task-5

August 3, 2024

[67]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[69]:
```
df_USA=pd.read_csv('accidents.csv')
```

[71]:
```
df_USA.head()
```

[71]:
```
  AccidentDate     Timing       State WeatherCondition       RoadCondition  \
0   04-02-2013    Morning      Alaska            Rainy  Under Construction
1   23-02-2005      Night     Arizona            Clear  Under Construction
2   08-10-2014  Afternoon  California            Clear                Fine
3   14-01-2015      Night    Colorado            Rainy               Rough
4   17-01-2006  Afternoon     Georgia            Clear                Fine

   Deaths              Reason
0      10       Drunk Driving
1       3  Weather Conditions
2       6     Poor Visibility
3       8     Road Conditions
4       2            Speeding
```

[72]:
```
df_USA.tail()
```

[72]:
```
      AccidentDate   Timing           State WeatherCondition  \
49995   20-08-2002    Night        Virginia            Clear
49996   15-05-2012    Night        Virginia            Clear
49997   19-05-2007  Evening  North Carolina            Rainy
49998   04-08-2019    Night  South Carolina            Clear
49999   25-04-2019  Evening         Georgia            Rainy

            RoadCondition  Deaths              Reason
49995  Under Construction       2  Mechanical Failure
49996  Under Construction       0  Mechanical Failure
49997  Under Construction       2      Driver Fatigue
49998                Fine       0  Distracted Driving
```

```
49999                Fine       2  Weather Conditions
```

[75]:
```
df_USA.columns
```

[75]:
```
Index(['AccidentDate', 'Timing', 'State', 'WeatherCondition', 'RoadCondition',
       'Deaths', 'Reason'],
      dtype='object')
```

[77]:
```
df_USA.dtypes.value_counts()
```

[77]:
```
object    6
int64     1
Name: count, dtype: int64
```

[78]:
```
df_USA.shape
```

[78]:
```
(50000, 7)
```

[79]:
```
df_USA.describe()
```

[79]:
```
             Deaths
count  50000.000000
mean       4.983040
std        3.160581
min        0.000000
25%        2.000000
50%        5.000000
75%        8.000000
max       10.000000
```

[80]:
```
df_USA.State.unique
```

[80]:
```
<bound method Series.unique of 0                Alaska
1                Arizona
2             California
3               Colorado
4                Georgia
              …
49995           Virginia
49996           Virginia
49997     North Carolina
49998     South Carolina
49999            Georgia
Name: State, Length: 50000, dtype: object>
```

[85]:
```
df1=df_USA[df_USA['State']=='Virginia']
```

[86]:
```
df1['IDD'] = df1['Timing'].astype('str').str.extractall('(\d+)').unstack().
↪fillna('').sum(axis=1).astype(int)
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\Dibyam Jyoti
Pradhan\AppData\Local\Temp\ipykernel_23116\3268597611.py:1: SyntaxWarning:
invalid escape sequence '\d'
  df1['IDD'] = df1['Timing'].astype('str').str.extractall('(\d+)').unstack().fil
lna('').sum(axis=1).astype(int)
C:\Users\Dibyam Jyoti
Pradhan\AppData\Local\Temp\ipykernel_23116\3268597611.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1['IDD'] = df1['Timing'].astype('str').str.extractall('(\d+)').unstack().fil
lna('').sum(axis=1).astype(int)
```

[89]:
```
df1
```

[89]:
```
      AccidentDate     Timing     State WeatherCondition       RoadCondition  \
22      29-11-2023  Afternoon  Virginia            Rainy  Under Construction
41      30-06-2023    Morning  Virginia            Foggy               Rough
60      14-05-2021    Morning  Virginia            Rainy                Fine
102     25-07-2007      Night  Virginia            Rainy  Under Construction
188     13-06-2012      Night  Virginia            Clear  Under Construction
…              …          …         …                …                   …
49925   29-01-2019      Night  Virginia            Rainy               Rough
49938   23-10-2002    Morning  Virginia            Foggy                Fine
49984   14-12-2021  Afternoon  Virginia            Foggy                Fine
49995   20-08-2002      Night  Virginia            Clear  Under Construction
49996   15-05-2012      Night  Virginia            Clear  Under Construction

       Deaths              Reason  IDD
22          6    Reckless Driving  NaN
41          9  Mechanical Failure  NaN
60          0     Road Conditions  NaN
102         2    Reckless Driving  NaN
188         1  Mechanical Failure  NaN
…         …                   …   …
49925       8  Distracted Driving  NaN
49938       2     Poor Visibility  NaN
49984       2    Reckless Driving  NaN
49995       2  Mechanical Failure  NaN
```

```
49996       0  Mechanical Failure  NaN

[1808 rows x 8 columns]
```

