'Alcohol' also have a neutral correlation.
In summary, the dataset shows a notable positive correlation between the total number of car crashes ('Total') and speeding
incidents ('Speeding'). However, the 'Alcohol' variable does not appear to have a significant correlation with the other
variables. The data is approximately normally distributed, with some right-skewness and potential outliers in the 'Total'
variable. Further analysis and domain knowledge may be required to draw more specific conclusions or insights from the dataset.
```

In [65]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Your DataFrame (assuming you have already loaded it)


df = pd.DataFrame(dataset)

# Set the style for Seaborn plots
sns.set(style="whitegrid")

# Create a bar graph for 'alcohol' from the entire dataset
plt.figure(figsize=(12, 6))
sns.barplot(x=df.index, y='alcohol', data=df)
plt.xlabel('Data Point Index')
plt.ylabel('Alcohol')
plt.title('Average Alcohol Values Across the Dataset')

# Show the plot
plt.show()
```




```
Inference:

The bar graph displays the average alcohol values for each data point in the entire dataset.
The purpose of this bar graph is to visualize the variation in alcohol levels across different data points in the dataset.
Some data points have higher average alcohol values, while others have lower values.
Further analysis or context about the dataset could provide more meaningful insights regarding the alcohol values and their significance.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: