CHENNAM RUSHWANTH-PRODIGY INFO TECH-DATA SCIENCE INTERN-TASK 5

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [6]:
```
df=pd.read_csv("Crash_Data.csv")
df
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_16148\530237032.py:1: DtypeWarning: Column
s (10,14,15,16,17) have mixed types. Specify dtype option on import or set low_memory
=False.
  df=pd.read_csv("Crash_Data.csv")
```

Out[6]:

	Crash ID	State	Month	Year	Dayweek	Time	Crash Type	Bus Involvement	Heavy Rigid Truck Involvement	Articu Involve
0	20212133	Vic	9	2021	Sunday	0:30	Single	NaN	NaN	
1	20214022	SA	9	2021	Saturday	23:31	Multiple	No	No	
2	20212096	Vic	9	2021	Saturday	23:00	Single	NaN	NaN	
3	20212145	Vic	9	2021	Saturday	22:25	Single	NaN	NaN	
4	20212075	Vic	9	2021	Saturday	5:15	Single	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
52838	19891246	NSW	1	1989	Wednesday	17:05	Single	Yes	NaN	
52839	19895088	WA	1	1989	Monday	6:00	Single	No	NaN	
52840	19895088	WA	1	1989	Monday	6:00	Single	No	NaN	
52841	19895088	WA	1	1989	Monday	6:00	Single	No	NaN	
52842	19896063	Tas	1	1989	Tuesday	12:40	Multiple	No	NaN	

52843 rows × 23 columns

In [8]:
```
df.head()
```

Out[8]:

	Crash ID	State	Month	Year	Dayweek	Time	Crash Type	Bus Involvement	Heavy Rigid Truck Involvement	Articulated Truck Involvement
0	20212133	Vic	9	2021	Sunday	0:30	Single	NaN	NaN	NaN
1	20214022	SA	9	2021	Saturday	23:31	Multiple	No	No	No
2	20212096	Vic	9	2021	Saturday	23:00	Single	NaN	NaN	NaN
3	20212145	Vic	9	2021	Saturday	22:25	Single	NaN	NaN	NaN
4	20212075	Vic	9	2021	Saturday	5:15	Single	NaN	NaN	NaN

5 rows × 23 columns

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52843 entries, 0 to 52842
Data columns (total 23 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   Crash ID                       52843 non-null  int64 
 1   State                          52843 non-null  object
 2   Month                          52843 non-null  int64 
 3   Year                           52843 non-null  int64 
 4   Dayweek                        52843 non-null  object
 5   Time                           52803 non-null  object
 6   Crash Type                     52843 non-null  object
 7   Bus Involvement                52821 non-null  object
 8   Heavy Rigid Truck Involvement  32328 non-null  object
 9   Articulated Truck Involvement  52821 non-null  object
 10  Speed Limit                    52141 non-null  object
 11  Road User                      52843 non-null  object
 12  Gender                         52816 non-null  object
 13  Age                            52843 non-null  int64 
 14  National Remoteness Areas      6878 non-null   object
 15  SA4 Name 2016                  6892 non-null   object
 16  National LGA Name 2017         6893 non-null   object
 17  National Road Type             6877 non-null   object
 18  Christmas Period               52843 non-null  object
 19  Easter Period                  52843 non-null  object
 20  Age Group                      52753 non-null  object
 21  Day of week                    52843 non-null  object
 22  Time of day                    52843 non-null  object
dtypes: int64(4), object(19)
memory usage: 9.3+ MB
```

In [12]:
```
df.describe()
```

Out[12]:

	Crash ID	Month	Year	Age
count	5.284300e+04	52843.000000	52843.000000	52843.000000
mean	2.003021e+07	6.568685	2002.729974	39.662377
std	9.383542e+04	3.457347	9.378570	21.806198
min	1.989100e+07	1.000000	1989.000000	-9.000000
25%	1.995111e+07	4.000000	1995.000000	22.000000
50%	2.002144e+07	7.000000	2002.000000	34.000000
75%	2.010408e+07	10.000000	2010.000000	55.000000
max	2.021801e+07	12.000000	2021.000000	101.000000

In [14]:
```
numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']

numeric_df = df.select_dtypes(include=numerics)
len(numeric_df.columns)
```

Out[14]:
```
4
```

In [16]:
```
missing_percentages = df.isna().sum().sort_values(ascending=False) / len(df)
missing_percentages
```

Out[16]:
```
National Road Type                0.869860
National Remoteness Areas         0.869841
SA4 Name 2016                     0.869576
National LGA Name 2017            0.869557
Heavy Rigid Truck Involvement     0.388225
Speed Limit                       0.013285
Age Group                         0.001703
Time                              0.000757
Gender                            0.000511
Bus Involvement                   0.000416
Articulated Truck Involvement     0.000416
Crash ID                          0.000000
Day of week                       0.000000
Easter Period                     0.000000
Christmas Period                  0.000000
Road User                         0.000000
Age                               0.000000
State                             0.000000
Crash Type                        0.000000
Dayweek                           0.000000
Year                              0.000000
Month                             0.000000
Time of day                       0.000000
dtype: float64
```

In [18]:
```
type(missing_percentages)
```

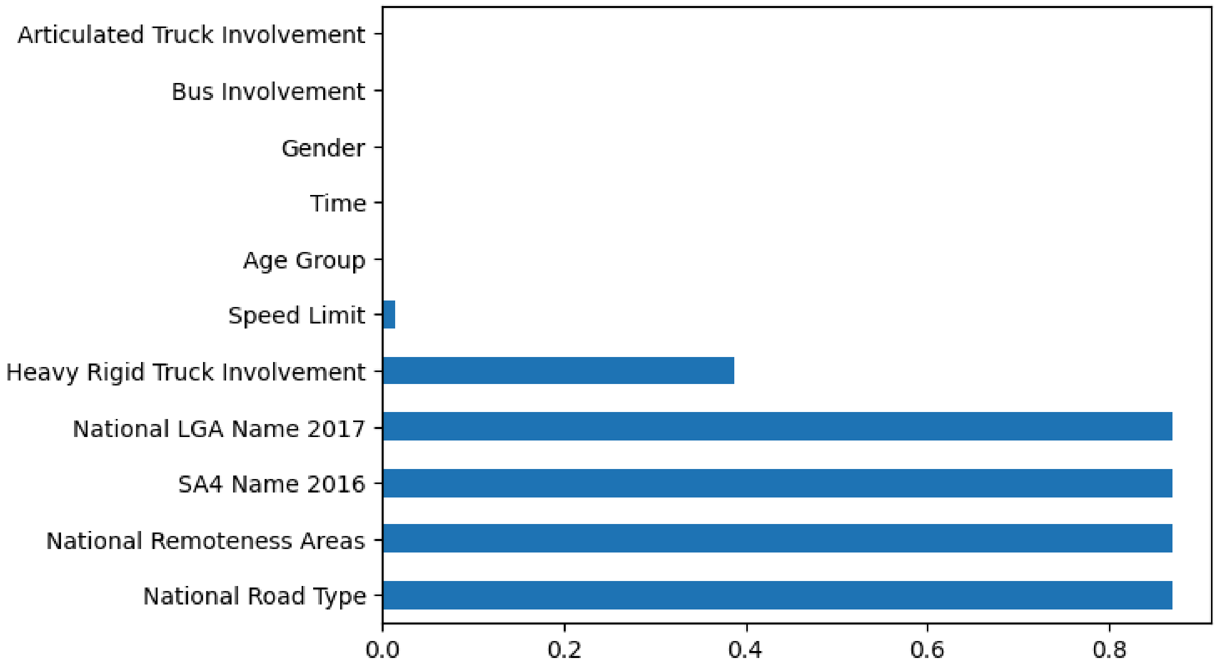
Out[18]:
```
pandas.core.series.Series
```

In [20]:

```
missing_percentages[missing_percentages != 0].plot(kind='barh')
```

Out[20]:

```
<Axes: >
```



In [22]:

```
df.columns
```

Out[22]:

```
Index(['Crash ID', 'State', 'Month', 'Year', 'Dayweek', 'Time', 'Crash Type',
       'Bus Involvement', 'Heavy Rigid Truck Involvement',
       'Articulated Truck Involvement', 'Speed Limit', 'Road User', 'Gender',
       'Age', 'National Remoteness Areas', 'SA4 Name 2016',
       'National LGA Name 2017', 'National Road Type', 'Christmas Period',
       'Easter Period', 'Age Group', 'Day of week', 'Time of day'],
      dtype='object')
```

In [24]:

```
df.State
```

Out[24]:

```
0        Vic
1         SA
2        Vic
3        Vic
4        Vic
        ... 
52838    NSW
52839     WA
52840     WA
52841     WA
52842    Tas
Name: State, Length: 52843, dtype: object
```

In [26]:

```
State = df.State.unique()
len(State)
```

Out[26]:

```
8
```

In [28]:

```
State_by_accident = df.State.value_counts()
State_by_accident
```

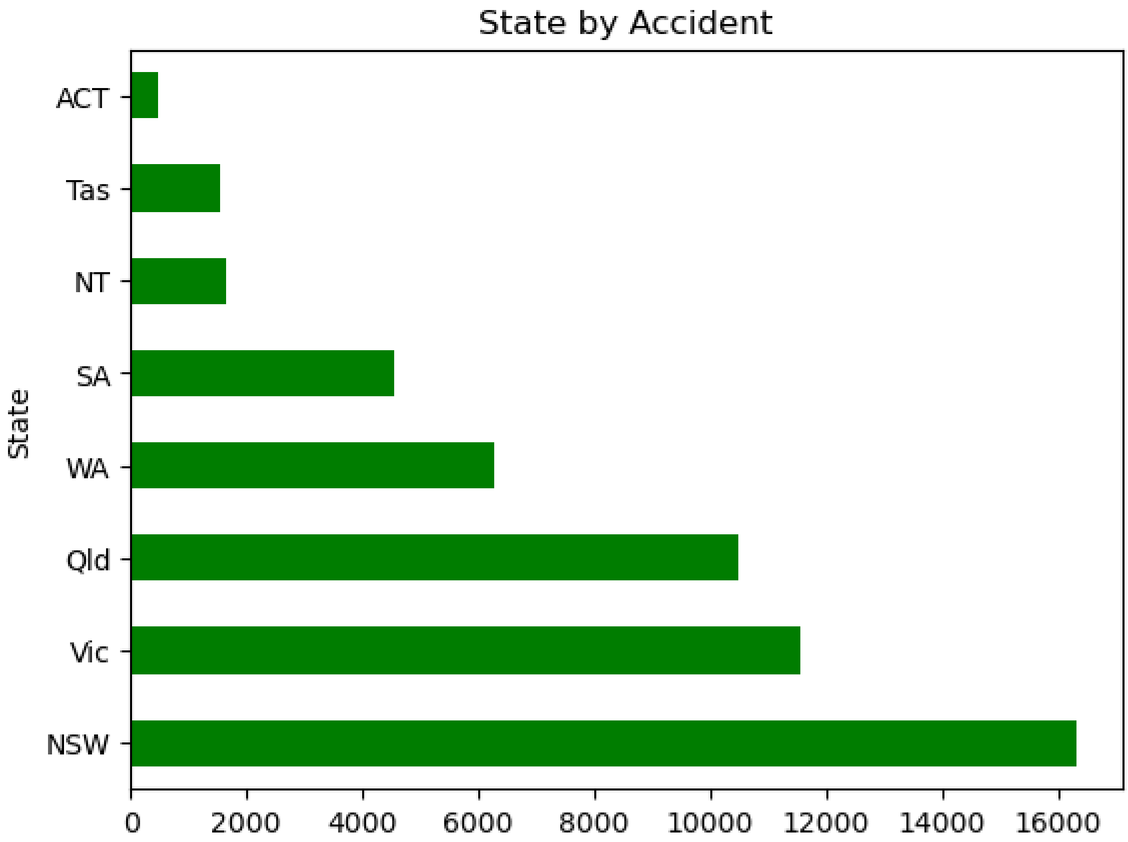
Out[28]:
```
State
NSW    16293
Vic    11562
Qld    10495
WA      6276
SA      4547
NT      1642
Tas     1550
ACT      478
Name: count, dtype: int64
```

In [36]:
```
type(State_by_accident)
```

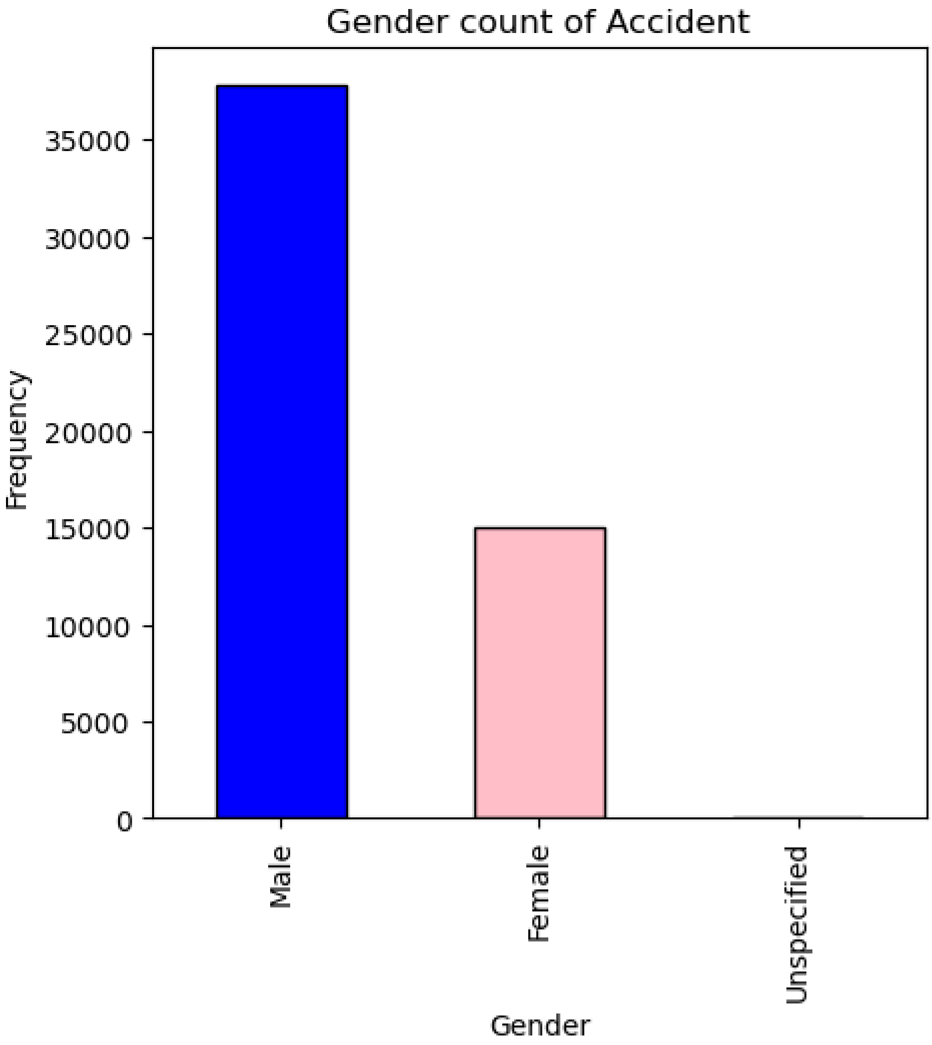
Out[36]:
```
pandas.core.series.Series
```

In [38]:
```
State_by_accident[:20].plot(kind='barh',color='green',title="State by Accident")
```

Out[38]:
```
<Axes: title={'center': 'State by Accident'}, ylabel='State'>
```



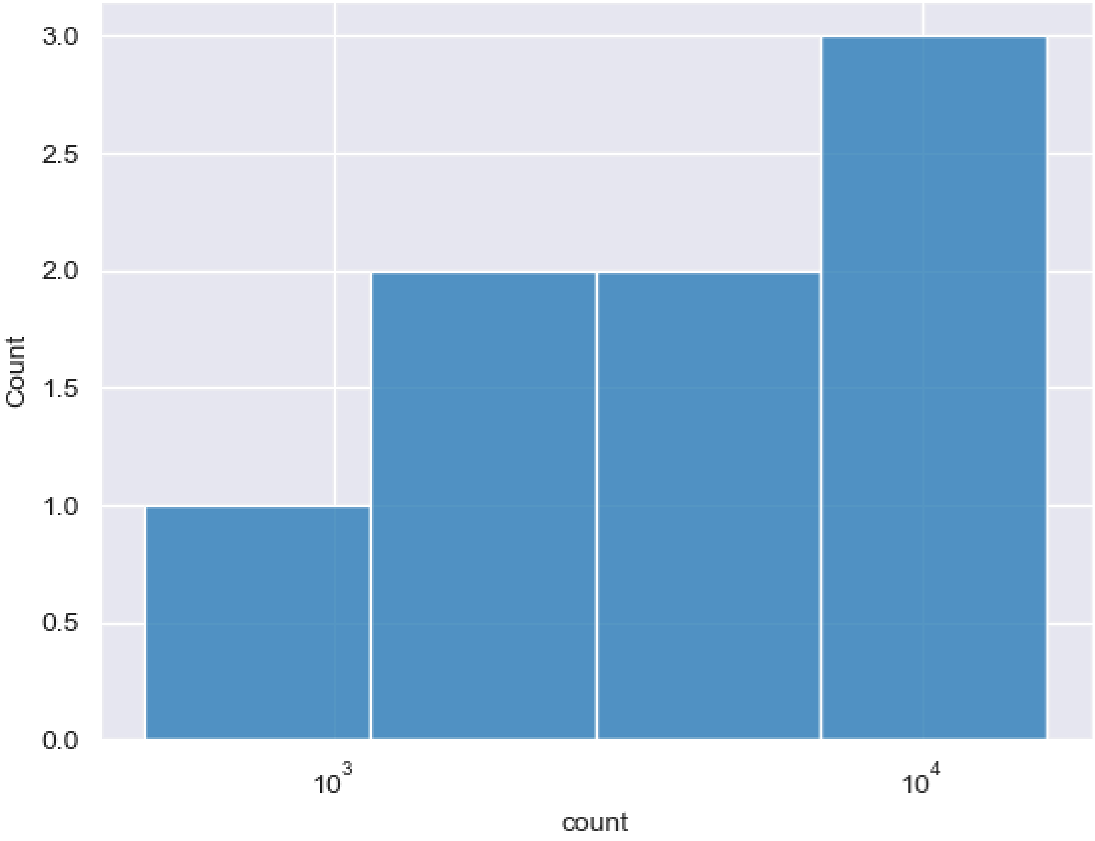
In [40]:
```
# Bar chart for Gender in the dataset