[90]:
```
df1.head()
```

[90]:
```
    AccidentDate     Timing     State WeatherCondition       RoadCondition  \
22    29-11-2023  Afternoon  Virginia            Rainy  Under Construction
41    30-06-2023    Morning  Virginia            Foggy               Rough
60    14-05-2021    Morning  Virginia            Rainy                Fine
102   25-07-2007      Night  Virginia            Rainy  Under Construction
188   13-06-2012      Night  Virginia            Clear  Under Construction

     Deaths             Reason  IDD
22        6    Reckless Driving  NaN
41        9  Mechanical Failure  NaN
60        0     Road Conditions  NaN
102       2    Reckless Driving  NaN
188       1  Mechanical Failure  NaN
```

[91]:
```
df1.tail()
```

[91]:
```
      AccidentDate     Timing     State WeatherCondition       RoadCondition  \
49925   29-01-2019      Night  Virginia            Rainy               Rough
49938   23-10-2002    Morning  Virginia            Foggy                Fine
49984   14-12-2021  Afternoon  Virginia            Foggy                Fine
49995   20-08-2002      Night  Virginia            Clear  Under Construction
49996   15-05-2012      Night  Virginia            Clear  Under Construction

       Deaths              Reason  IDD
49925       8  Distracted Driving  NaN
49938       2     Poor Visibility  NaN
49984       2    Reckless Driving  NaN
49995       2  Mechanical Failure  NaN
49996       0  Mechanical Failure  NaN
```

[95]:
```
df1.shape
```

[95]:
```
(1808, 8)
```

[96]:
```
df1.columns
```

[96]:
```
Index(['AccidentDate', 'Timing', 'State', 'WeatherCondition', 'RoadCondition',
       'Deaths', 'Reason', 'IDD'],
      dtype='object')
```

[97]:
```
d1f=df1.dropna(subset=['AccidentDate'])
```

```
[ ]:
```

```
[102]: f1=df1.dropna(subset=['AccidentDate', 'Timing', 'State', 'WeatherCondition',␣
         ↪'RoadCondition',
              'Deaths', 'Reason', 'IDD'])
```

```
[103]: df1.isna().sum()/len(df1)*100
```

```
[103]: AccidentDate          0.0
       Timing                0.0
       State                 0.0
       WeatherCondition      0.0
       RoadCondition         0.0
       Deaths                0.0
       Reason                0.0
       IDD                 100.0
       dtype: float64
```

```
[106]: df_cat=df1.select_dtypes('object')
       col_name=[]
       length=[]

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

```
[106]:             feature  count_of_unique_values
       0      AccidentDate                    1612
       1            Timing                       4
       2             State                       1
       3  WeatherCondition                       3
       4     RoadCondition                       3
       5            Reason                       9
```

```
[108]: df1.drop(['RoadCondition','Reason','Timing'],axis=1,inplace=True)
```

```
C:\Users\Dibyam Jyoti
Pradhan\AppData\Local\Temp\ipykernel_23116\1704517305.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1.drop(['RoadCondition','Reason','Timing'],axis=1,inplace=True)
```

```
[110]: del df1['AccidentDate']
```

```
[111]: df_num.columns
```

```
[111]: Index(['Deaths', 'IDD'], dtype='object')
```

```
[114]: len(df_num.columns)
```

```
[114]: 2
```

```
[115]: df_cat.columns
```

```
[115]: Index(['AccidentDate', 'Timing', 'State', 'WeatherCondition', 'RoadCondition',
              'Reason'],
             dtype='object')
```

```
[154]: len(df1['AccidentDate'].unique())
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.
  ↪get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtable.
  ↪PyObjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtable.
  ↪PyObjectHashTable.get_item()

KeyError: 'Acciden tDate'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[154], line 1
----> 1 len(df1['Acciden tDate'].unique())

