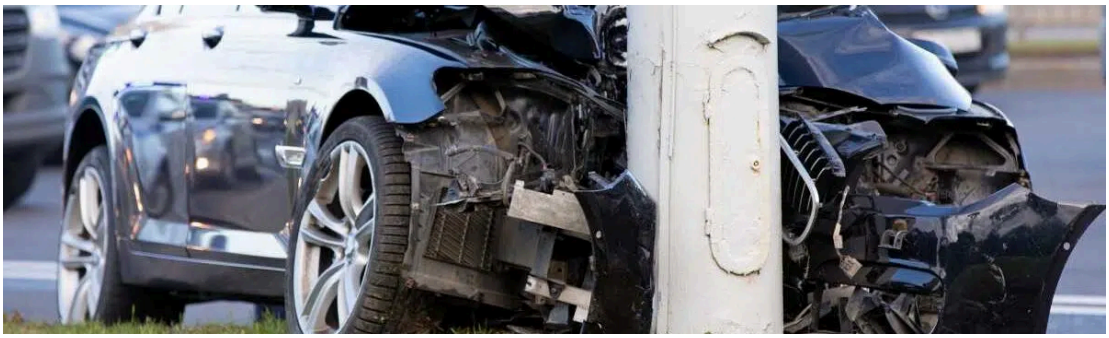


Setup and Context

Introduction

Road traffic accidents remain a major public safety challenge in Kenya, contributing to significant loss of life, injuries, and economic burden each year. The goal of this project is to explore, analyze, and derive insights from a real-world dataset of reported road accidents across the country. By examining patterns across time, geography, vehicle types, and accident characteristics, this project seeks to uncover the key factors associated with road incidents.

This analysis focuses on:

- Cleaning and preparing the raw accident data for accurate exploration.
- Identifying temporal trends such as high-risk hours, days, and months.
- Highlighting geographical hotspots and dangerous road segments.
- Examining accident characteristics including vehicle involvement, victim demographics, and contributing factors.
- Extracting meaningful patterns from text-based accident descriptions.
- Laying the foundation for future predictive modelling or dashboard development.

Ultimately, this project aims to provide clear, data-driven insights that can support safer road-use strategies, guide policy discussions, and demonstrate practical data science skills through exploratory analysis, visualization, and feature engineering.

Import Statements

In [3]:
```
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\alvo\anaconda3\lib\site-package
s (0.20.0)
Requirement already satisfied: branca>=0.6.0 in c:\users\alvo\anaconda3\lib\site-
packages (from folium) (0.8.2)
Requirement already satisfied: jinja2>=2.9 in c:\users\alvo\anaconda3\lib\site-pa
ckages (from folium) (3.1.4)
Requirement already satisfied: numpy in c:\users\alvo\anaconda3\lib\site-packages
(from folium) (1.26.4)
Requirement already satisfied: requests in c:\users\alvo\anaconda3\lib\site-packa
ges (from folium) (2.32.2)
Requirement already satisfied: xyzservices in c:\users\alvo\anaconda3\lib\site-pa
ckages (from folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\alvo\anaconda3\lib\sit
e-packages (from jinja2>=2.9->folium) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\alvo\anaconda
3\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\alvo\anaconda3\lib\site-p
ackages (from requests->folium) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\alvo\anaconda3\lib
\site-packages (from requests->folium) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\alvo\anaconda3\lib
\site-packages (from requests->folium) (2025.10.5)
```

In [125…

```
import numpy as np
import pandas as pd
import requests
import time

import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import folium
import calendar


import scipy.stats as stats
from datetime import datetime
```

Load the Data

In [7]:

```
data =  pd.read_csv('accidents-database-.csv')
```

Understand the accident dataset

Characteristics:

```
    :Number of Instances: 1,119

    :Number of Attributes: 13 numeric/categorical predictive.

    :Attribute Information (in order):
          1. TIME 24 HRS                The time of the accident
    recorded in 24-hour format.
          2. BASE/SUB BASE              The police base or sub-base
    station that handled or reported the accident.
```

```
      3. COUNTY                     The county where the
accident occurred.
      4. ROAD                       The specific road or highway
where the accident took place.
      5. PLACE                      The specific location or
landmark along the road where the accident happened.
      6. MV INVOLVED                The type(s) of motor
vehicle(s) involved in the accident.
      7. BRIEF ACCIDENT DETAILS     A short narrative describing
how the accident occurred.
      8. NAME OF VICTIM             The name of the individual
involved in the accident (if reported).
      9. GENDER                     Gender of the victim.
     10. AGE                        Age of the victim (sometimes
mixed with text and may require cleaning).
     11. CAUSE CODE                 Categorized reason or
official cause of the accident (coded format).
     12. VICTIM TYPE                Role of the victim in the
accident (e.g., pedestrian, passenger, driver).
     13. VICTIM NO.                 Identification number
assigned to the victim in the report (not a unique ID).
     14. DATE DD/MM/YYYY            The date the accident
occurred, in day/month/year format.


:Creator: Waweru Fidelis (Kaggle)
```

This is a copy of Kaggle accidents_kenya dataset. This dataset was taken from a Kaggle dataset. The dataset was posted on Kaggle in 2021 but the information recorded is from 2016. This dataset is the most comprehensively compiled data that could be obtained for this project seeing as most accident records in Kenya are still recorded manually on paper reports and there is no central database repository for such data.

Preliminary Data Cleaning and Preparation 🧹

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

Out[8]:

	TIME 24 HOURS	BASE/SUB BASE	COUNTY	ROAD	PLACE	MV INVOLVED	BRIEF ACCIDENT DETAILS	
0	630	KITUI	MAKUENI	KITUI-ITHOKWE	KITUI SCHOOL	KAV 479E & KMCZ 732L SKYGO	HEAD ON COLLISION	U
1	830	VOI	TAITA TAVETA	MOMBASA-NAIROBI	IKANGA	KCC 763/ZF 0097 MAKE RENAULT TRUCK & KMDN 759R...	HEAD ON COLLISION	U
2	1330	MARIAKANI	KILIFI	MOMBASA-NAIROBI	KATOLANI	KMDK 017U HAOJIN	THE UNKNOWN M/V HIT THE MOTOR CYCLE	U
3	2100	ONGATA RONGAI	NAKURU	NAKURU-NAIROBI	MAASAI LODGE	KAL 900K RANGE ROVER	THE VEHICLE KNOCKED DOWN A PEDESTRIAN WHO WAS ...	U
4	1900	MATUU	MACHAKOS	MATUU-MWINGI	KIVANDINI	KAQ 865Z TOYOTA COROLLA	THE VEHICLE OVERTOOK A M/CYCLE AND LOST CONTRO...	U

In [9]:
```
data.shape
```

Out[9]:
```
(1119, 15)
```

In [10]:
```
data.columns
```

Out[10]:
```
Index(['TIME 24 HOURS', 'BASE/SUB BASE', 'COUNTY', 'ROAD', 'PLACE',
       'MV INVOLVED', 'BRIEF ACCIDENT DETAILS', 'NAME OF VICTIM', 'GENDER',
       'AGE', 'CAUSE CODE', 'VICTIM', 'NO.', 'Date DD/MM/YYYY', 'Unnamed: 14'],
      dtype='object')
```

In [11]:
```
data.isna().values.any()
```

Out[11]:
```
True
```

Dropping and renaming columns

In [12]:
```
data.drop('Unnamed: 14', axis=1, inplace=True)
```

In [13]:
```
data.columns
```

Out[13]:
```
Index(['TIME 24 HOURS', 'BASE/SUB BASE', 'COUNTY', 'ROAD', 'PLACE',
       'MV INVOLVED', 'BRIEF ACCIDENT DETAILS', 'NAME OF VICTIM', 'GENDER',
       'AGE', 'CAUSE CODE', 'VICTIM', 'NO.', 'Date DD/MM/YYYY'],
      dtype='object')
```

In [14]:
```
data.dtypes
```

Out[14]:
```
TIME 24 HOURS             object
BASE/SUB BASE             object
COUNTY                    object
ROAD                      object
PLACE                     object
MV INVOLVED               object
BRIEF ACCIDENT DETAILS    object
NAME OF VICTIM            object
GENDER                    object
AGE                       object
CAUSE CODE                object
VICTIM                    object
NO.                       object
Date DD/MM/YYYY           object
dtype: object
```

In [15]:
```
data.rename(columns={'Date DD/MM/YYYY':'Date',
                     'TIME 24 HOURS':'Time',
                     'NO.': 'NUMBER OF VICTIMS'}, inplace=True)
```

In [16]:
```
data.dtypes
```

Out[16]:
```
Time                      object
BASE/SUB BASE             object
COUNTY                    object
ROAD                      object
PLACE                     object
MV INVOLVED               object
BRIEF ACCIDENT DETAILS    object
NAME OF VICTIM            object
GENDER                    object
AGE                       object
CAUSE CODE                object
VICTIM                    object
NUMBER OF VICTIMS         object
Date                      object
dtype: object
```

In [17]:
```
row_index = data[data['AGE'] == 'AGE'].index
print(row_index)
```

```
Index([378], dtype='int64')
```

In [18]:
```
data.iloc[row_index]
```

Out[18]:

	Time	BASE/SUB BASE	COUNTY	ROAD	PLACE	MV INVOLVED	BRIEF ACCIDENT DETAILS	NAME OF VICTIM	GEN
378	TIME 24 HOURS	BASE/SUB BASE	COUNTY	ROAD	PLACE	MV INVOLVED	BRIEF ACCIDENT DETAILS	NAME OF VICTIM	GEN

In [19]:
```
data.drop(row_index, inplace=True)
```

Changing date and time columns into datetime and time objects

In [20]:
```
def parse_date_with_year(x):
    try:
        # Try parsing normally (with year)
        return pd.to_datetime(x, infer_datetime_format=True)
    except:
        # If year is missing, append 2016
        return pd.to_datetime(f"{x}-2016", format="%d-%b-%Y")
```

In [21]:
```
data['Date'] = data['Date'].apply(parse_date_with_year)
```

```
C:\Users\Alvo\AppData\Local\Temp\ipykernel_13504\1319050334.py:4: UserWarning: Th
e argument 'infer_datetime_format' is deprecated and will be removed in a future
version. A strict version of it is now the default, see https://pandas.pydata.or
g/pdeps/0004-consistent-to-datetime-parsing.html. You can safely remove this argu
ment.
  return pd.to_datetime(x, infer_datetime_format=True)
```

In [22]:
```
data['Date'] = data['Date'].dt.strftime('%d-%m-%y')
data['Date']
```

Out[22]:
```
0       25-06-16
1       25-06-16
2       25-06-16
3       25-06-16
4       25-06-16
          ...
1114    22-02-16
1115    22-02-16
1116    22-02-16
1117    22-02-16
1118    22-02-16
Name: Date, Length: 1118, dtype: object
```

In [23]:
```
# Step 1: Convert all values to string, then pad numeric strings
def pad_time(x):
    x_str = str(x)  # Convert to string
    if x_str.isdigit():
        return x_str.zfill(4)  # Pad numeric strings
    else:
        return x_str  # Leave non-numeric as-is
```

In [24]:
```
data['Time'] = data['Time'].apply(pad_time)
```

In [25]:
```
data['Time'] = pd.to_datetime(data['Time'], format='%H%M', errors='coerce').dt.t
```

In [26]:
```
data['Date'] = pd.to_datetime(data['Date'])
```

```
C:\Users\Alvo\AppData\Local\Temp\ipykernel_13504\1453708669.py:1: UserWarning: Co
uld not infer format, so each element will be parsed individually, falling back t
o `dateutil`. To ensure parsing is consistent and as-expected, please specify a f
ormat.
  data['Date'] = pd.to_datetime(data['Date'])
```

In [27]:
```
data['Time']
```

Out[27]:
```
0       06:30:00
1       08:30:00
2       13:30:00
3       21:00:00
4       19:00:00
          ...
1114    17:30:00
1115    19:30:00
1116    21:45:00
1117    20:39:00
1118    01:39:00
Name: Time, Length: 1118, dtype: object
```

Data Cleaning - Check for Missing Values and Duplicates

In [28]:
```
data[data.isna().any(axis=1)]
```

Out[28]:

	Time	BASE/SUB BASE	COUNTY	ROAD	PLACE	MV INVOLVED	AC I
44	NaT	TARU	KWALE	MSA VOI ROAD	PIPELINE	UNKNOWN	H
121	15:30:00	KAYOLE	NAIROBI	KANGUNDO	KAMULU	KBJ 687Z ISUZU LORRY	KM DO
163	08:15:00	BOMET	BOMET	BOMET NAROK	MULOT	KBH 598W MITSUBISHI LORRY,KAY 311B MITSUBISHI ...	H CO
164	NaT	VOI	TAITA TAVETA	MWATATE VOI	CKAHARELI	KBZ 485A TOYOTA ISIS KMCG 028P TVS	H CO
165	NaT	VOI	TAITA TAVETA	MWATATE VOI	CKAHARELI	KBZ 485A TOYOTA ISIS KMCG 028P TVS	H CO
...	...	...	...	...	...	...	
1035	15:30:00	LAIKIPIA	LAIKIPIA	MATANYA ROAD	MUGETHO AREA	KAX 407B T/SALOON & KMDQ 204N M/CYCLE	VEH M
1038	22:15:00	KANGUNDO	MACHAKOS	NAIROBI-TALA ROAD	NEAR TALA HIGH SCHOOL	KBT 651 T/COROLLA & M/A/PED	KM D PEDI
1078	07:30:00	NaN	NaN	MWANGULU MWELENI ROAD	NEAR MWANAKALA JUNCTION	KBQ 830Z CANTER	THE OUT M
1080	20:30:00	MASENO	KISUMU	KISUMU BUSIA	LELA AREA	KMDR 774X BOXER,KMDQ 008N TVS & KBR 128T FAW P...	H CO
1081	20:30:00	MASENO	KISUMU	KISUMU BUSIA	LELA AREA	KMDR 774X BOXER,KMDQ 008N TVS & KBR 128T FAW P...	H CO

76 rows × 14 columns

In [29]:
```
na_counts = data.isna().sum().sort_values(ascending=False)
na_counts
```

Out[29]:
```
Time                     33
CAUSE CODE               28
PLACE                     6
NUMBER OF VICTIMS         4
COUNTY                    3
ROAD                      3
BASE/SUB BASE             2
BRIEF ACCIDENT DETAILS    1
NAME OF VICTIM            1
AGE                       1
MV INVOLVED               0
GENDER                    0
VICTIM                    0
Date                      0
dtype: int64
```

In [30]:
```
data.duplicated().values.any()
```

Out[30]:
```
True
```

In [31]:
```
duplicates = data[data.duplicated()]
print(duplicates)
```

```
         Time BASE/SUB BASE    COUNTY             ROAD  \
865  06:45:00      KAKAMEGA  KAKAMEGA    EKERO EBUHANGA
866  09:00:00     KENDU BAY  HOMA BAY  KENDU BAY-KATITO
867  21:00:00      NAROMORU     NYERI     NYERI-NANYUKI
868  02:20:00       NTULELE     NAROK    NAROK-MAIMAHIU
869  20:15:00        BURETI   KERICHO      BURET NGOINA

                         PLACE  \
865                     EBOKOLO
866                      BIAFRA
867  NATIONAL OIL PETROL STATION
868                    MALTAURO
869              NGOINA JUNCTION

                                    MV INVOLVED  \
865                                     UNKNOWN
866                                     UNKNOWN
867                           KBZ 710R T/AVENSIS
868                     KCD 078M/ZD 9607 SHACKMAN
869  KCE 051E T/PROBOX & MITSUBISHI PETROL TANKER

                              BRIEF ACCIDENT DETAILS  NAME OF VICTIM GENDER  \
865                                         HIT AND RAN  COLLINS OTIENO      M
866                                         HIT AND RAN         UNKNOWN      M
867               THE VEHICLE KNOCKED DOWN THE VICTIM         UNKNOWN      M
868   THE VEHICLE LOST CONTROL AND LANDED INTO A DITCH         UNKNOWN      F
869  THE PROBOX WAS OVERTAKING UNKNOWN M/V HENCE CO...         UNKNOWN      M

    AGE CAUSE CODE                VICTIM NUMBER OF VICTIMS       Date
865  26         98             PEDESTRIAN                 1 2016-06-16
866   A         98             PEDESTRIAN                 1 2016-06-16
867   A         50          PEDAL CYCLIST                 1 2016-06-16
868   A         78              PASSENGER                 1 2016-06-16
869   A         10  7 PASSENGERS 1 DRIVER                 8 2016-06-16
```

In [32]:
```
subset_cols = [col for col in data.columns if col != 'VICTIM']
```

In [33]:
```
mask = data.duplicated(subset=subset_cols, keep=False)
```

In [34]:
```
duplicates = data[mask]
```

In [35]:
```
data = data.drop_duplicates(subset=subset_cols, keep='first').reset_index(drop=T
data.head()
```

Out[35]:

	Time	BASE/SUB BASE	COUNTY	ROAD	PLACE	MV INVOLVED	BRIEF ACCIDENT DETAILS	
0	06:30:00	KITUI	MAKUENI	KITUI-ITHOKWE	KITUI SCHOOL	KAV 479E & KMCZ 732L SKYGO	HEAD ON COLLISION	U
1	08:30:00	VOI	TAITA TAVETA	MOMBASA-NAIROBI	IKANGA	KCC 763/ZF 0097 MAKE RENAULT TRUCK & KMDN 759R...	HEAD ON COLLISION	U
2	13:30:00	MARIAKANI	KILIFI	MOMBASA-NAIROBI	KATOLANI	KMDK 017U HAOJIN	THE UNKNOWN M/V HIT THE MOTOR CYCLE	U
3	21:00:00	ONGATA RONGAI	NAKURU	NAKURU-NAIROBI	MAASAI LODGE	KAL 900K RANGE ROVER	THE VEHICLE KNOCKED DOWN A PEDESTRIAN WHO WAS ...	U
4	19:00:00	MATUU	MACHAKOS	MATUU-MWINGI	KIVANDINI	KAQ 865Z TOYOTA COROLLA	THE VEHICLE OVERTOOK A M/CYCLE AND LOST CONTRO...	U

In [36]:
```
data.duplicated().values.any()
```

Out[36]: False

In [37]:
```
mapping = { 'HOMABAY' : 'HOMA BAY',
            'HOMA-BAY' : 'HOMA BAY',
            'KERCHO' : 'KERICHO',
            'KILFI' : 'KILIFI',
            'MAKURU' : 'NAKURU',
            'MURANGA' : "MURANG'A",
            'TAITA TAVETA' : 'TAITA-TAVETA',
            'UASIN NGISHU' : 'UASIN GISHU'
          }
```

In [38]:
```
data['COUNTY'] = data['COUNTY'].replace(mapping)
```

In [39]:
```
# Replace 'Tigania' with 'Meru'
data['COUNTY'] = data['COUNTY'].replace('TIGANIA', 'MERU')
```

In [40]:
```
data.drop(data[data['COUNTY'] == 'COUNTY'].index, axis=0, inplace=True)
```

In [41]:
```
data.COUNTY.unique()
```

Out[41]:
```
array(['MAKUENI', 'TAITA-TAVETA', 'KILIFI', 'NAKURU', 'MACHAKOS',
       'HOMA BAY', 'KIRINYAGA', 'NAIROBI', 'KIAMBU', 'MOMBASA', 'MERU',
       'MIGORI', 'KISII', 'SIAYA', 'BUNGOMA', 'KWALE', 'NANDI',
       'TRANS NZOIA', 'KAJIADO', 'UASIN GISHU', 'NAROK', 'KERICHO',
       'KAKAMEGA', 'NYERI', 'EMBU', 'KISUMU', "MURANG'A", 'BUSIA',
       'VIHIGA', 'GARISSA', 'BARINGO', 'LAIKIPIA', 'MWINGI', 'BOMET',
       'NYAMIRA', 'ELGEYO MARAKWET', 'NYAHURURU', 'KITUI', nan,
       'MARAKWET', 'NYANDARUA', 'WEST POKOT', 'THARAKA NITHI', 'MALINDI',
       'MANDERA', 'MARSABIT', 'TRAN NZOIA', 'ISIOLO', 'MOYALE'],
      dtype=object)
```

Changing Age, Number of victims and Cause code columns into integer/numeric objects

In [42]:
```
data['AGE'] = pd.to_numeric(data['AGE'], errors='coerce')
```

In [43]:
```
data.dtypes
```

Out[43]:
```
Time                          object
BASE/SUB BASE                 object
COUNTY                        object
ROAD                          object
PLACE                         object
MV INVOLVED                   object
BRIEF ACCIDENT DETAILS        object
NAME OF VICTIM                object
GENDER                        object
AGE                          float64
CAUSE CODE                    object
VICTIM                        object
NUMBER OF VICTIMS             object
Date                  datetime64[ns]
dtype: object
```

In [44]:
```
data['CAUSE CODE'] = pd.to_numeric(data['CAUSE CODE'], errors='coerce').astype('
```

In [45]:
```
data.dtypes
```

Out[45]:
```
Time                          object
BASE/SUB BASE                 object
COUNTY                        object
ROAD                          object
PLACE                         object
MV INVOLVED                   object
BRIEF ACCIDENT DETAILS        object
NAME OF VICTIM                object
GENDER                        object
AGE                          float64
CAUSE CODE                     Int64
VICTIM                        object
NUMBER OF VICTIMS             object
Date                  datetime64[ns]
dtype: object
```

In [46]:
```
data['NUMBER OF VICTIMS'] = pd.to_numeric(data['NUMBER OF VICTIMS'], errors='coe
```

In [47]:
```
data.dtypes
```

Out[47]:
```
Time                          object
BASE/SUB BASE                 object
COUNTY                        object
ROAD                          object
PLACE                         object
MV INVOLVED                   object
BRIEF ACCIDENT DETAILS        object
NAME OF VICTIM                object
GENDER                        object
AGE                          float64
CAUSE CODE                     Int64
VICTIM                        object
NUMBER OF VICTIMS              Int64
Date                  datetime64[ns]
dtype: object
```

In [48]:
```
data['MV INVOLVED'] = data['MV INVOLVED'].replace('UNKNWON', 'UNKNOWN')
```

In [49]:
```
data['MV INVOLVED'] = data['MV INVOLVED'].replace('UNKNOWN M/V (HIT AND RUN)', '
```

In [50]:
```
data['MV INVOLVED'] = data['MV INVOLVED'].replace('UNKNOWN M/V ( HIT AND RUN)',
```

Standardizing the gender column

This is done in order to standardize the values entered into the column as one of three options: Male, Female or both

In [51]:
```
gender_mapping = {
    'M': 'M',
    'F': 'F',
    'M&F': 'Both',
    'M & F': 'Both',
    'M AND F': 'Both',
    'M&f': 'Both',
    'M/F': 'Both',
    'M & F/J': 'Both',
```

```
    'M/J': 'Both',
    'F/J': 'Both',
    '2M & 3F': 'Both',
    '2F & M': 'Both',
    '2M & 2F': 'Both',
    '2M & F': 'Both',
    '2M, F': 'Both',
    '4F & INF': 'Both',
    'M/A': 'Unknown',
    'J': 'Unknown',
    '1': 'Unknown'
}
```

In [52]:
```
data['GENDER'] = data['GENDER'].replace(gender_mapping)
```

In [53]:
```
data['GENDER'] = data['GENDER'].str.strip().str.upper()
```

In [54]:
```
data['GENDER'].value_counts()
```

Out[54]:
```
GENDER
M          915
F          141
BOTH        28
UNKNOWN      7
Name: count, dtype: int64
```

Merging the cause codes dataset with the main dataset

This will help us determine the accurate description of the accident from the corresponding accident cause code

In [55]:
```
# Load the cuase codes dataset
df_codes = pd.read_csv('cause codes.csv')
```

In [56]:
```
df_codes.head()
```

Out[56]:

	Code	Description
0	Driver	NaN
1	1	Fatigued
2	2	Asleep
3	3	Ill
4	4	Under the influence of drink or a drug

In [57]:
```
# Identify group headers: Code is non-numeric
df_codes["Category"] = df_codes["Code"].where(
    ~df_codes["Code"].astype(str).str.isdigit()
)
```

In [58]:
```
# Forward-fill category names downward
df_codes["Category"] = df_codes["Category"].ffill()
```

In [59]:
```
# Keep only rows where Code is numeric
df_codes_clean = df_codes[df_codes["Code"].astype(str).str.isdigit()].copy()
```

In [60]:
```
# Convert Code to integer
df_codes_clean["Code"] = df_codes_clean["Code"].astype(int)
```

In [61]:
```
df_codes_clean.head()
```

Out[61]:

	Code	Description	Category
1	1	Fatigued	Driver
2	2	Asleep	Driver
3	3	Ill	Driver
4	4	Under the influence of drink or a drug	Driver
5	5	Physically defective	Driver

In [62]:
```
data = data.merge(
    df_codes_clean,
    how="left",
    left_on="CAUSE CODE",
    right_on="Code"
)
```

In [63]:
```
data.drop('Code', axis=1, inplace=True)
```

In [64]:
```
data.head()
```

Out[64]:

	Time	BASE/SUB BASE	COUNTY	ROAD	PLACE	MV INVOLVED	BRIEF ACCIDENT DETAILS	
0	06:30:00	KITUI	MAKUENI	KITUI-ITHOKWE	KITUI SCHOOL	KAV 479E & KMCZ 732L SKYGO	HEAD ON COLLISION	U
1	08:30:00	VOI	TAITA-TAVETA	MOMBASA-NAIROBI	IKANGA	KCC 763/ZF 0097 MAKE RENAULT TRUCK & KMDN 759R...	HEAD ON COLLISION	U
2	13:30:00	MARIAKANI	KILIFI	MOMBASA-NAIROBI	KATOLANI	KMDK 017U HAOJIN	THE UNKNOWN M/V HIT THE MOTOR CYCLE	U
3	21:00:00	ONGATA RONGAI	NAKURU	NAKURU-NAIROBI	MAASAI LODGE	KAL 900K RANGE ROVER	THE VEHICLE KNOCKED DOWN A PEDESTRIAN WHO WAS ...	U
4	19:00:00	MATUU	MACHAKOS	MATUU-MWINGI	KIVANDINI	KAQ 865Z TOYOTA COROLLA	THE VEHICLE OVERTOOK A M/CYCLE AND LOST CONTRO...	U

In [65]:
```
data.columns = (
    data.columns.str.strip()
                .str.lower()
                .str.replace(' ', '_')
                .str.replace('-', '_')
)
data.columns
```

Out[65]:
```
Index(['time', 'base/sub_base', 'county', 'road', 'place', 'mv_involved',
       'brief_accident_details', 'name_of_victim', 'gender', 'age',
       'cause_code', 'victim', 'number_of_victims', 'date', 'description',
       'category'],
      dtype='object')
```

In [66]:
```
# save reshaped data to CSV
data.to_csv('kenyan-accidents-reshaped.csv')
```

In [143…
```
data.columns
```

Out[143…
```
Index(['time', 'base/sub_base', 'county', 'road', 'place', 'mv_involved',
       'brief_accident_details', 'name_of_victim', 'gender', 'age',
       'cause_code', 'victim', 'number_of_victims', 'date', 'description',
       'category', 'month'],
      dtype='object')
```

Exploratory Data Analysis (EDA) 🔎

Descriptive Statistics

In [67]:
```
data.describe()
```

Out[67]:

	age	cause_code	number_of_victims	date
count	509.000000	1065.0	1087.0	1091
mean	34.116896	47.78216	1.179393	2016-08-13 08:22:52.685609728
min	1.500000	1.0	1.0	2016-01-03 00:00:00
25%	25.000000	20.0	1.0	2016-04-16 00:00:00
50%	32.000000	37.0	1.0	2016-06-20 00:00:00
75%	42.000000	76.0	1.0	2016-10-03 00:00:00
max	90.000000	99.0	18.0	2017-12-10 00:00:00
std	15.900484	33.248374	0.955751	NaN

In [68]:
```
data.describe(percentiles=[.25, .5, .75])
```

Out[68]:

	age	cause_code	number_of_victims	date
count	509.000000	1065.0	1087.0	1091
mean	34.116896	47.78216	1.179393	2016-08-13 08:22:52.685609728
min	1.500000	1.0	1.0	2016-01-03 00:00:00
25%	25.000000	20.0	1.0	2016-04-16 00:00:00
50%	32.000000	37.0	1.0	2016-06-20 00:00:00
75%	42.000000	76.0	1.0	2016-10-03 00:00:00
max	90.000000	99.0	18.0	2017-12-10 00:00:00
std	15.900484	33.248374	0.955751	NaN

In [69]:
```
data.describe(include='object')
```

Out[69]:

	time	base/sub_base	county	road	place	mv_involved	brief_acc
count	1059	1089	1088	1088	1086	1091	
unique	185	212	48	714	1022	967	
top	20:30:00	NAKURU	NAIROBI	NAIROBI MOMBASA	NEAR KANGEMI STAGE	UNKNOWN	KNOCKE
freq	43	31	183	21	3	53	

From the descriptive statistics of the numerical columns we can see that only the age column is of relevance to our analysis. The average age of accident victims is **34** years with the minimum age being **1.5** and the maximum age being **90**. In both cases these values are outliers. The standard deviation of ages is **34.12** years which is relatively high. There are only **509** entries in the age column meaning that more than half of the entries have null entries. This will affect the analysis of age further in the project. The records with null age entries cannot be dropped as they still have valuable data hence the records were retained.

Univariate Analysis

In [70]:
```
data['base/sub_base'].value_counts().head()
```

Out[70]:
```
base/sub_base
NAKURU         31
EMBAKASI       20
ATHI RIVER     20
STAREHE        19
KIKUYU         18
Name: count, dtype: int64
```

The police station/base with the most reported number of accidents is **Nakuru** sub base with 31 accidents recorded.

In [71]:
```
data['county'].value_counts().head()
```

Out[71]:
```
county
NAIROBI     183
KIAMBU       98
NAKURU       77
MAKUENI      57
MACHAKOS     50
Name: count, dtype: int64
```

The county with the most reported number of accidents is Nairobi county. This number can be attributed to the high number of motorists in the capital city seeing as Nairobi county has the largest population of any county in the country with an estimated population of **4,434,756**.

In [72]:
```
data['road'].value_counts().head()
```

Out[72]:
```
road
NAIROBI MOMBASA       21
MOMBASA NAIROBI       20
WAIYAKI WAY           17
THIKA SUPERHIGHWAY    16
MOMBASA ROAD          15
Name: count, dtype: int64
```

From the descriptive statistics it is determined that the Nairobi-Mombasa highway has the highest recorded number of accidents at 21 with Mombasa-Nairobi following at a close 20. These two entries refer to the same road corridor, but they differ based on the direction of movement:

- Nairobi → Mombasa (outbound direction)
- Mombasa → Nairobi (return direction)

Even though the physical roadway is identical, the dataset treats directionality as separate values because crashes, traffic flow, or incidents may vary depending on which direction vehicles were traveling.

In [73]:
```
data['county'].describe()
```

Out[73]:
```
count        1088
unique         48
top       NAIROBI
freq          183
Name: county, dtype: object
```

In [74]:
```
data.date.min()
```

Out[74]:
```
Timestamp('2016-01-03 00:00:00')
```

In [75]:
```
data.date.max()
```

Out[75]:
```
Timestamp('2017-12-10 00:00:00')
```

In [76]:
```
data['date'].dt.year.value_counts()
```

Out[76]:
```
date
2016    937
2017    154
Name: count, dtype: int64
```

From an analysis of the date column we determine that from the dataset the earliest recorded accident was on 3rd January 2016 and the latest recorded accident was on 10th December 2016. This means that the dataset record spans across almost two years falling short only by a month. The dataset shows a significant decrease in the number of recorded cases between 2016 and 2017. In 2016, there were 937 cases, while in 2017, the number dropped sharply to 154 cases. This represents an 84% reduction in cases. This reduction isn't practical in terms of accident reduction hence we can infer that there was an underpresentation of data for the year 2017. This can come from a lack of reporting cases or maybe poor data collection practices.

In [77]:
```
data['date'].dt.month.value_counts()
```

Out[77]:
```
date
4     160
6     146
7     140
3     130
10     89
8      89
9      82
5      82
11     49
2      47
12     42
1      35
Name: count, dtype: int64
```

The highest number of accidents occurred in April (160), followed by June (146), July (140), and March (130). The lowest counts were in November (49), February (47), December (42), and January (35). There is a clear seasonal pattern, with more accidents occurring in the second quarter and mid-year months (March–July), which may correspond to factors such as weather conditions, increased travel, or road usage patterns. The start and end of the year (January, February, December) tend to have fewer recorded accidents.

In [78]:
```
data['cause_code'].value_counts().head()
```

Out[78]:
```
cause_code
98    218
26    159
10     93
7      68
29     59
Name: count, dtype: Int64
```

The most frequent cause code is 98 with 218 occurrences, followed by 26 (159), 10 (93), 7 (68), and 29 (59). This suggests that a small number of cause codes account for the majority of incidents, with code 98 being especially dominant.

In [79]:
```
data['description'].value_counts().head()
```

Out[79]:
```
description
Cause not traced                                                218
Losing control (particulars to be specified)                    170
Overtaking improperly                                           109
Proceeding at excessive speed having regard to conditions        78
Failing to keep to near side or to the proper traffic lane       71
Name: count, dtype: int64
```

A significant portion of cases (218) have causes that were not traced, which limits actionable insights. Driver behavior-related issues (losing control, improper overtaking, speeding, lane discipline) are prominent among traced incidents, highlighting driver error as a key factor in incidents. The descending frequency indicates that a small number of behaviors account for most traffic incidents.

In [80]:
```
data['category'].value_counts()
```

Out[80]:
```
category
Driver            512
Other Cause       219
Pedestrian        163
Cyclist            93
Vehicle Defect     40
Passengers         28
Road Defect         5
Animal              4
Name: count, dtype: int64
```

Driver-related incidents dominate, making up more than 50% of all recorded incidents, reinforcing the trend seen in the description column. Pedestrian, cyclist, and vehicle defect causes are relatively less frequent, indicating that human driver error is the primary contributor. "Other Cause" remains substantial at 219, suggesting there are additional factors not captured by standard categories.

In [81]:
```
data['place'].value_counts().head()
```

Out[81]:
```
place
NEAR KANGEMI STAGE           3
STATE HOUSE                  3
NEAR SHELL PETROL STATION    3
GICHIEGO                     2
KINGEERO AREA                2
Name: count, dtype: int64
```

In [82]:
```
data['gender'].value_counts()
```

Out[82]:
```
gender
M          915
F          141
BOTH        28
UNKNOWN      7
Name: count, dtype: int64
```

In [83]:
```
data['gender'].value_counts(normalize=True) * 100
```

Out[83]:
```
gender
M          83.868011
F          12.923923
BOTH        2.566453
UNKNOWN     0.641613
Name: proportion, dtype: float64
```

The dataset shows that the majority of recorded cases involve males, with **915** occurrences which represents **83.9%** of victims, accounting for the bulk of the data. Females account for **141** cases which represents **12.9%** of accident victims, a much smaller proportion compared to males. Both genders involved are recorded in **28** cases, indicating incidents with multiple parties of different genders. Unknown gender appears in **7** cases, representing missing or ambiguous entries.

In [84]:
```
# Select only numeric columns
numeric_data = data.select_dtypes(include='number')
```

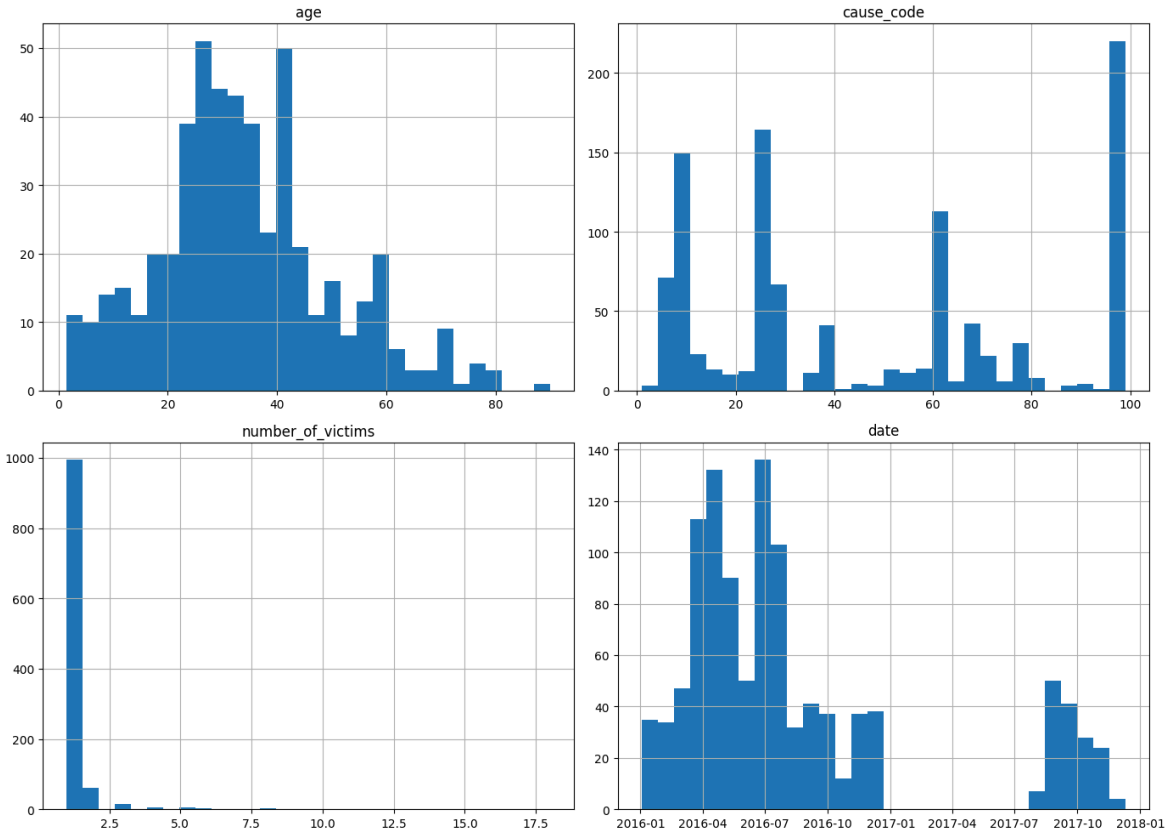
```
# Compute correlation matrix
corr_matrix = numeric_data.corr()

corr_matrix
```

Out[84]:

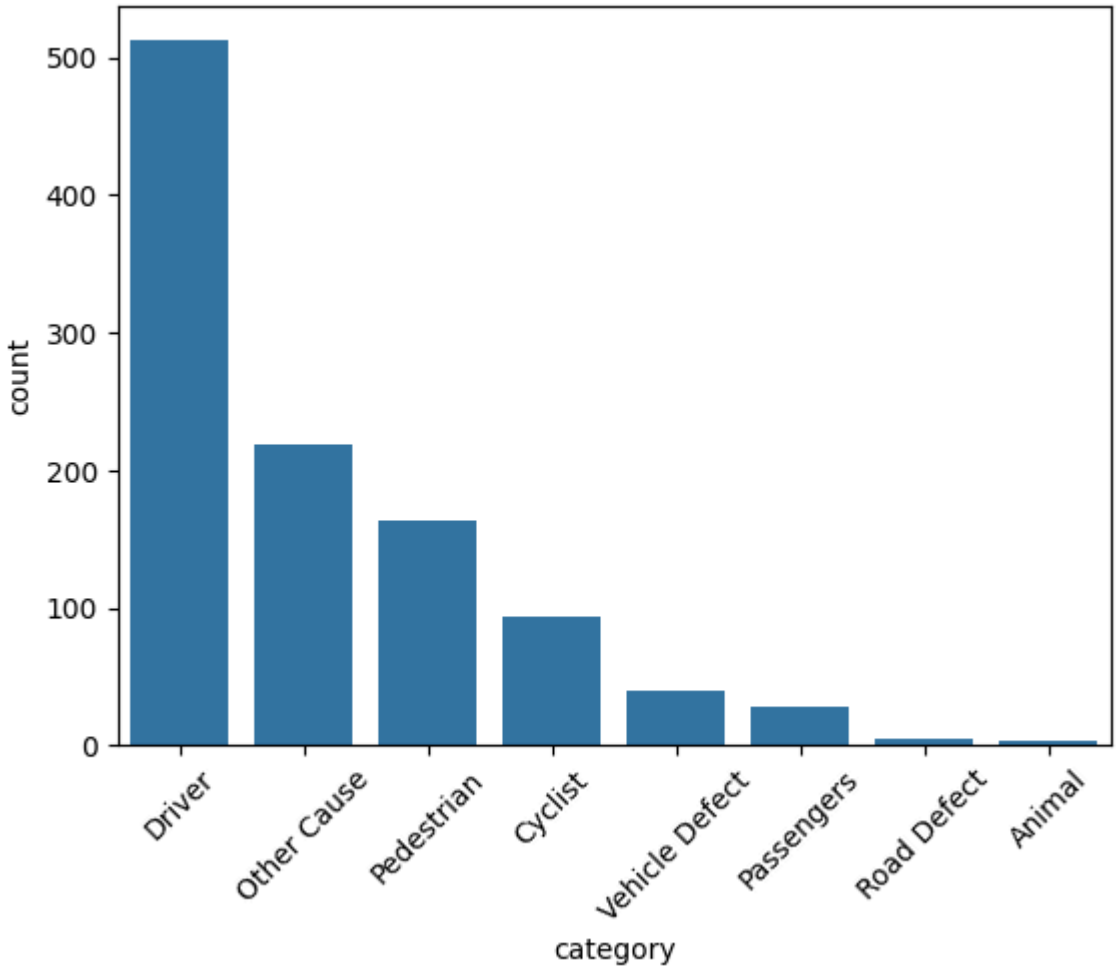
	age	cause_code	number_of_victims
age	1.000000	0.006154	0.002499
cause_code	0.006154	1.000000	-0.043454
number_of_victims	0.002499	-0.043454	1.000000

Visualizations

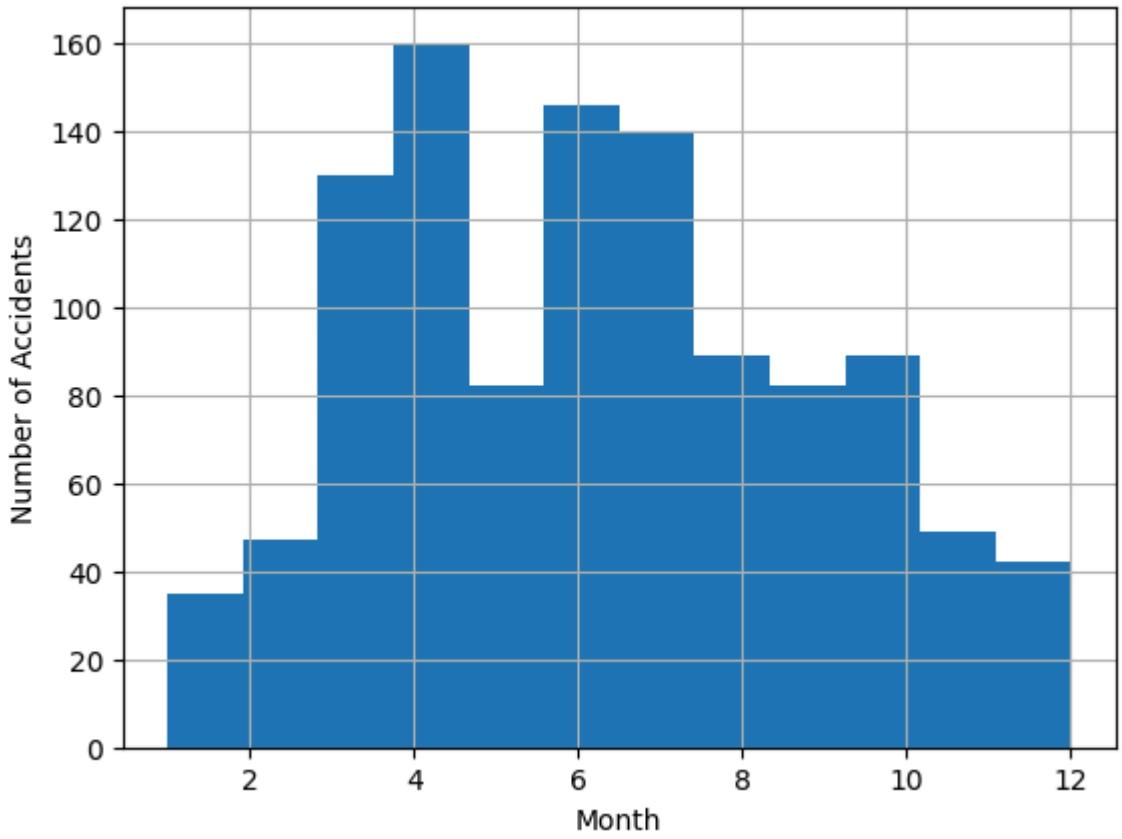
In [85]:
```
data.hist(figsize=(14,10), bins=30)
plt.tight_layout()
plt.show()
```



In [86]:
```
# plt.figure(figsize=(8,10))
sns.countplot(data=data, x='category', order=data['category'].value_counts().ind
plt.xticks(rotation=45)
plt.show()
```

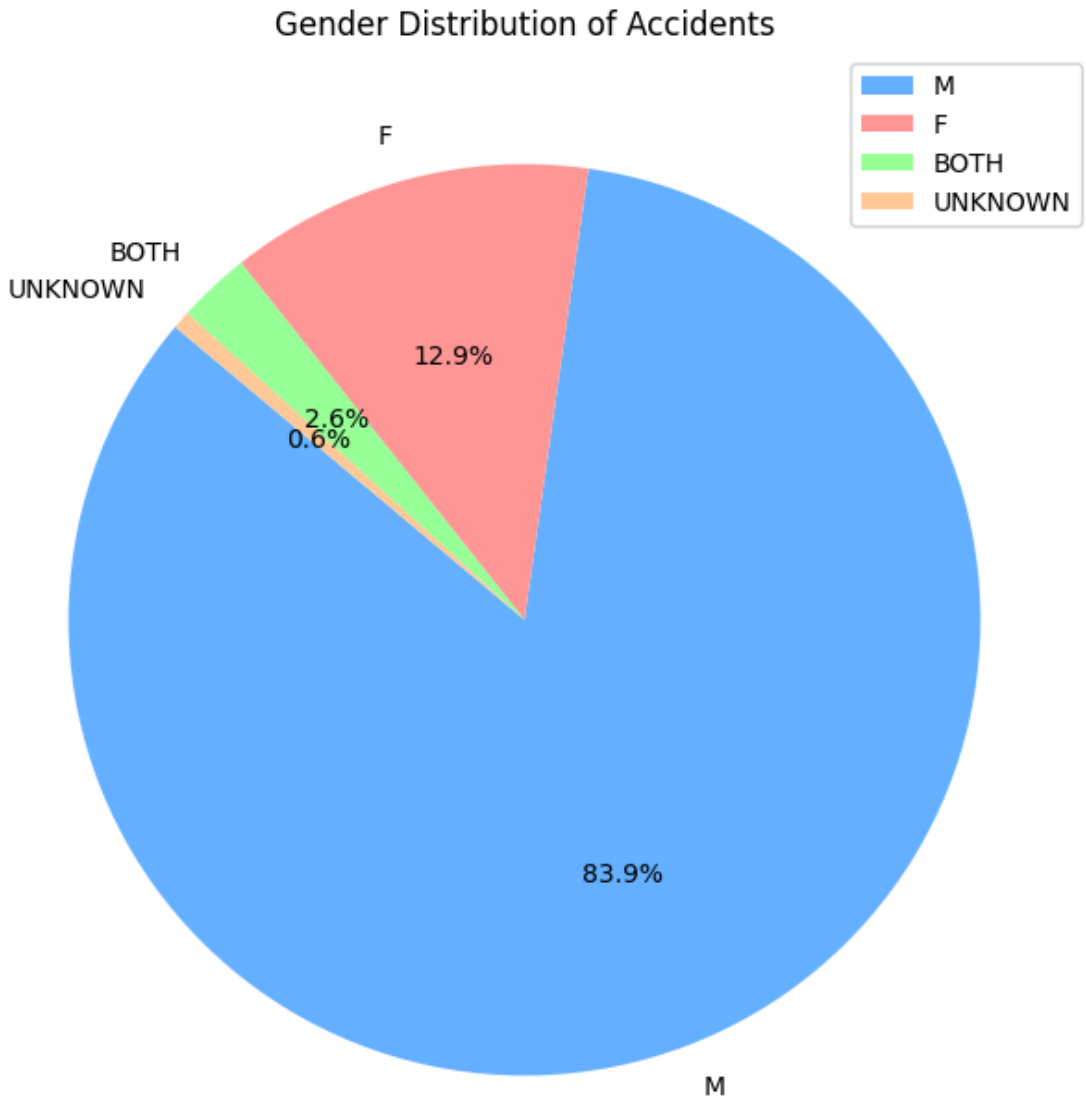


In [87]:
```
data['date'].dt.month.hist(bins=12)
plt.xlabel('Month')
plt.ylabel('Number of Accidents')
plt.show()
```



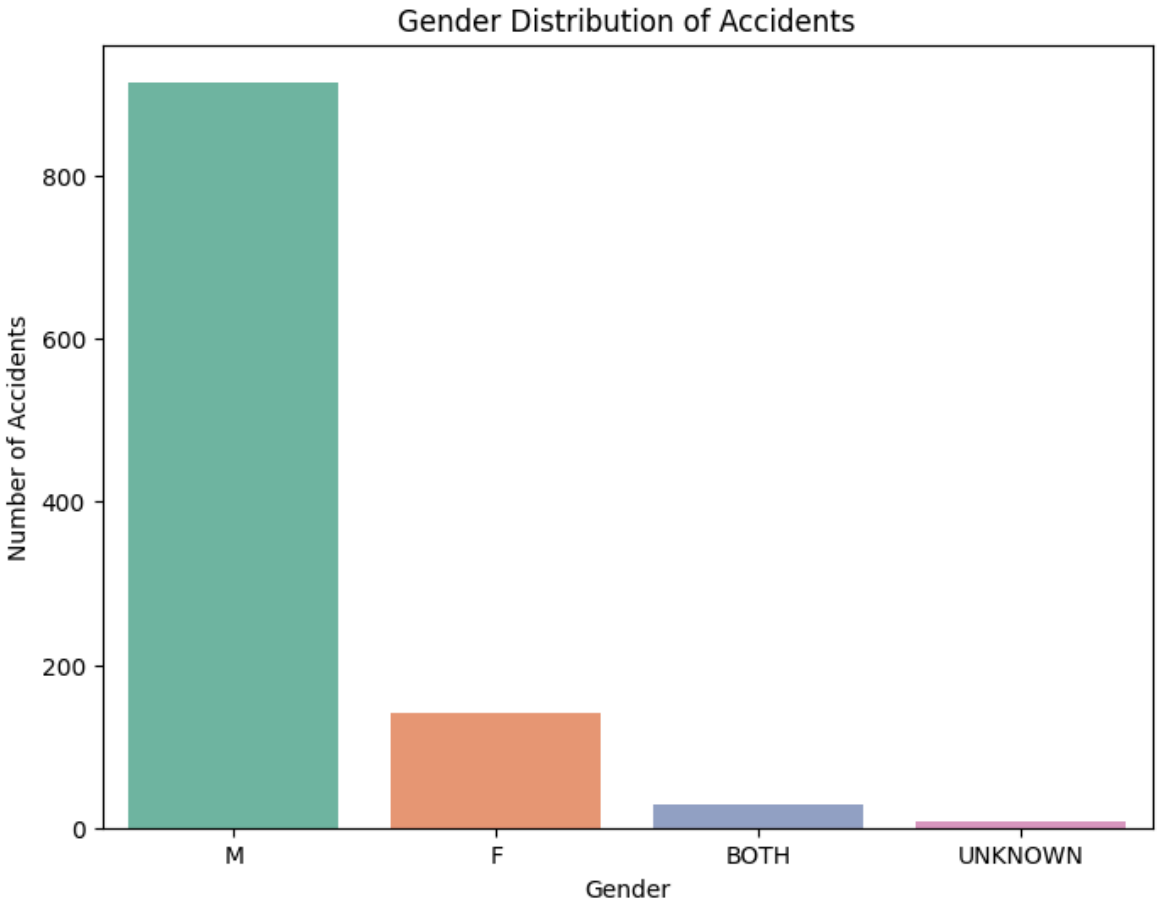
In [88]:
```
gender_counts = data['gender'].value_counts()
```

In [89]:
```
plt.figure(figsize=(8,8))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle
plt.title("Gender Distribution of Accidents")
plt.legend()
plt.show()
```



In [90]:
```
plt.figure(figsize=(8,6))
sns.barplot(x=gender_counts.index, y=gender_counts.values, palette='Set2')
plt.title("Gender Distribution of Accidents")
plt.xlabel("Gender")
plt.ylabel("Number of Accidents")
plt.show()
```