plt.figure(figsize = (5,5))
df['Gender'].value_counts().plot(kind='bar', color =['blue','pink'],edgecolor='black',
plt.xlabel("Gender")
plt.ylabel("Frequency")
plt.show()
```



In [42]:
```
import seaborn as sns
sns.set_style("darkgrid")
```

In [44]:
```
sns.histplot(State_by_accident, log_scale=True)
```

Out[44]:
```
<Axes: xlabel='count', ylabel='Count'>
```



In [46]:
```
State_by_accident[State_by_accident == 1]
```

Out[46]:
```
Series([], Name: count, dtype: int64)
```

In [48]:
```
df.Time
```

Out[48]:
```
0         0:30
1        23:31
2        23:00
3        22:25
4         5:15
         ...  
52838    17:05
52839     6:00
52840     6:00
52841     6:00
52842    12:40
Name: Time, Length: 52843, dtype: object
```

In [50]:
```
df.Time = pd.to_datetime(df.Time)
```

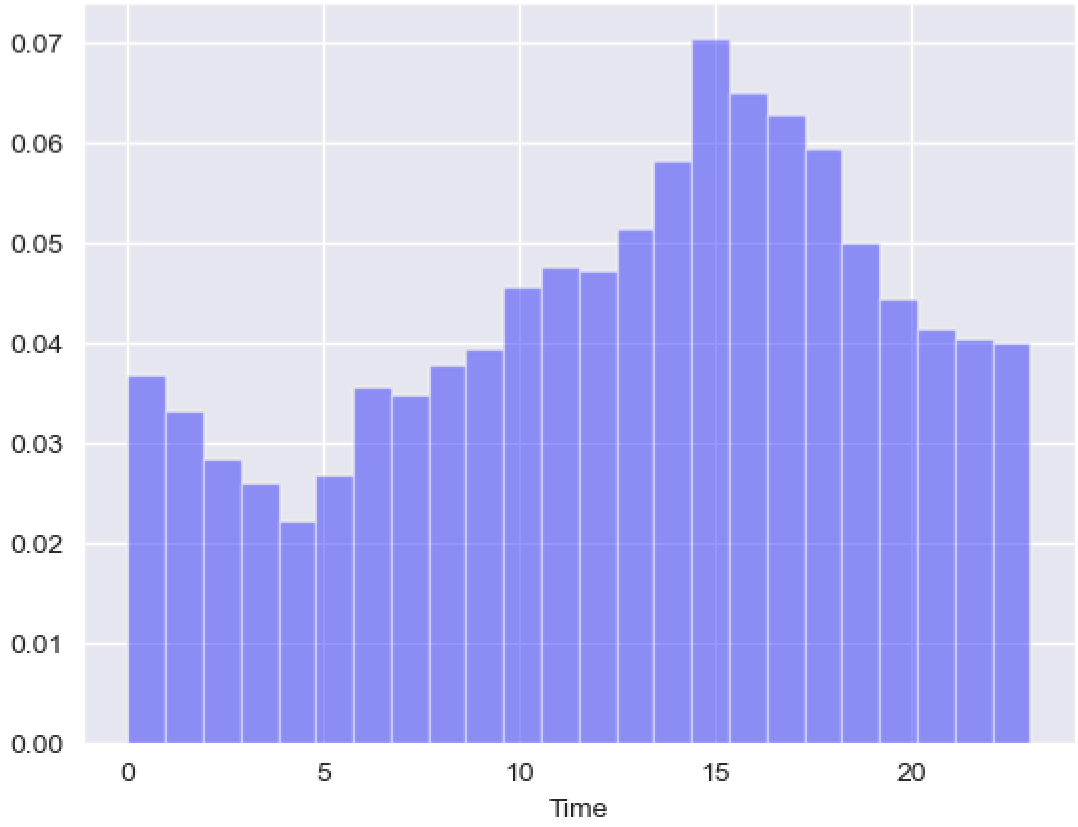
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_16148\3099346244.py:1: UserWarning: Could
not infer format, so each element will be parsed individually, falling back to `dateu
til`. To ensure parsing is consistent and as-expected, please specify a format.
  df.Time = pd.to_datetime(df.Time)
```