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.
  ↪__getitem__(self, key)
   4100 if self.columns.nlevels > 1:
```

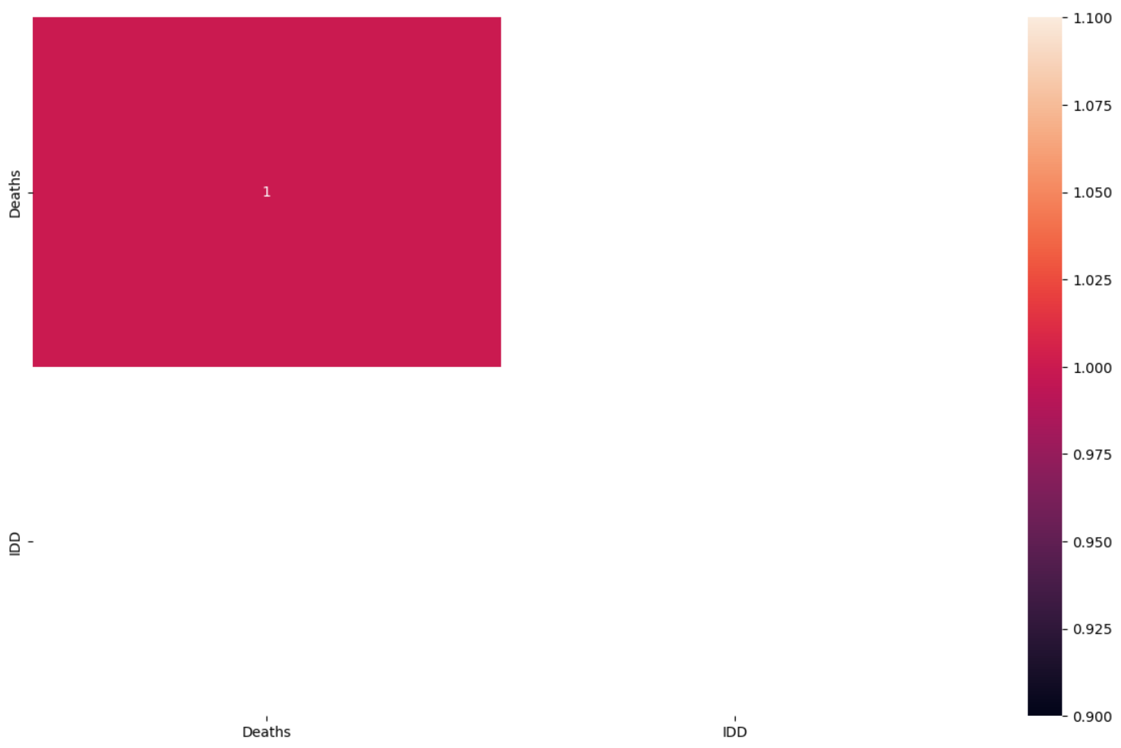
```
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'Acciden tDate'
```

[120]:
```
df_num=df1.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
```

[120]:
```
  feature  count_of_unique_values
0  Deaths                      11
1     IDD                       1
```

[122]:
```
plt.figure(figsize=(15 ,9))
sns.heatmap(df_num.corr() , annot=True)
```

[122]:
```
<Axes: >
```



[123]:
```
Date = df1['AccidentDate'].unique()
len(Date)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.
  ↪get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtable.
  ↪PyObjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtable.
  ↪PyObjectHashTable.get_item()

KeyError: 'AccidentDate'
```

```
The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[123], line 1
----> 1 Date = df1['AccidentDate'].unique()
      2 len(Date)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.
  ↪__getitem__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.
  ↪get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'AccidentDate'
```

[125]:
```
Data = df1['AccidentDate'].value_counts()
Data
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.
  ↪get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()
File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()

KeyError: 'AccidentDate'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[125], line 1
----> 1 Data = df1['AccidentDate'].value_counts()
      2 Data