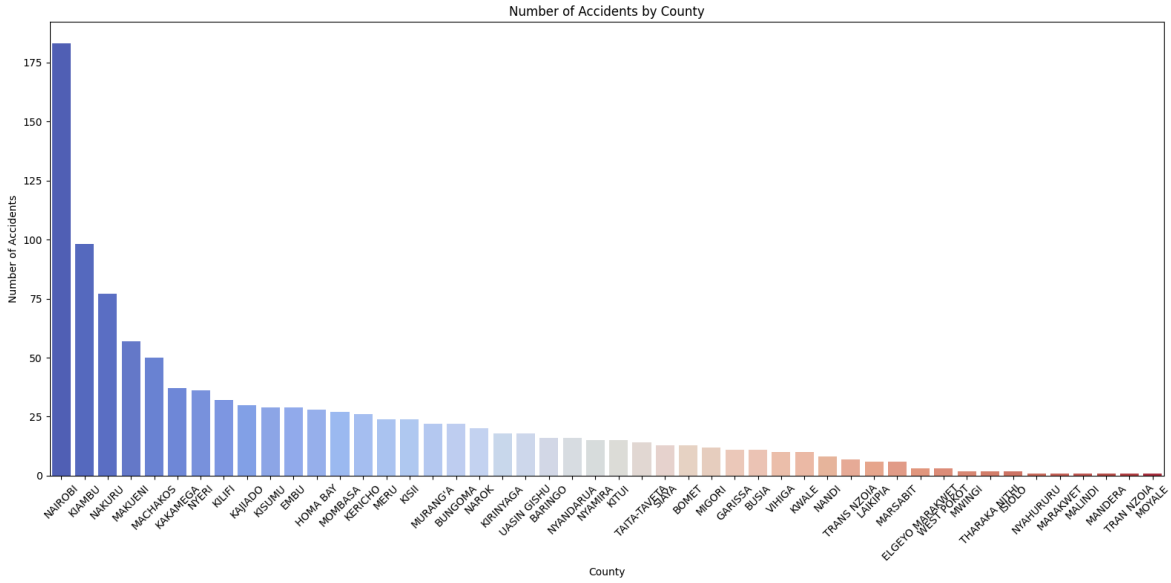
```
C:\Users\Alvo\AppData\Local\Temp\ipykernel_13504\3977890749.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x=gender_counts.index, y=gender_counts.values, palette='Set2')
```



In [91]:
```
county_counts = data['county'].value_counts()
```

In [92]:
```
plt.figure(figsize=(16,8))
sns.barplot(x=county_counts.index, y=county_counts.values, palette='coolwarm')
plt.xticks(rotation=45)
plt.title("Number of Accidents by County")
plt.xlabel("County")
plt.ylabel("Number of Accidents")
plt.tight_layout()
plt.show()
```

```
C:\Users\Alvo\AppData\Local\Temp\ipykernel_13504\3221071651.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x=county_counts.index, y=county_counts.values, palette='coolwarm')
```



In [93]:
```
cause_desc_counts = data.groupby(['cause_code', 'description']).size().reset_ind
```

In [94]:
```
# Get total accidents per cause code
total_per_cause = cause_desc_counts.groupby('cause_code')['Count'].sum().reset_i
top_cause_codes = total_per_cause.sort_values('Count', ascending=False).head(20)
```

In [95]:
```
# Filter only top 10 cause codes
top_cause_desc = cause_desc_counts[cause_desc_counts['cause_code'].isin(top_caus
```

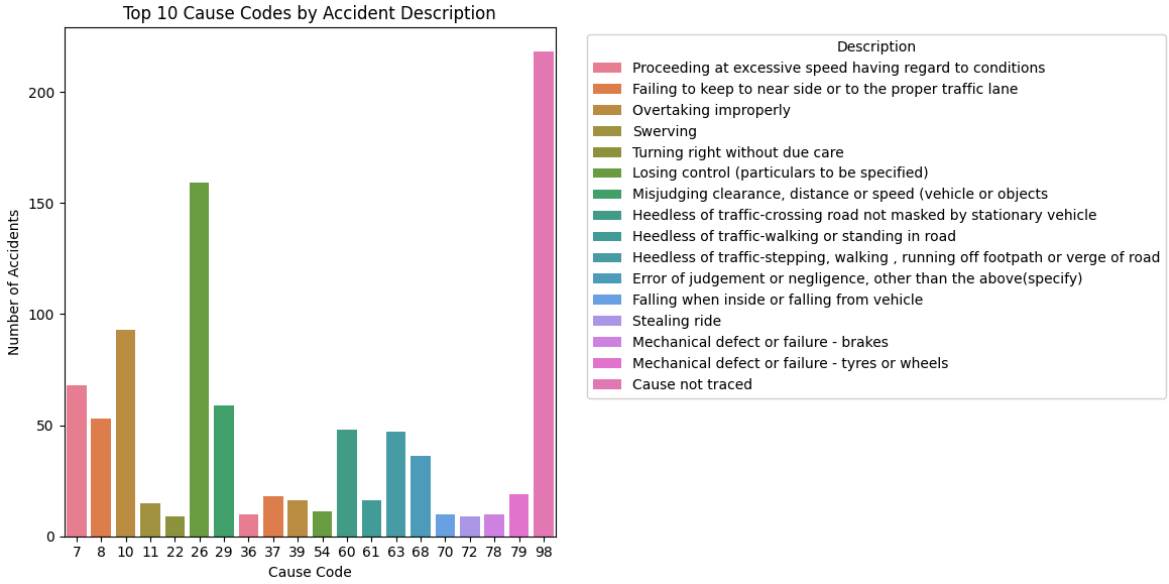
In [96]:
```
top_cause_desc
```

Out[96]:

	cause_code	description	Count
3	7	Proceeding at excessive speed having regard to...	68
4	8	Failing to keep to near side or to the proper ...	53
6	10	Overtaking improperly	93
7	11	Swerving	15
18	22	Turning right without due care	9
21	26	Losing control (particulars to be specified)	159
23	29	Misjudging clearance, distance or speed (vehic...	59
26	36	Proceeding at excessive speed having regard to...	10
27	37	Failing to keep to near side or to the proper ...	18
29	39	Overtaking improperly	16
39	54	Losing control (particulars to be specified)	11
43	60	Heedless of traffic-crossing road not masked b...	48
44	61	Heedless of traffic-walking or standing in road	16
46	63	Heedless of traffic-stepping, walking , runnin...	47
50	68	Error of judgement or negligence, other than t...	36
52	70	Falling when inside or falling from vehicle	10
54	72	Stealing ride	9
58	78	Mechanical defect or failure - brakes	10
59	79	Mechanical defect or failure - tyres or wheels	19
70	98	Cause not traced	218

In [97]:

```
# Plot
plt.figure(figsize=(12,6))
sns.barplot(
    data=top_cause_desc.sort_values('Count', ascending=False),
    x='cause_code',
    y='Count',
    hue='description'
)
plt.title("Top 10 Cause Codes by Accident Description")
plt.xlabel("Cause Code")
plt.ylabel("Number of Accidents")
plt.legend(title='Description', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



Time Series Analysis

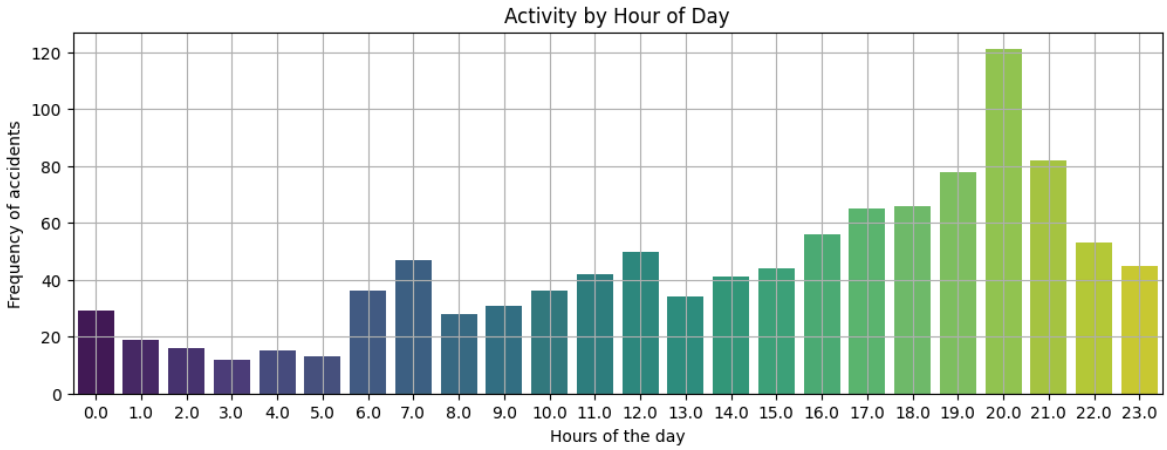
In [98]:
```
df = data.copy()
df.rename(columns={'time':'timestamp'}, inplace=True)
```

In [99]:
```
df['timestamp'] = pd.to_timedelta(df['timestamp'].astype(str))
```

In [100…
```
df['hour'] = df['timestamp'].dt.components.hours
hourly_series = df.groupby('hour').size()
```

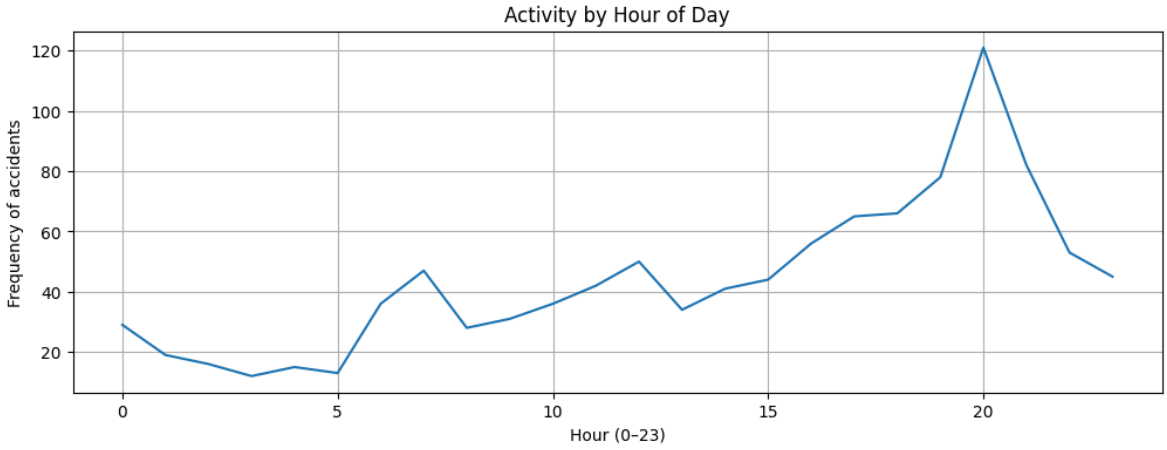
In [101…
```
hourly_df = hourly_series.reset_index()
hourly_df.columns = ['hour', 'count']  # rename for seaborn

plt.figure(figsize=(12,4))
sns.barplot(x='hour', y='count', data=hourly_df, palette='viridis')
plt.title("Activity by Hour of Day")
plt.xlabel("Hours of the day")
plt.ylabel("Frequency of accidents")
plt.grid(True)
plt.show()
```