In [52]:
```
sns.distplot(df.Time.dt.hour, bins=24, kde=False, norm_hist=True, color="blue")
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_16148\3759089705.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Time.dt.hour, bins=24, kde=False, norm_hist=True, color="blue")
```

Out[52]:

```
<Axes: xlabel='Time'>
```



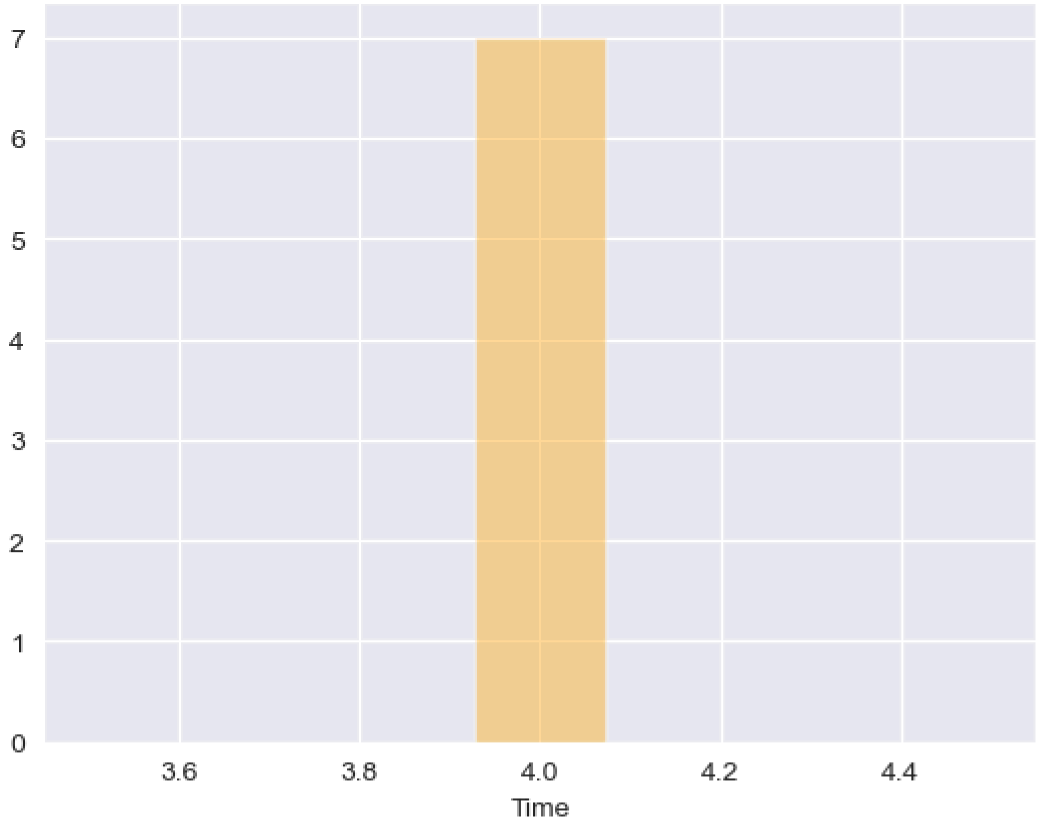
In [54]:

```
sns.distplot(df.Time.dt.dayofweek, bins=7, kde=False, norm_hist=True, color="orange")
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_16148\3691396186.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Time.dt.dayofweek, bins=7, kde=False, norm_hist=True, color="orang
e")
```

Out[54]:

```
<Axes: xlabel='Time'>
```



In [56]:

```
df_num=df.select_dtypes(np.number)
col_name=[]
length=[]

for i in df_num.columns:
    col_name.append(i)
    length.append(len(df_num[i].unique()))
df_2=pd.DataFrame(zip(col_name,length),columns=['feature','count_of_unique_values'])
df_2
```

Out[56]:

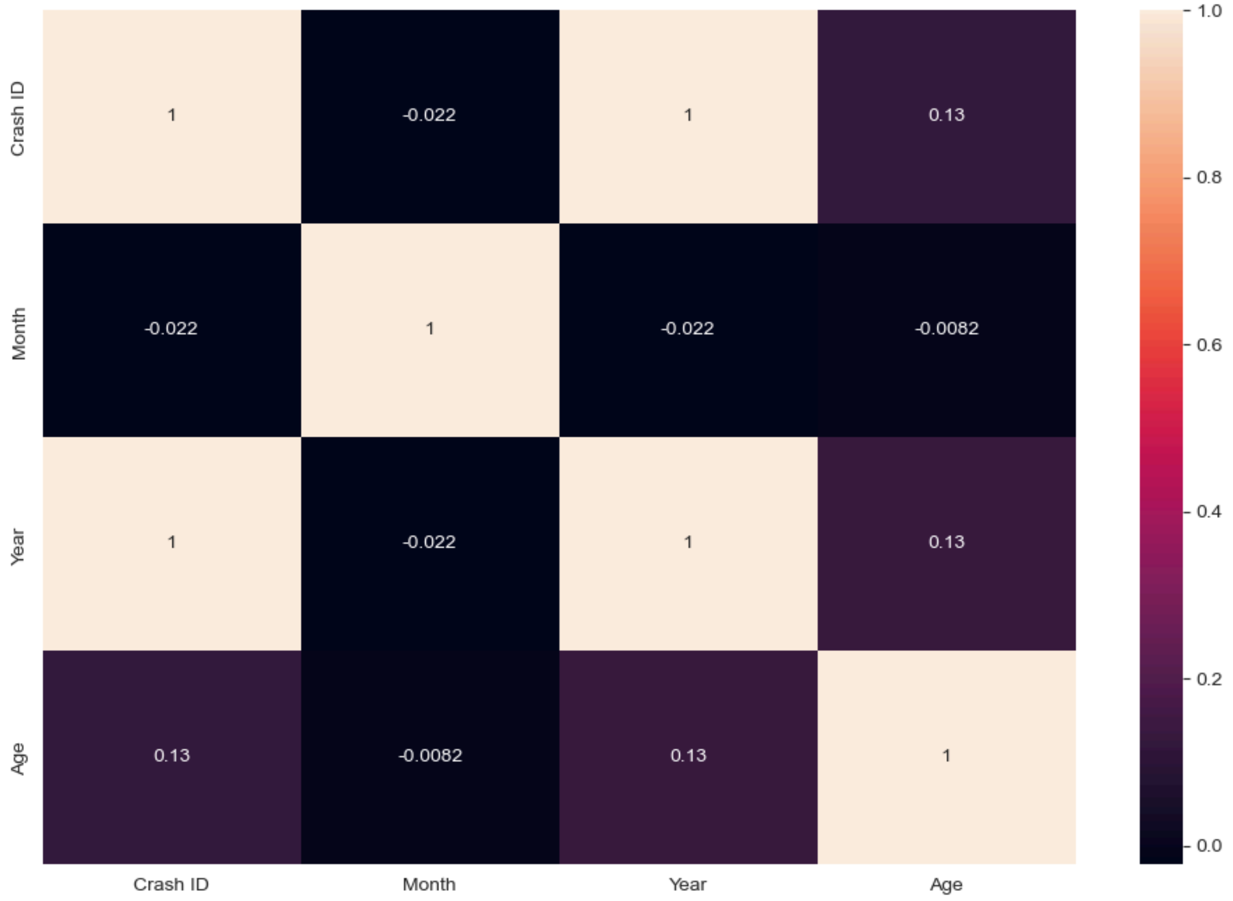
	feature	count_of_unique_values
0	Crash ID	47567
1	Month	12
2	Year	33
3	Age	103