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.
↪__getitem__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.
↪get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'AccidentDate'
```

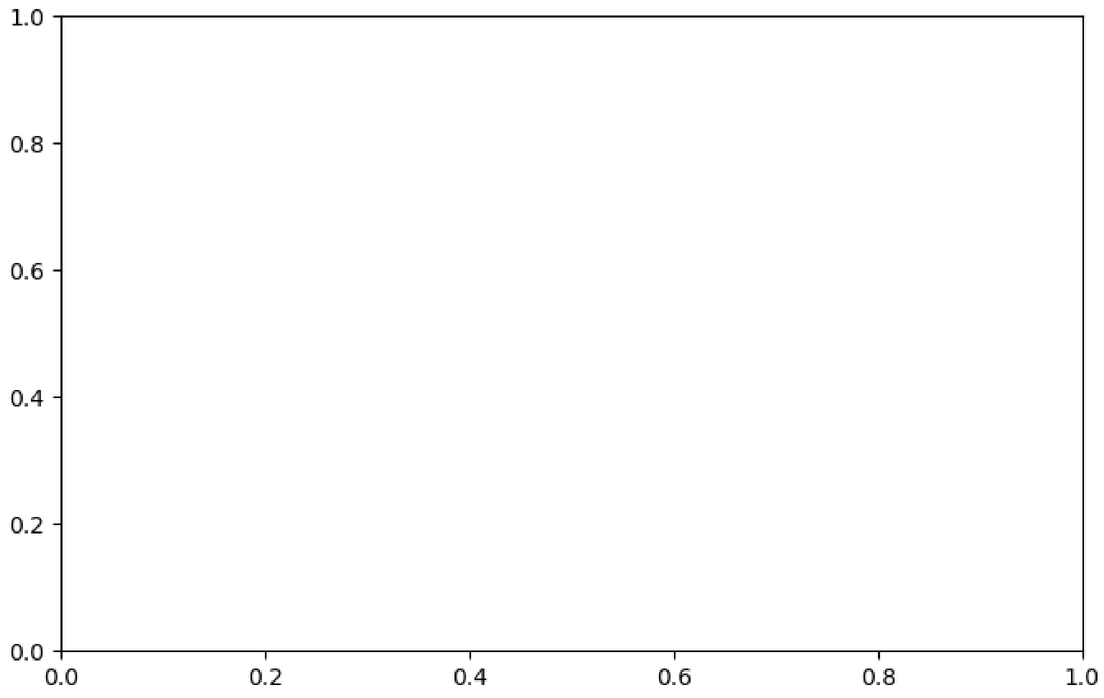
[127]:
```
Data[:10]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[127], line 1
----> 1 Data[:10]
```

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

[129]:
```
fig, ax = plt.subplots(figsize=(8,5))
Data[:10].plot(kind='bar')
ax.set(title = 'Top 10 Accidents Date',
       xlabel = 'AccidentDate',
       ylabel = 'Accidents Count')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[129], line 2
      1 fig, ax = plt.subplots(figsize=(8,5))
----> 2 Data[:10].plot(kind='bar')
      3 ax.set(title = 'Top 10 Accidents Date',
      4        xlabel = 'AccidentDate',
      5        ylabel = 'Accidents Count')
      6 plt.show()

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



[131]:
```
Accident_Date= df1.groupby('AccidentDate').count()['IDD']
Accident_Date
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[131], line 1
----> 1 Accident_Date= df1.groupby('AccidentDate').count()['IDD']
      2 Accident_Date

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:9183, in DataFrame.
 ↪groupby(self, by, axis, level, as_index, sort, group_keys, observed, dropna)
   9180 if level is None and by is None:
   9181     raise TypeError("You have to supply one of 'by' and 'level'")
-> 9183 return DataFrameGroupBy(
   9184     obj=self,
   9185     keys=by,
   9186     axis=axis,
   9187     level=level,
   9188     as_index=as_index,
   9189     sort=sort,
   9190     group_keys=group_keys,
   9191     observed=observed,
   9192     dropna=dropna,
   9193 )

File ~\anaconda3\Lib\site-packages\pandas\core\groupby\groupby.py:1329, in␣
 ↪GroupBy.__init__(self, obj, keys, axis, level, grouper, exclusions, selection ␣
 ↪as_index, sort, group_keys, observed, dropna)
   1326 self.dropna = dropna
   1328 if grouper is None:
-> 1329     grouper, exclusions, obj = get_grouper(
   1330         obj,
   1331         keys,
   1332         axis=axis,
   1333         level=level,
   1334         sort=sort,
   1335         observed=False if observed is lib.no_default else observed,
   1336         dropna=self.dropna,
   1337     )
   1339 if observed is lib.no_default:
   1340     if any(ping._passed_categorical for ping in grouper.groupings):

File ~\anaconda3\Lib\site-packages\pandas\core\groupby\grouper.py:1043, in␣
 ↪get_grouper(obj, key, axis, level, sort, observed, validate, dropna)
   1041         in_axis, level, gpr = False, gpr, None
   1042     else:
-> 1043         raise KeyError(gpr)
   1044 elif isinstance(gpr, Grouper) and gpr.key is not None:
   1045     # Add key to exclusions
   1046     exclusions.add(gpr.key)
```