```
C:\Users\Alvo\AppData\Local\Temp\ipykernel_13504\4100117960.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x='hour', y='count', data=hourly_df, palette='viridis')
```



In [102…
```
plt.figure(figsize=(12,4))
sns.lineplot(x='hour', y='count', data=hourly_df, palette='viridis')
plt.title("Activity by Hour of Day")
plt.xlabel("Hour (0-23)")
plt.ylabel("Frequency of accidents")
plt.grid(True)
plt.show()
```

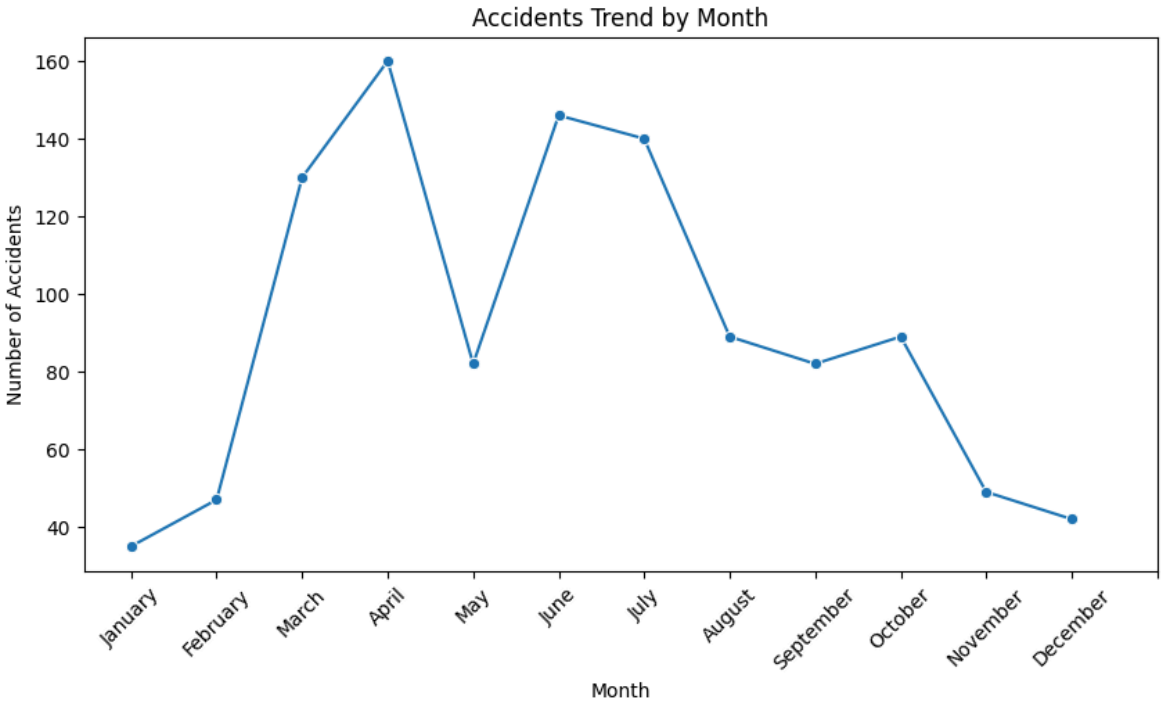
```
C:\Users\Alvo\AppData\Local\Temp\ipykernel_13504\3206648552.py:2: UserWarning: Ig
noring `palette` because no `hue` variable has been assigned.
  sns.lineplot(x='hour', y='count', data=hourly_df, palette='viridis')
```



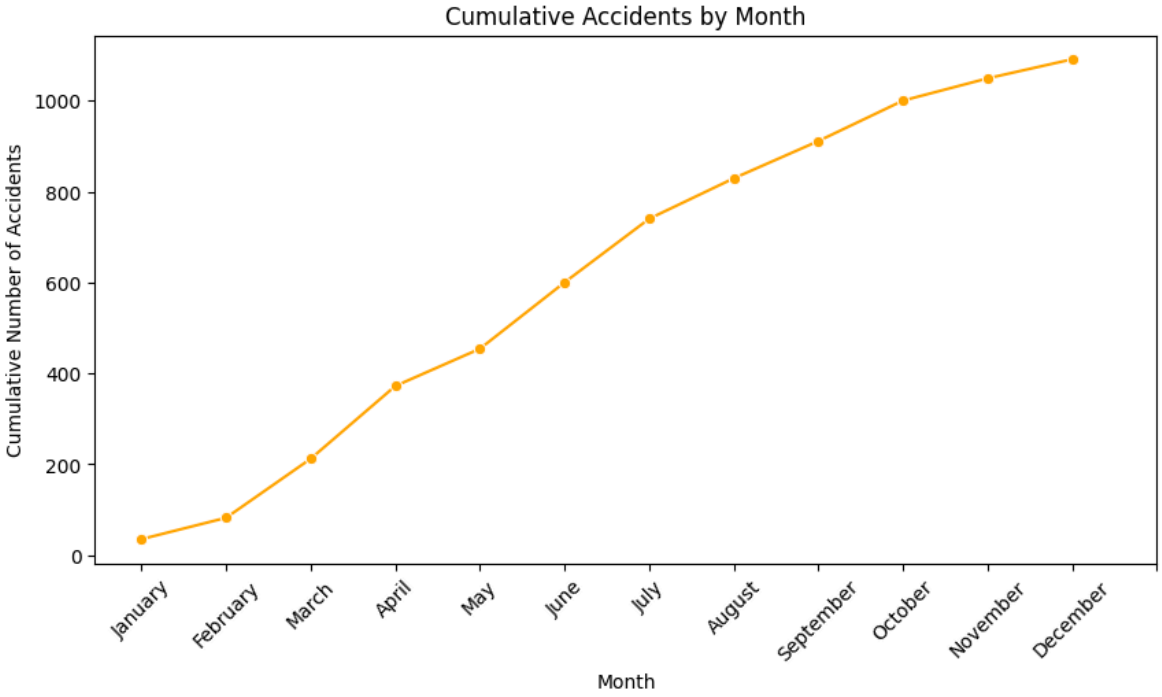
In [103…
```
month_counts = data['date'].dt.month.value_counts().sort_index()
```

In [ ]:
```
month_names = [calendar.month_name[m] for m in month_counts.index]
```

In [133…
```
# Plot
plt.figure(figsize=(10,5))
sns.lineplot(x=month_names, y=month_counts.values, marker='o')
plt.title("Accidents Trend by Month")
plt.xlabel("Month")
plt.ylabel("Number of Accidents")
plt.xticks(range(0,13))
plt.xticks(rotation=45)
plt.show()
```



In [139…
```
# Cumulative accidents over months
plt.figure(figsize=(10,5))
sns.lineplot(x=month_names, y=month_counts.cumsum(), marker='o', color='orange')
plt.title("Cumulative Accidents by Month")
plt.xlabel("Month")
plt.ylabel("Cumulative Number of Accidents")
plt.xticks(range(0,13))
plt.xticks(rotation=45)
plt.show()
```



In [106…
```
# Aggregate accidents by month
monthly_data = data.groupby(data['date'].dt.to_period('M')).size()
```

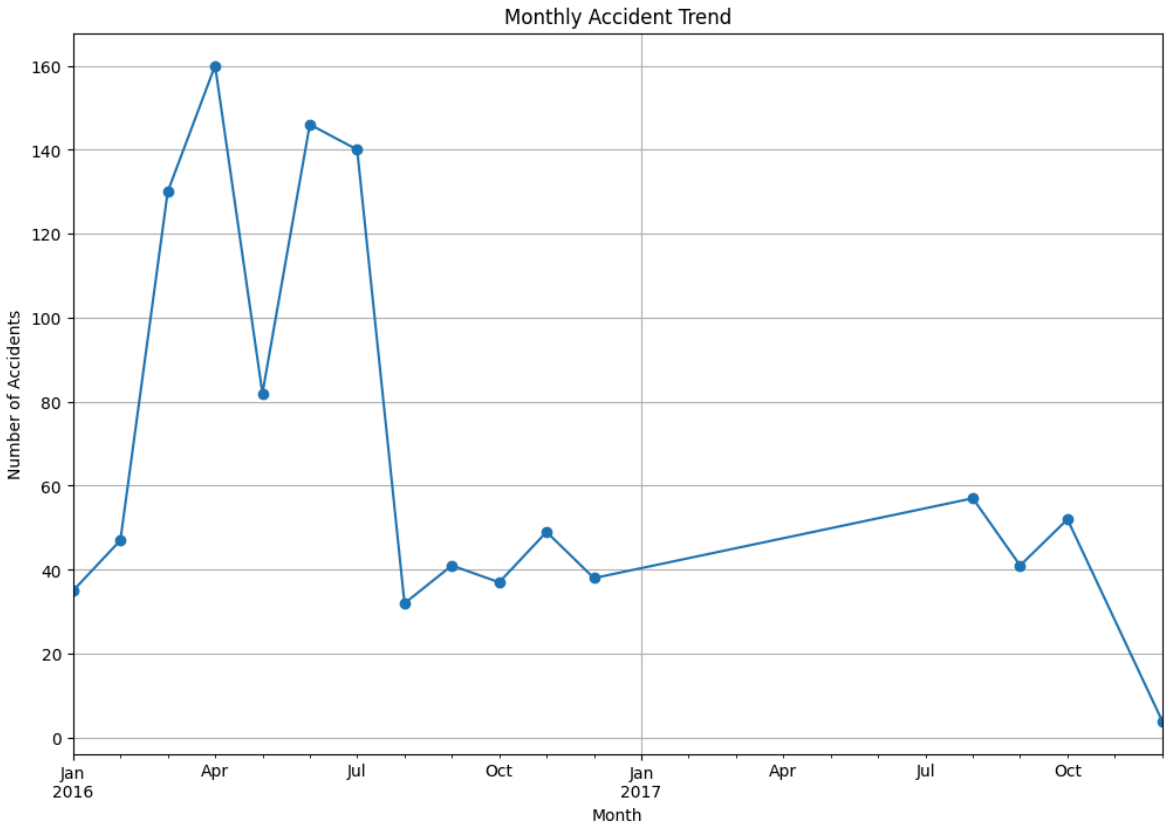
In [107…
```
monthly_data
```

Out[107…

```
date
2016-01     35
2016-02     47
2016-03    130
2016-04    160
2016-05     82
2016-06    146
2016-07    140
2016-08     32
2016-09     41
2016-10     37
2016-11     49
2016-12     38
2017-08     57
2017-09     41
2017-10     52
2017-12      4
Freq: M, dtype: int64
```

In [136…

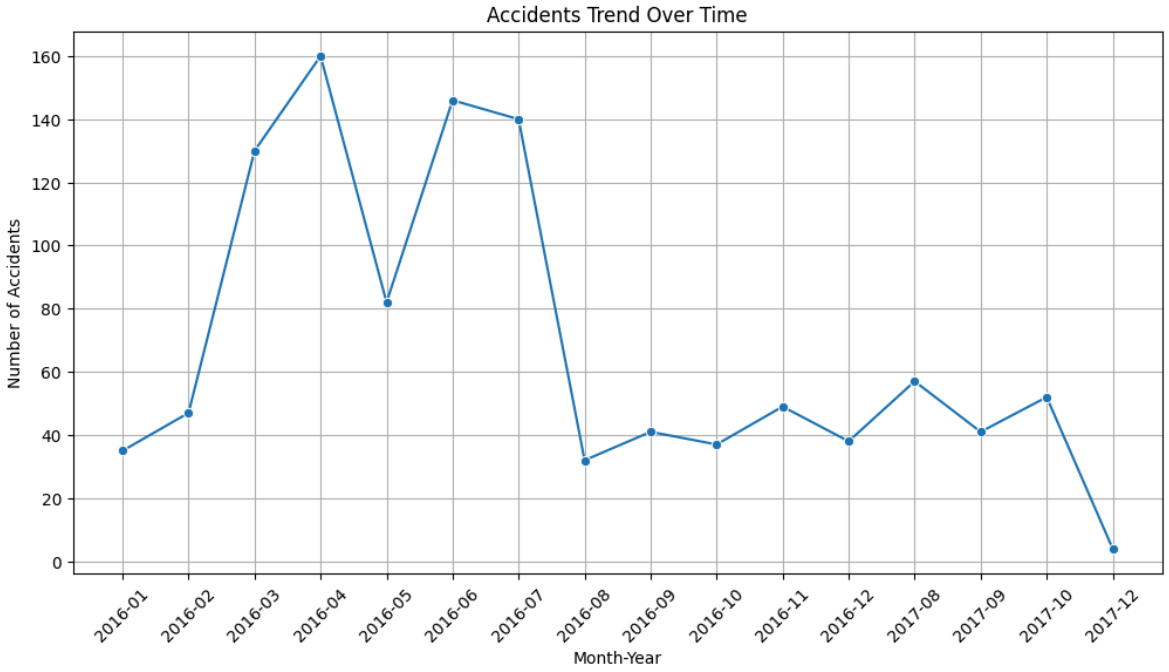
```
data['month'] = data['date'].dt.to_period('M')

monthly = data.groupby('month').size()

plt.figure(figsize=(12,8))
monthly.plot(kind='line', marker='o')
plt.title("Monthly Accident Trend")
plt.xlabel("Month")
plt.ylabel("Number of Accidents")
plt.grid()
plt.show()
```



In [109…

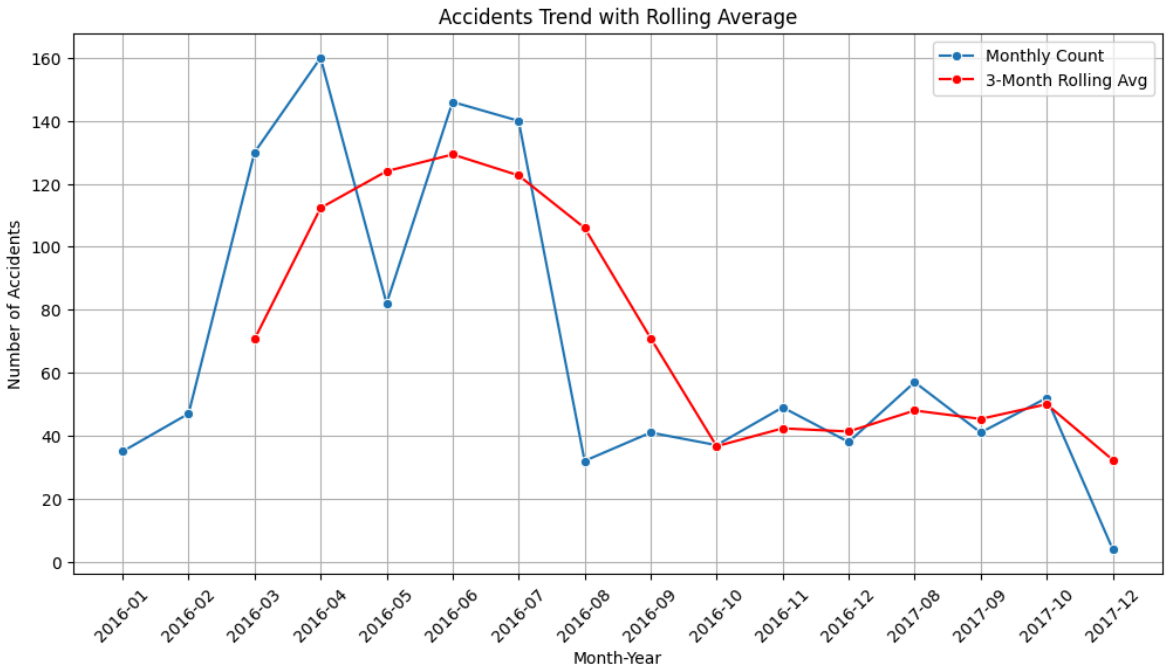
```
plt.figure(figsize=(12,6))
sns.lineplot(x=monthly_data.index.astype(str), y=monthly_data.values, marker='o'
plt.title("Accidents Trend Over Time")
```

```
plt.xlabel("Month-Year")
plt.ylabel("Number of Accidents")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```



In [110…

```
rolling_avg = monthly_data.rolling(window=3).mean()
plt.figure(figsize=(12,6))
sns.lineplot(x=monthly_data.index.astype(str), y=monthly_data.values, marker='o'
sns.lineplot(x=rolling_avg.index.astype(str), y=rolling_avg.values, marker='o',
plt.title("Accidents Trend with Rolling Average")
plt.xlabel("Month-Year")
plt.ylabel("Number of Accidents")
plt.xticks(rotation=45)
plt.grid()
plt.legend()
plt.show()
```