In [58]:

```
#Correlation Matrix
plt.figure(figsize=(12 ,8))
sns.heatmap(df_num.corr() , annot=True)
```

Out[58]:

```
<Axes: >
```

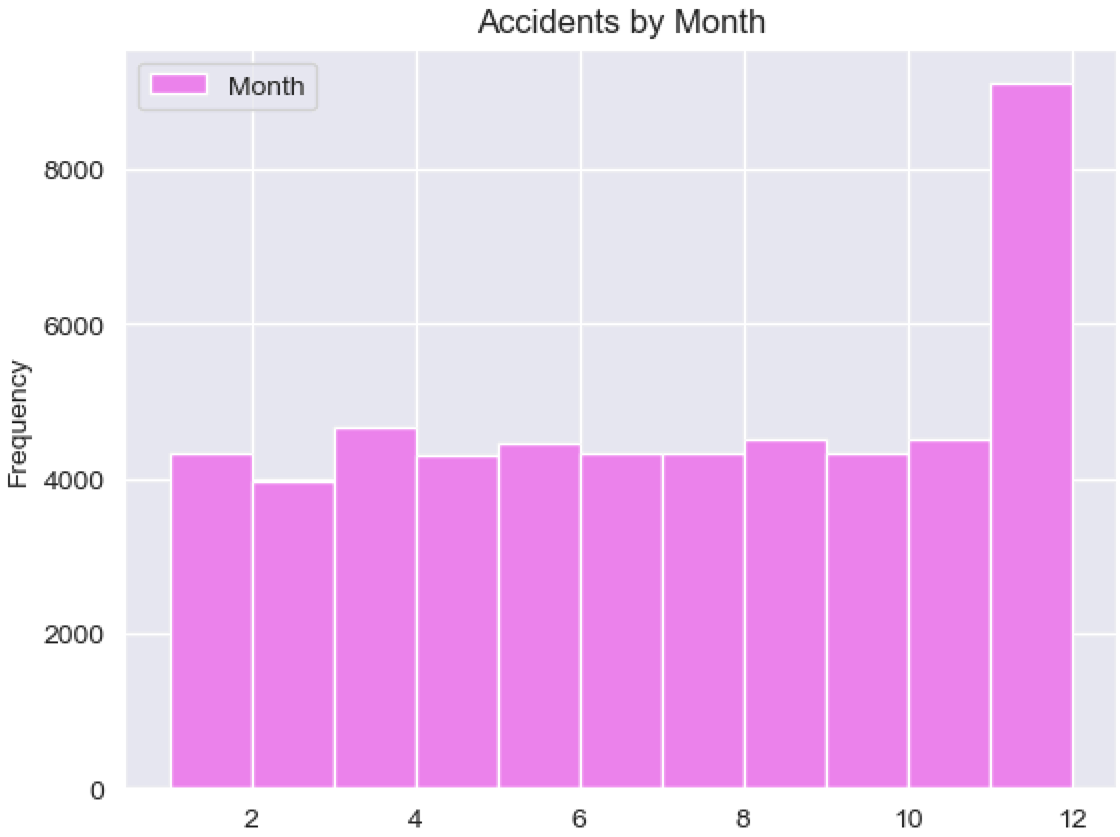


In [60]:
```
accidents_by_Month= df.groupby('Month').count()['Crash ID']
accidents_by_Month
```

Out[60]:
```
Month
1     4329
2     3975
3     4673
4     4298
5     4447
6     4333
7     4321
8     4512
9     4337
10    4509
11    4388
12    4721
Name: Crash ID, dtype: int64
```

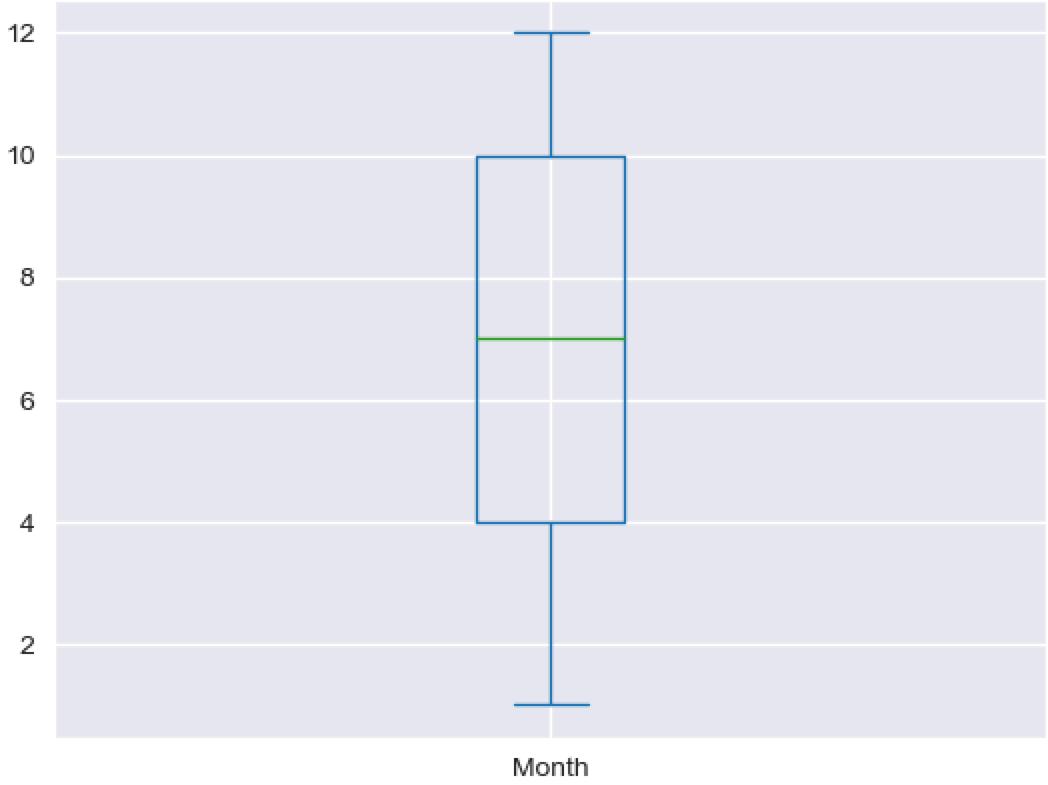
In [62]:
```
df_num.plot(kind='hist', y='Month', x='Crash ID', bins=11, color="violet",title="Accic
```