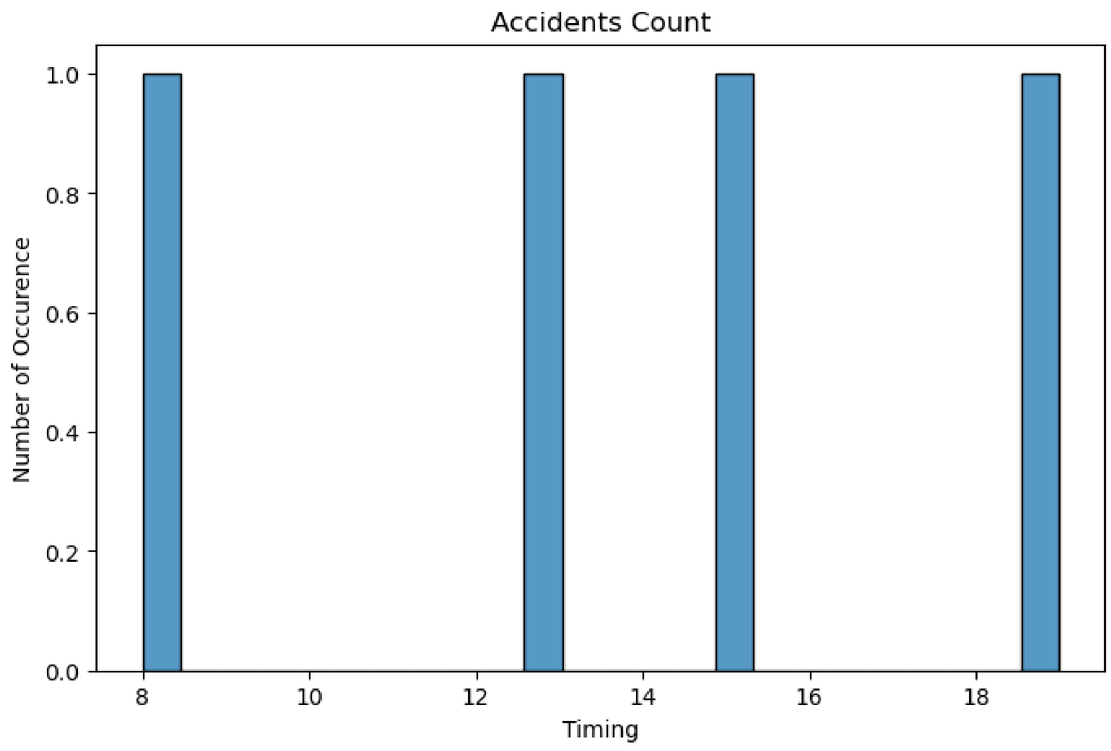
```
KeyError: 'AccidentDate'
```

[ ]:
```
fig, ax = plt.subplots(figsize=(8, 6), subplot_kw=dict(aspect="equal"))
labels = [10,20,30,40]
labels = ['Accident_Date 1', 'Accident_Date 2', 'Accident_Date 3',␣
 ↪'Accident_Date 4']
plt.pie(Accident_Date, labels=labels,
        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_Date",fontdict={'fontsize': 16})
plt.tight_layout()
plt.show()
```

[144]:
```
import pandas as pd
data = {
    'Timing': ['2024-06-25 13:00:00', 'Afternoon', '2024-06-26 08:30:00',␣
 ↪'Evening']
}
df1 = pd.DataFrame(data)
def convert_timing(value):
    if value == 'Afternoon':
        return '2024-06-25 15:00:00'
    elif value == 'Evening':
        return '2024-06-25 19:00:00'
    else:
        return value
df1['Timing'] = df1['Timing'].apply(convert_timing)
df1['Timing'] = pd.to_datetime(df1['Timing'])
print(df1['Timing'].dtypes)
print(df1)
```

```
datetime64[ns]
               Timing
0 2024-06-25 13:00:00
1 2024-06-25 15:00:00
2 2024-06-26 08:30:00
3 2024-06-25 19:00:00
```