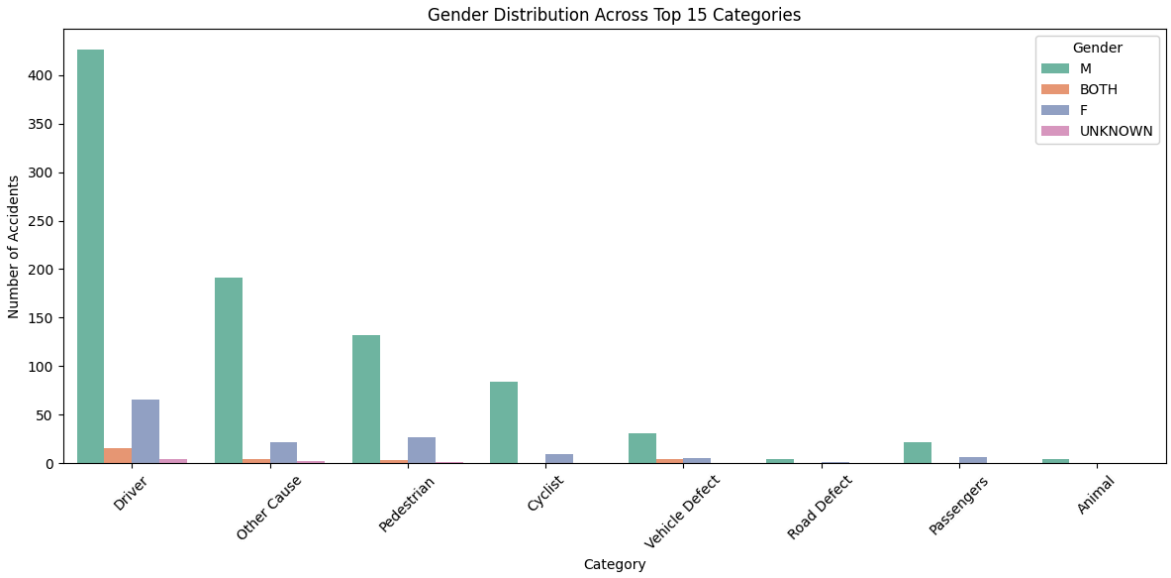


Bivariate Anlaysis

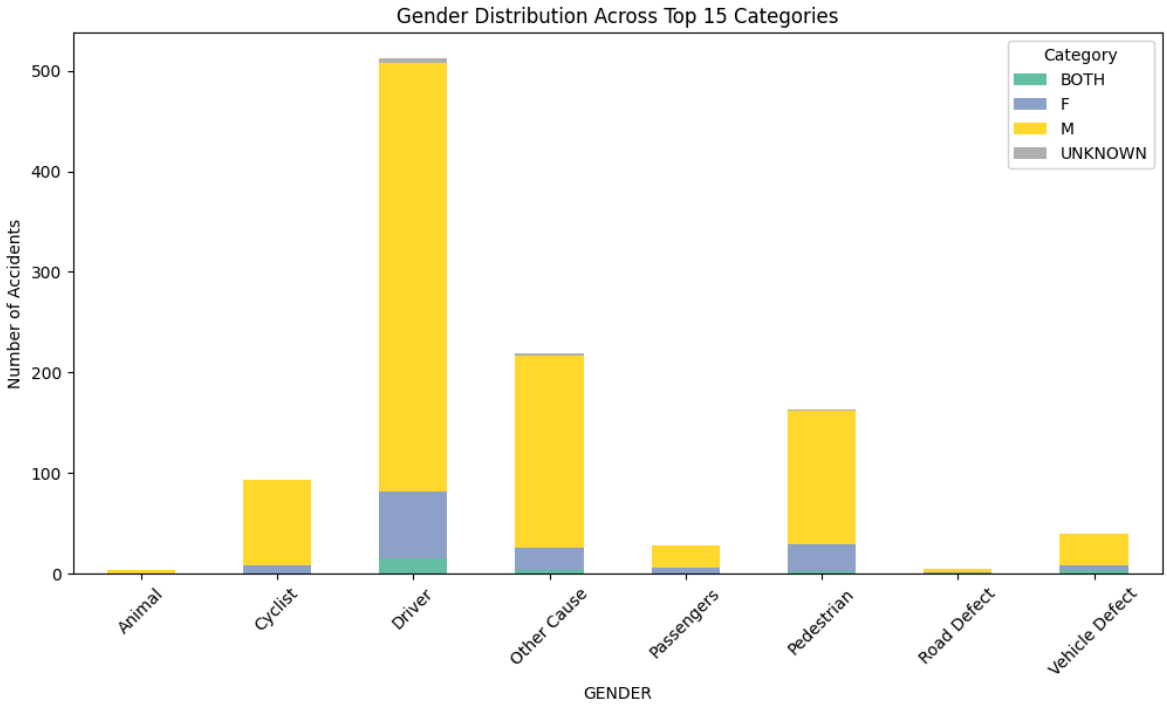
1. Gender vs Category/Accident Description

- Are certain types of incidents more common in males vs females?

In [111…
```
top_categories = data['category'].value_counts().head(15).index  # top 15 catego
subset_cat_gender = data[data['category'].isin(top_categories)]

plt.figure(figsize=(12,6))
sns.countplot(x='category', hue='gender', data=subset_cat_gender, palette='Set2'
plt.title("Gender Distribution Across Top 15 Categories")
plt.xticks(rotation=45)
plt.xlabel("Category")
plt.ylabel("Number of Accidents")
plt.legend(title='Gender')
plt.tight_layout()
plt.show()
```



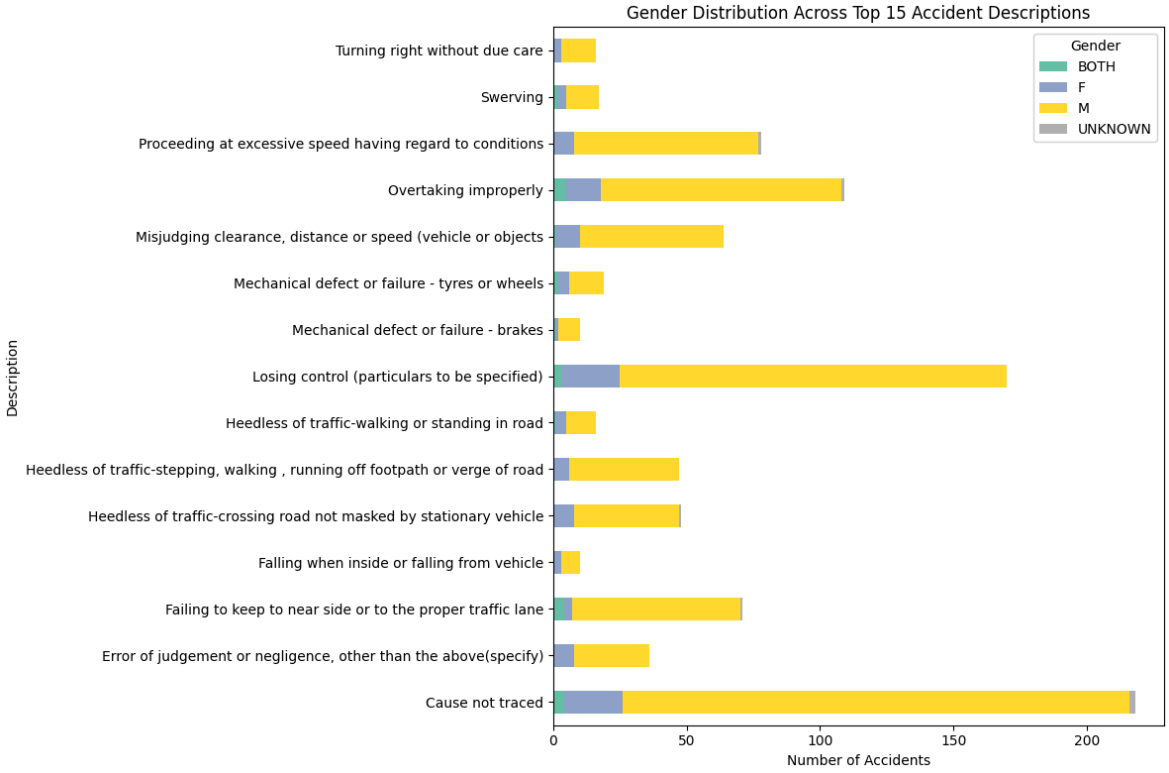
In [112…
```
top_categories = data['category'].value_counts().head(15).index
subset_cat_gender = data[data['category'].isin(top_categories)]

# Create a pivot table for stacked bar
pivot_gender_cat = pd.crosstab(subset_cat_gender['category'], subset_cat_gender[

# Plot stacked bar
pivot_gender_cat.plot(kind='bar', stacked=True, figsize=(12,6), colormap='Set2')
plt.title("Gender Distribution Across Top 15 Categories")
plt.xlabel("GENDER")
plt.ylabel("Number of Accidents")
plt.xticks(rotation=45)
plt.legend(title='Category')
plt.show()
```



- Which accident type is most prevalent for each gender?

In [113…

```
# Optionally, focus on top 15 descriptions
top_desc = data['description'].value_counts().head(15).index
subset = data[data['description'].isin(top_desc)]

# Pivot for stacked bar
pivot = pd.crosstab(subset['description'], subset['gender'])

# Plot horizontal stacked bar
pivot.plot(kind='barh', stacked=True, figsize=(12,8), colormap='Set2')
plt.title("Gender Distribution Across Top 15 Accident Descriptions")
plt.xlabel("Number of Accidents")
plt.ylabel("Description")
plt.legend(title='Gender')
plt.tight_layout()
plt.show()
```

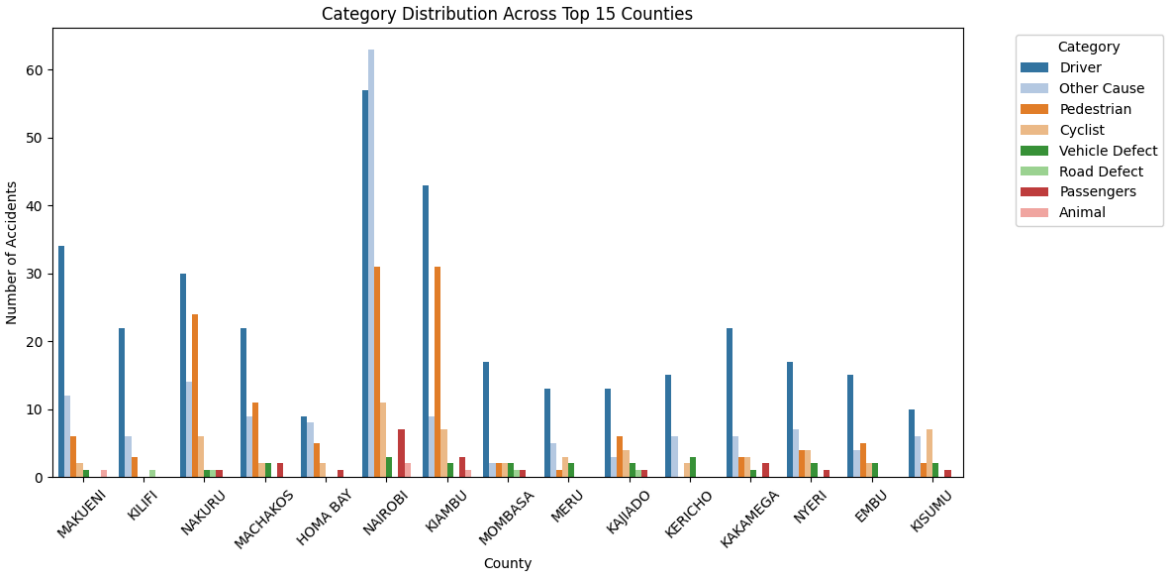


2. County vs Category / Cause Code

- Which counties have more driver-related incidents?