Out[62]:
```
<Axes: title={'center': 'Accidents by Month'}, ylabel='Frequency'>
```



```
In [64]: #Box Plot
df_num.plot(kind='box', y='Month', x='Crash ID')
```

```
Out[64]: <Axes: >
```



In [66]:

```
accidents_by_Year = df.groupby('Year').count()['Crash ID']
accidents_by_Year
```

Out[66]:

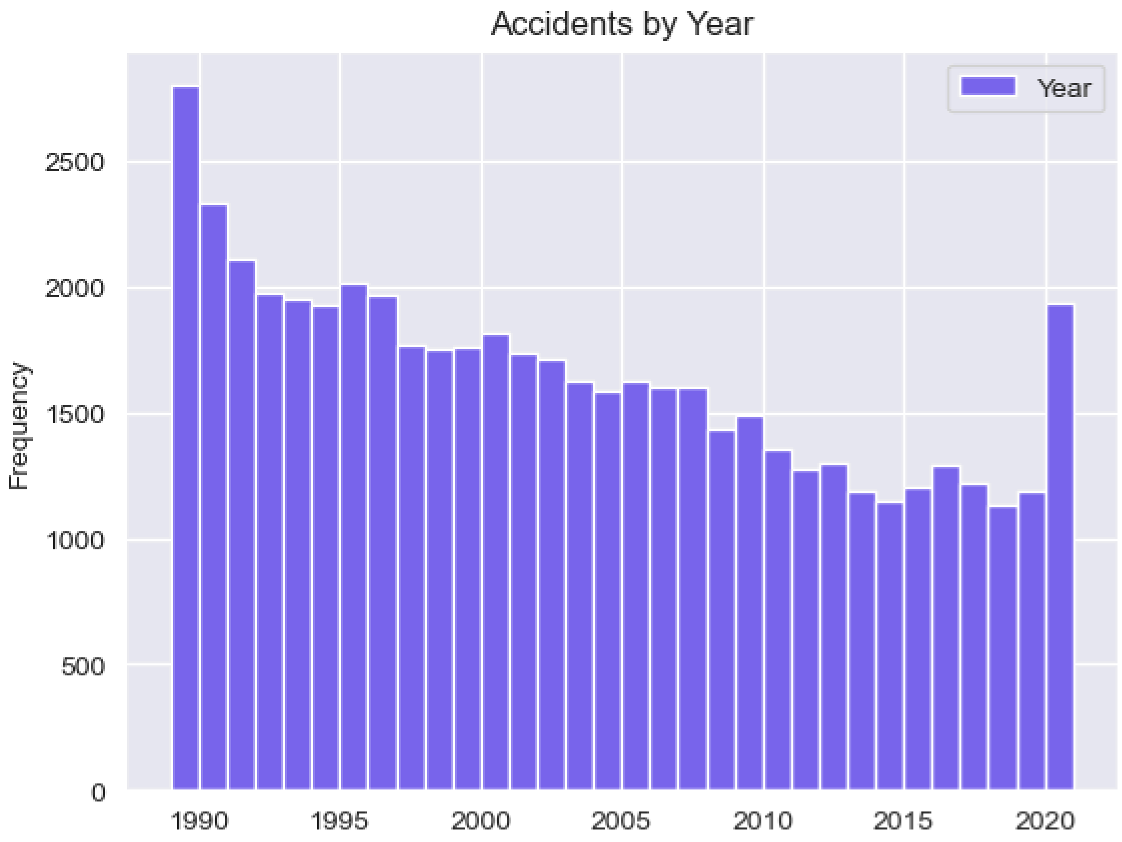
```
Year
1989    2800
1990    2331
1991    2113
1992    1974
1993    1953
1994    1928
1995    2017
1996    1970
1997    1767
1998    1755
1999    1764
2000    1817
2001    1737
2002    1715
2003    1621
2004    1583
2005    1627
2006    1598
2007    1603
2008    1437
2009    1491
2010    1353
2011    1277
2012    1300
2013    1187
2014    1151
2015    1204
2016    1292
2017    1222
2018    1134
2019    1186
2020    1093
2021     843
Name: Crash ID, dtype: int64
```

In [68]:

```
df_num.plot(kind='hist', y='Year', x='Crash ID', bins=32, color="mediumslateblue",titl
```

Out[68]:

```
<Axes: title={'center': 'Accidents by Year'}, ylabel='Frequency'>
```



In [70]:

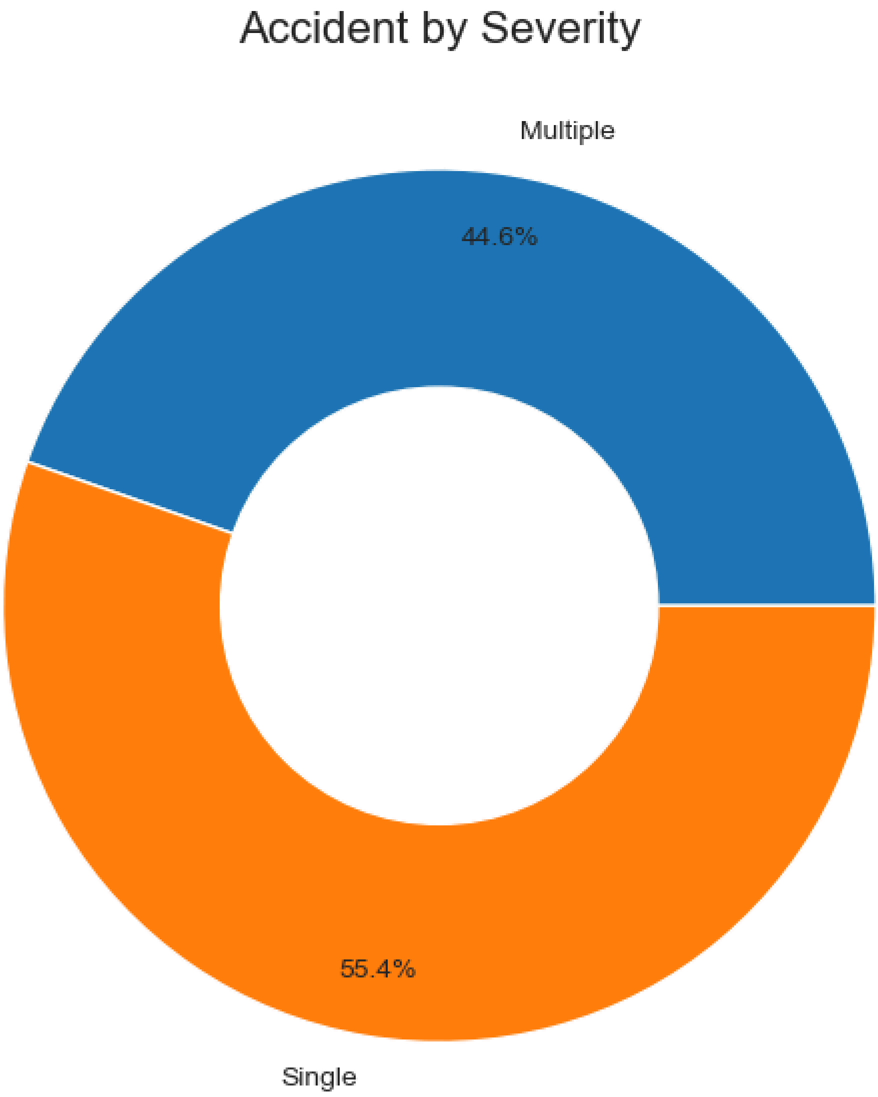
```
accidents_severity = df.groupby('Crash Type').count()['Crash ID']
accidents_severity
```