[146]:
```
fig, ax = plt.subplots(figsize=(8,5))
sns.histplot(df1['Timing'].dt.hour, bins = 24)

plt.xlabel("Timing")
plt.ylabel("Number of Occurence")
plt.title('Accidents Count')
plt.show()
```



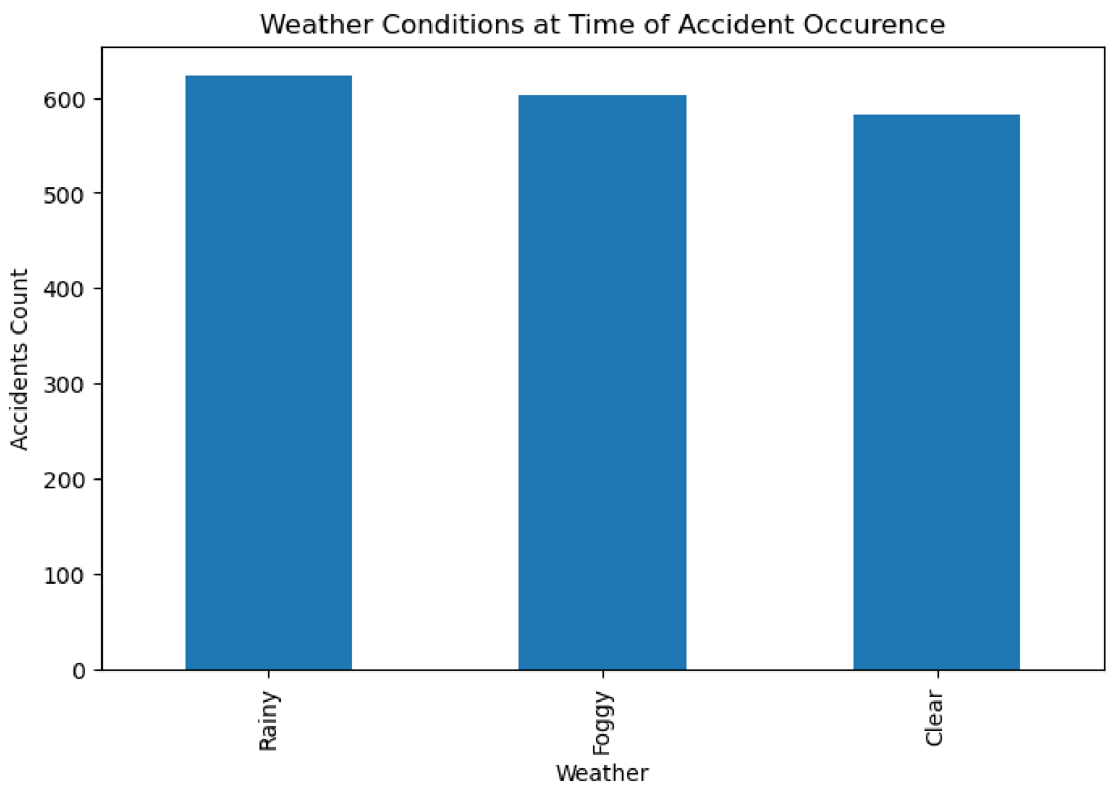
```
[148]: df1.columns
```

```
[148]: Index(['Timing'], dtype='object')
```

```
[152]: Weather_Condition =df1['WeatherCondition'].value_counts()
       Weather_Condition
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.
 ↪get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()
File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()

KeyError: 'WeatherCondition'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[152], line 1
----> 1 Weather_Condition =df1['WeatherCondition'].value_counts()
      2 Weather_Condition

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.
↪__getitem__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.
↪get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'WeatherCondition'
```

[141]:
```
fig, ax = plt.subplots(figsize=(8,5))
Weather_Condition.sort_values(ascending=False)[:20].plot(kind='bar')
ax.set(title = 'Weather Conditions at Time of Accident Occurence',
       xlabel = 'Weather',
       ylabel = 'Accidents Count')
plt.show()
```


Weather Conditions at Time of Accident Occurence
Accidents Count
0
100
200
300
400
500
600
Rainy
Foggy
Clear
Weather

```
[ ]:
```

```
[ ]:
```