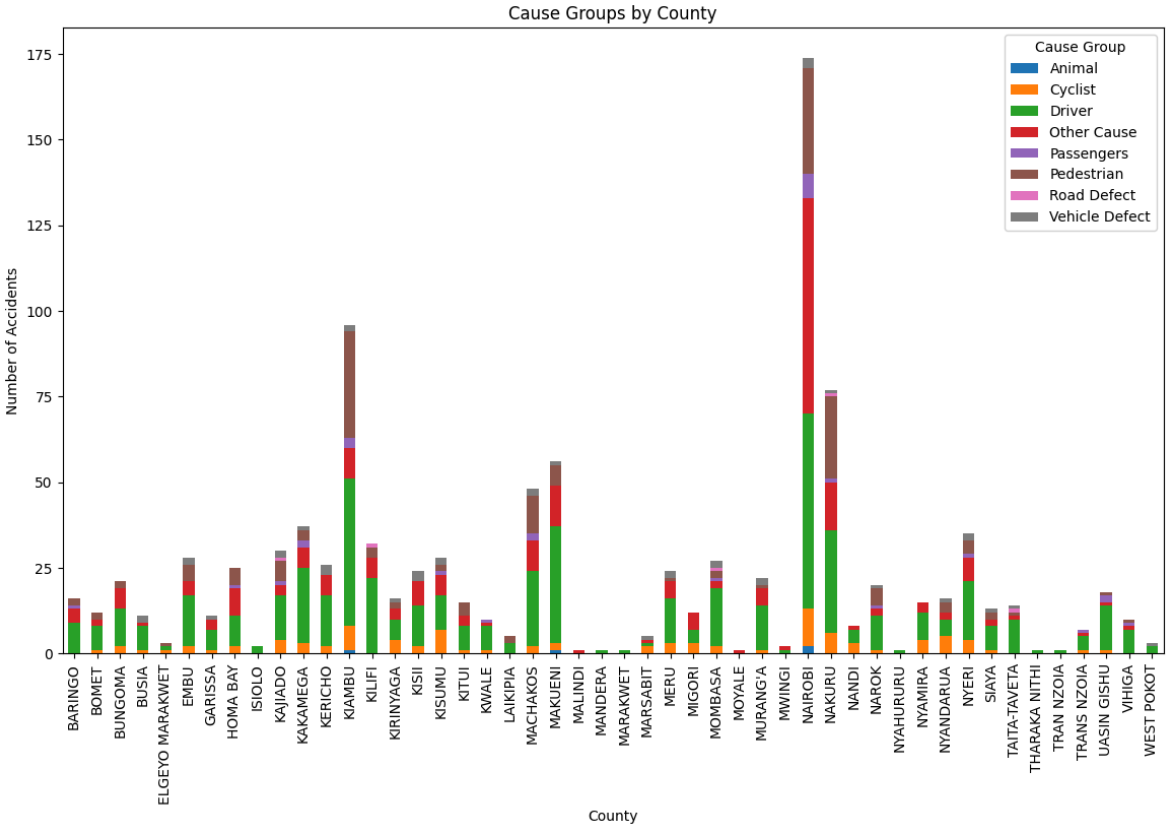
In [114...

```
top_counties = data['county'].value_counts().head(15).index  # top 15 counties
subset_county_cat = data[data['county'].isin(top_counties)]

plt.figure(figsize=(12,6))
sns.countplot(x='county', hue='category', data=subset_county_cat, palette='tab20
plt.title("Category Distribution Across Top 15 Counties")
plt.xticks(rotation=45)
plt.xlabel("County")
plt.ylabel("Number of Accidents")
plt.legend(title='Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



- Which categorical accident takes place the most in any given county?

In [115…
```
stacked = pd.crosstab(data['county'], data['category'])

stacked.plot(kind='bar', stacked=True, figsize=(14,8))
plt.title("Cause Groups by County")
plt.xlabel("County")
plt.ylabel("Number of Accidents")
plt.legend(title='Cause Group')
plt.show()
```



Folium Choropleth

In [116…
```
data['county'] = data['county'].str.title()
```

In [117…
```
data['county'].values
```

Out[117…
```
array(['Makueni', 'Taita-Taveta', 'Kilifi', ..., 'Nairobi', 'Makueni',
       'Nairobi'], dtype=object)
```

In [118…
```
import json

geojson_path = 'kenyan_counties.geojson'
with open(geojson_path) as f:
    geojson_data = json.load(f)

# Print the keys of the first feature to see the property names
print(geojson_data['features'][0]['properties'])
```

```
{'OBJECTID': 1, 'AREA': 5.677, 'PERIMETER': 15.047, 'COUNTY3_': 2, 'COUNTY3_ID':
1, 'COUNTY': 'Turkana', 'Shape_Leng': 15.0468376942, 'Shape_Area': 5.67698507035}
```

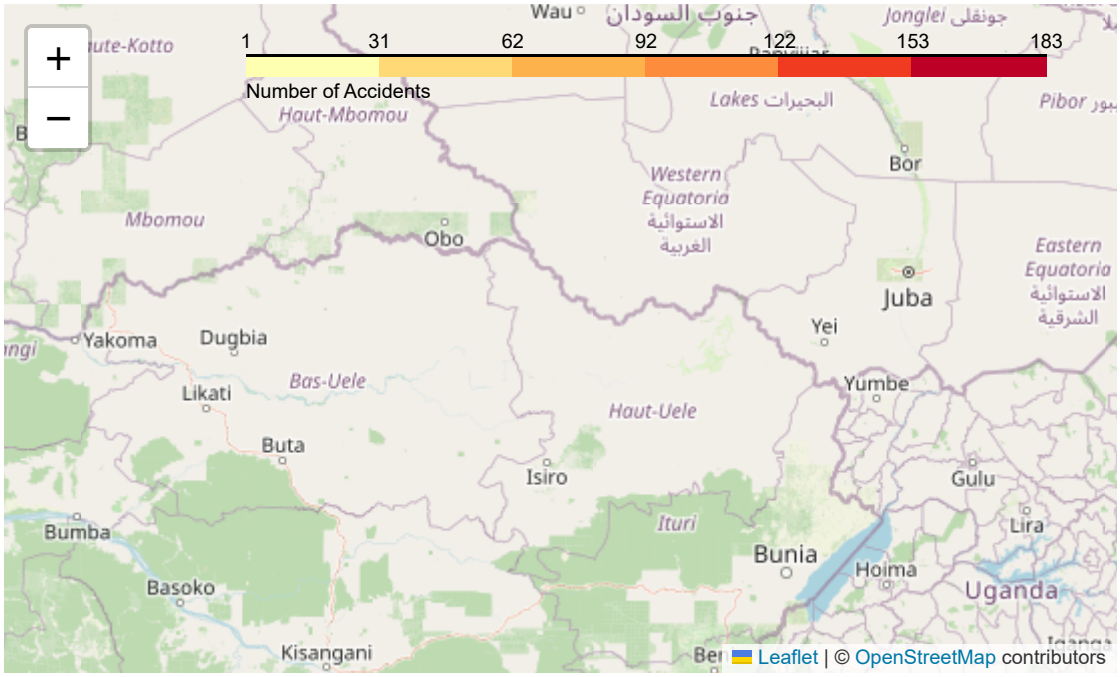
In [119…

```
import pandas as pd
import folium
import json

# --- Load the GeoJSON file ---
geojson_path = "kenyan_counties.geojson"
with open(geojson_path) as f:
    counties_geo = json.load(f)

# --- Prepare accident data ---
accidents_per_county = data['county'].value_counts().reset_index()
accidents_per_county.columns = ['county', 'Accidents']

# Create a dictionary of accident counts per county
acc_dict = accidents_per_county.set_index("county")["Accidents"].to_dict()

# Add accident count into each feature's properties
for feature in counties_geo["features"]:
    county_name = feature["properties"]["COUNTY"]
    feature["properties"]["ACCIDENTS"] = acc_dict.get(county_name, 0)

# --- Create map centered on Kenya ---
m = folium.Map(location=[-1.286389, 36.817223], zoom_start=6)

# --- Create the choropleth ---
folium.Choropleth(
    geo_data=counties_geo,
    data=accidents_per_county,
    columns=['county', 'Accidents'],
    key_on='feature.properties.COUNTY',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Number of Accidents'
).add_to(m)

# --- Add tooltips for better interactivity ---
folium.GeoJson(
    counties_geo,
    name="County Boundaries",
    tooltip=folium.GeoJsonTooltip(
        fields=['COUNTY', 'ACCIDENTS'],
        aliases=['County:', 'Accidents:'],
        localize=True
    )
).add_to(m)

# --- Save output ---
m.save("accidents_choropleth.html")
```

In [120…

```
m
```

Out[120…



Summary

In [121…

```
data.isna().sum().sort_values(ascending=False)
```

Out[121…

```
age                    582
time                    32
category                27
description             27
cause_code              26
place                    5
number_of_victims        4
county                   3
road                     3
base/sub_base            2
brief_accident_details   1
name_of_victim           1
date                     0
gender                   0
victim                   0
mv_involved              0
month                    0
dtype: int64
```

In [122…

```
numeric_cols = data.select_dtypes(include='number').columns

for col in numeric_cols:
    q1 = data[col].quantile(0.25)
    q3 = data[col].quantile(0.75)
    iqr = q3 - q1
    lower = q1 - 1.5 * iqr
    upper = q3 + 1.5 * iqr
    outliers = data[(data[col] < lower) | (data[col] > upper)]
    print(f"{col}: {len(outliers)} outliers")
```

```
age: 21 outliers
cause_code: 0 outliers
number_of_victims: 93 outliers
```

In [144…

```
cat_cols = data.select_dtypes(include='object').columns

for col in cat_cols:
    print(f"\n{col.upper()}:\n", data[col].value_counts().head(10))
```

```
TIME:
 time
20:30:00    43
21:00:00    38
20:00:00    35
19:00:00    28
19:30:00    27
18:00:00    25
17:30:00    24
18:30:00    24
22:30:00    22
21:30:00    20
Name: count, dtype: int64

BASE/SUB_BASE:
 base/sub_base
NAKURU        31
EMBAKASI      20
ATHI RIVER    20
STAREHE       19
KIKUYU        18
NYERI         17
MAKINDU       17
RUIRU         16
KERICHO       15
KITUI         15
Name: count, dtype: int64

COUNTY:
 county
Nairobi     183
Kiambu       98
Nakuru       77
Makueni      57
Machakos     50
Kakamega     37
Nyeri        36
Kilifi       32
Kajiado      30
Kisumu       29
Name: count, dtype: int64

ROAD:
 road
NAIROBI MOMBASA       21
MOMBASA NAIROBI       20
WAIYAKI WAY           17
THIKA SUPERHIGHWAY    16
MOMBASA ROAD          15
NAIROBI-MOMBASA       15
MOMBASA-NAIROBI       12
ELDORET NAKURU         9
KISUMU BUSIA           8
NAKURU ELDORET         8
Name: count, dtype: int64

PLACE:
 place
NEAR KANGEMI STAGE          3
STATE HOUSE                 3
```

```
NEAR SHELL PETROL STATION    3
GICHIEGO                     2
KINGEERO AREA                2
CHULAIMBO AREA               2
NYAKOE AREA                  2
N MARKET                     2
CHEMOSIT                     2
ICHUNI AREA                  2
Name: count, dtype: int64

MV_INVOLVED:
 mv_involved
UNKNOWN                                                          53
UNKNOWN M/V                                                      45
UNKNOWN M/V (HIT & RUN)                                           7
UNKNOWN M/V AND M/A/PED                                           3
KCL 879M & KBR 759P MITSUBISHI FH LORRIES                         2
KBX 432Z TOYOTA MATATU                                            2
KWN 383 ISUZU LORRY,KCJ 424L ISUZU BUS & KCJ 431M TOYOTA MATATU   2
KBU 743D/ZE 1567 M/BENZ TRAILER & KCC 683W ISUZU LORRY            2
KCK 407Z T/MATATU & KMEA 878P BOXER                               2
UNKNOWN M/V PEDESTRIAN                                            2
Name: count, dtype: int64

BRIEF_ACCIDENT_DETAILS:
 brief_accident_details
THE VEHICLE KNOCKED DOWN THE VICTIM                          216
HEAD ON COLLISION                                            117
HIT AND RUN                                                   86
THE VEHICLE KNOCKED DOWN THE VICTIM WHILE CROSSING THE ROAD   23
HIT & RUN                                                     16
THE VEHICLE KNOCKED DOWN A VICTIM                             15
THE VEHICLE HIT THE M/CYCLE                                   11
THE VEHICLE KNOCKED DOWN A PEDESTRIAN                         11
THE VEHICLE LOST CONTROL AND ROLLED SEVERAL TIMES              8
THE CYCLE KNOCKED DOWN THE VICTIM                              8
Name: count, dtype: int64

NAME_OF_VICTIM:
 name_of_victim
UNKNOWN              718
MUTINDA KOMU           3
JOHN MWANGI            2
MARY WANJIKU           2
MUTAURA                2
DENNIS OTIENO          1
WYCLIFF SAIYA          1
ANTHONY MWANGI         1
JULIUS NDIWA           1
MARITINA NABALAYO      1
Name: count, dtype: int64

GENDER:
 gender
M          915
F          141
BOTH        28
UNKNOWN      7
Name: count, dtype: int64
```

```
VICTIM:
 victim
PEDESTRIAN             446
M/CYCLIST              198
PASSENGER              133
DRIVER                 115
P/PASSENGER             64
P/CYCLIST               22
RIDER                   21
PASSENGERS              20
RIDER & P/PASSENGER      6
DRIVER & PASSENGER       6
Name: count, dtype: int64

DESCRIPTION:
 description
Cause not traced
218
Losing control (particulars to be specified)
170
Overtaking improperly
109
Proceeding at excessive speed having regard to conditions
78
Failing to keep to near side or to the proper traffic lane
71
Misjudging clearance, distance or speed (vehicle or objects
64
Heedless of traffic-crossing road not masked by stationary vehicle
48
Heedless of traffic-stepping, walking , running off footpath or verge of road
47
Error of judgement or negligence, other than the above(specify)
36
Mechanical defect or failure - tyres or wheels
19
Name: count, dtype: int64

CATEGORY:
 category
Driver            512
Other Cause       219
Pedestrian        163
Cyclist            93
Vehicle Defect     40
Passengers         28
Road Defect         5
Animal              4
Name: count, dtype: int64
```

In [124…

```
from IPython.display import Markdown

Markdown("""
# **Summary of Key Findings**
- Accident rates are highest in **Nairobi, Kiambu and Nakuru counties.**.
- These counties are all designated urban areas.
- The top causes of accidents include **excessive speeding, losing control of th
- The highest-risk hours are between **17:00 - 20:00 hrs**.
- Severe accidents cluster around **Kangemi stage, in Nairobi**.
- Trends show an increase in accidents gradually peaking in **July** then a drop
```

```
# **Recommendations**
- Improve enforcement in high-risk areas.
- Target campaigns at most common causes.
- Upgrade signage and lighting in hotspot zones.
""")
```

Out[124...

Summary of Key Findings

- Accident rates are highest in **Nairobi, Kiambu and Nakuru counties.**.
- These counties are all designated urban areas.
- The top causes of accidents include **excessive speeding, losing control of the car and unknown circumstances**.
- The highest-risk hours are between **17:00 - 20:00 hrs**.
- Severe accidents cluster around **Kangemi stage, in Nairobi**.
- Trends show an increase in accidents gradually peaking in **July** then a drop off towards the end of year.

Recommendations

- Improve enforcement in high-risk areas.
- Target campaigns at most common causes.
- Upgrade signage and lighting in hotspot zones.