Out[70]:

```
Crash Type
Multiple    23594
Single      29249
Name: Crash ID, dtype: int64
```

In [72]:

```
fig, ax = plt.subplots(figsize=(7, 6), subplot_kw=dict(aspect="equal"))
label = ["Multiple","Single"]
plt.pie(accidents_severity,labels=label,autopct='%1.1f%%', pctdistance=0.85)
circle = plt.Circle( (0,0), 0.5, color='white')
p=plt.gcf()
p.gca().add_artist(circle)
ax.set_title("Accident by Severity",fontdict={'fontsize': 16})
plt.tight_layout()
plt.show()
```



In [74]:

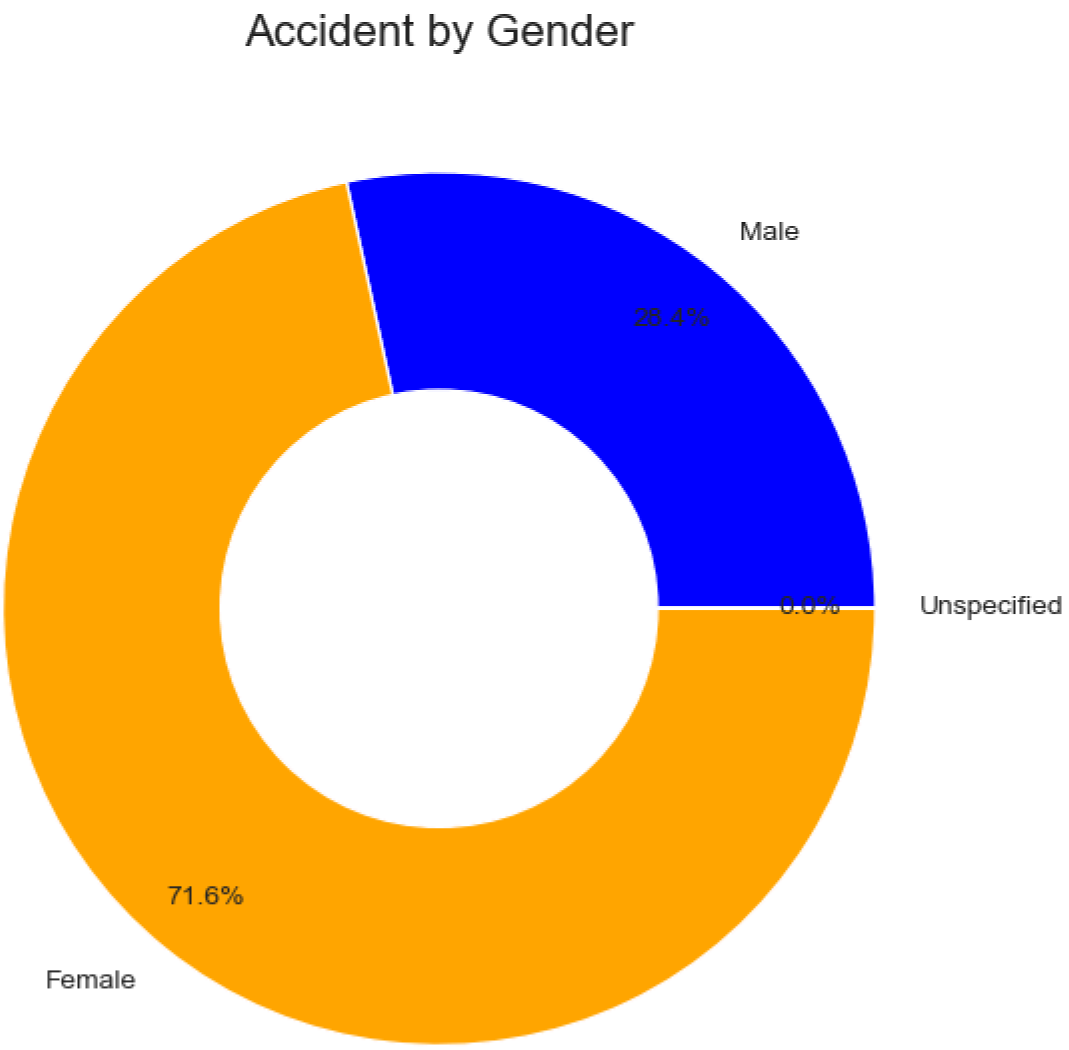
```
accidents_by_Gender = df.groupby('Gender').count()['Crash ID']
accidents_by_Gender
```

Out[74]:

```
Gender
Female         15002
Male           37813
Unspecified        1
Name: Crash ID, dtype: int64
```

In [78]:

```
fig, ax = plt.subplots(figsize=(8, 6), subplot_kw=dict(aspect="equal"))
label = ["Male","Female","Unspecified"]
colors=["blue","orange","green"]
plt.pie(accidents_by_Gender,labels=label,autopct='%1.1f%%', pctdistance=0.85, colors=c
circle = plt.Circle( (0,0), 0.5, color='white')
p=plt.gcf()
p.gca().add_artist(circle)
ax.set_title("Accident by Gender",fontdict={'fontsize': 16})
plt.tight_layout()
plt.show()
```

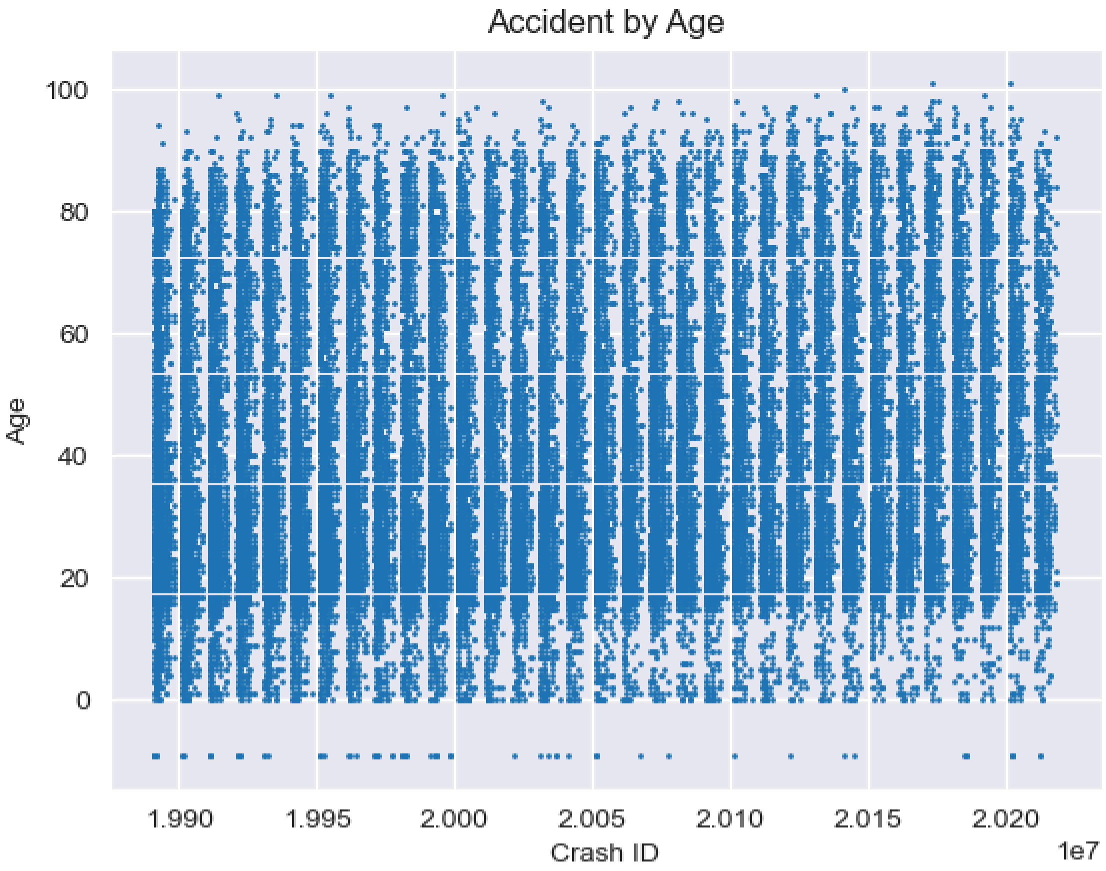


In [80]:

```
df_num.plot(kind='scatter', y='Age', x='Crash ID', s=1, title="Accident by Age")
```

Out[80]:

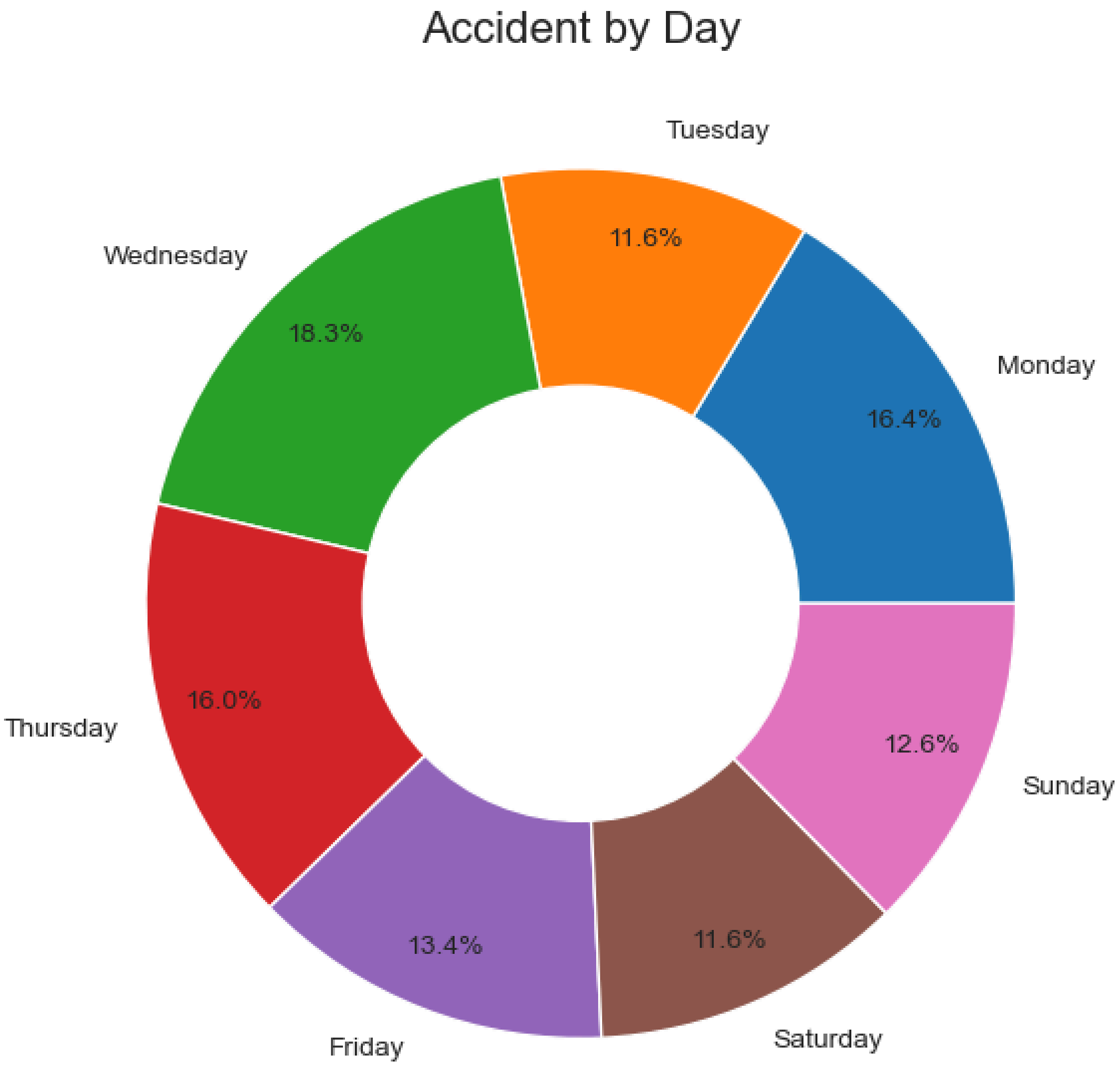
```
<Axes: title={'center': 'Accident by Age'}, xlabel='Crash ID', ylabel='Age'>
```



In [82]:
```
accidents_by_Day = df.groupby('Dayweek').count()['Crash ID']
accidents_by_Day
```

Out[82]:
```
Dayweek
Friday       8665
Monday       6108
Saturday     9696
Sunday       8460
Thursday     7106
Tuesday      6145
Wednesday    6663
Name: Crash ID, dtype: int64
```

In [84]:
```
fig, ax = plt.subplots(figsize=(8, 6), subplot_kw=dict(aspect="equal"))
label = ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
plt.pie(accidents_by_Day,labels=label,autopct='%1.1f%%', pctdistance=0.85)
circle = plt.Circle( (0,0), 0.5, color='white')
p=plt.gcf()
p.gca().add_artist(circle)
ax.set_title("Accident by Day",fontdict={'fontsize': 16})
plt.tight_layout()
plt.show()
```


Accident by Day
Tuesday
11.6%
Wednesday
18.3%
Monday
16.4%
Thursday
16.0%
12.6%
Sunday
13.4%
11.6%
Friday
Saturday

```
In [ ]:
```