Understanding different Datasets for the Analysis of Traffic-Accidents-in-Kenya

In [20]:
```
# importing necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Exploring the road death 2019 dataset from kaggle by Kamau Munyori

In [21]:
```
# Load the road death data by Kamau Munyori in kaggle
road_death = pd.read_csv(r"Data\road_death_2019.csv")
road_death.head(5)
```

Out[21]:

	continent	code	country	year	road traffic death rate
0	Americas	ATG	Antigua and Barbuda	2019	0.00
1	Western Pacific	FSM	Micronesia (Federated States of)	2019	0.16
2	South-East Asia	MDV	Maldives	2019	1.63
3	Western Pacific	KIR	Kiribati	2019	1.92
4	Eastern Mediterranean	EGY	Egypt	2019	10.10

In [22]:
```
# Convert 'year' to datetime (will default to YYYY-01-01 format)
road_death['year'] = pd.to_datetime(road_death['year'], format='%Y')

# Set 'year' as the index
road_death = road_death.set_index('year')
```

In [23]:
```
road_death.head()
```

Out[23]:

year	continent	code	country	road traffic death rate
2019-01-01	Americas	ATG	Antigua and Barbuda	0.00
2019-01-01	Western Pacific	FSM	Micronesia (Federated States of)	0.16
2019-01-01	South-East Asia	MDV	Maldives	1.63
2019-01-01	Western Pacific	KIR	Kiribati	1.92
2019-01-01	Eastern Mediterranean	EGY	Egypt	10.10

Research Questions

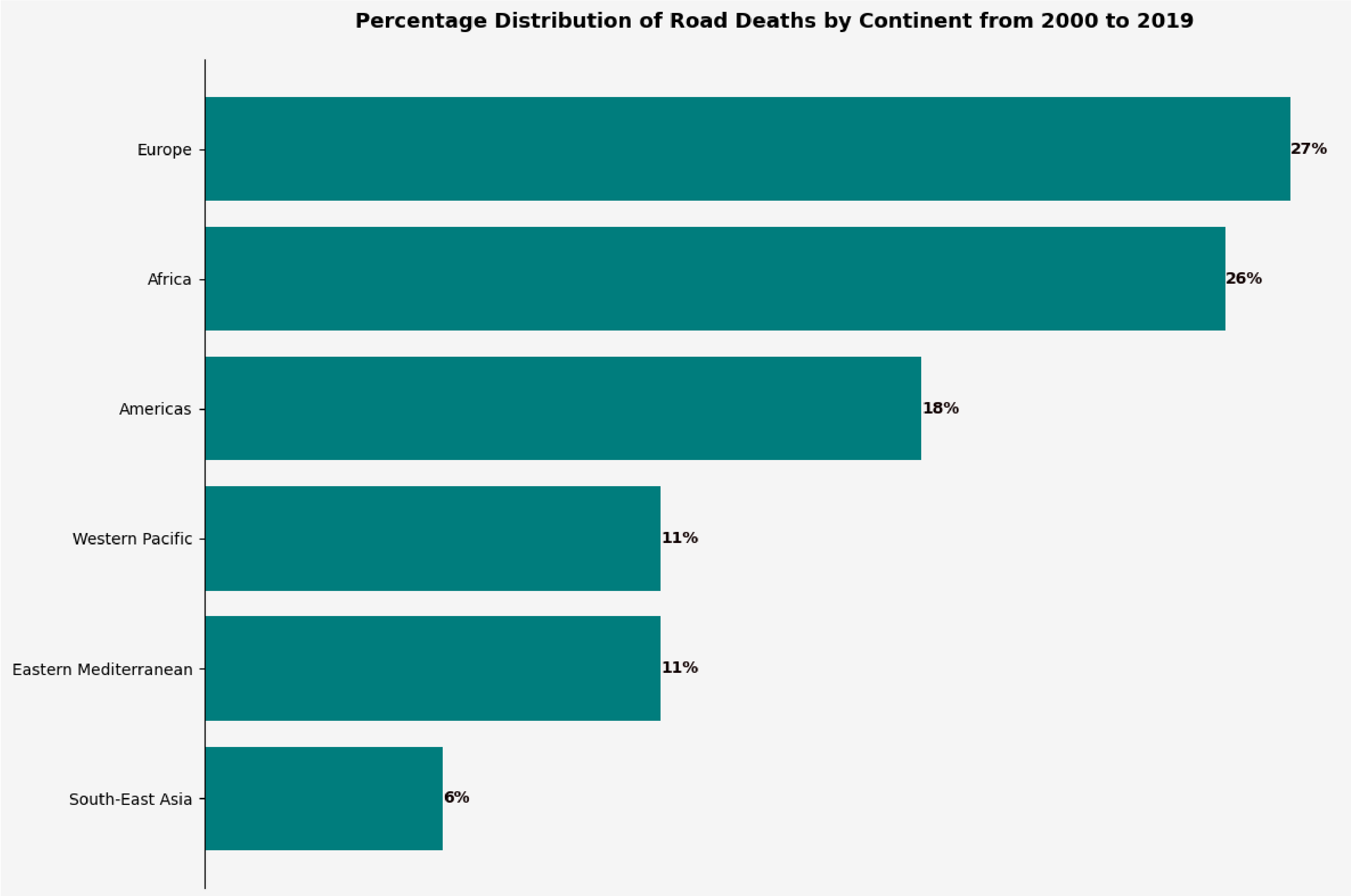
1. How do road traffic death rates vary across continents and African countries, and where does Kenya rank globally and regionally over recent years?

Africa status report on road safety 2025 which can be accessed in https://sdglocalaction.org/africaroadsafety-2025/

In 2025, Africa has the highest road traffic death rate globally, despite having a small percentage of the world's vehicles. Kenya's road traffic death rate is also high, both regionally and globally, with a rate of 28.2 deaths per 100,000 people. This puts Kenya among the most dangerous places to drive in the world. In Africa, Kenya ranks sixth in terms of road traffic fatalities.

In [24]:
```
# Calculate the percentage count of each continent
continent_counts = road_death['continent'].value_counts(normalize=True) * 100

# Create a new figure with specified size
plt.figure(figsize=(12, 8))

# Create a bar plot with continent names on y-axis and percentages on x-axis
bars = plt.barh(continent_counts.index,
                continent_counts.values,
                color = 'teal'
               )
```

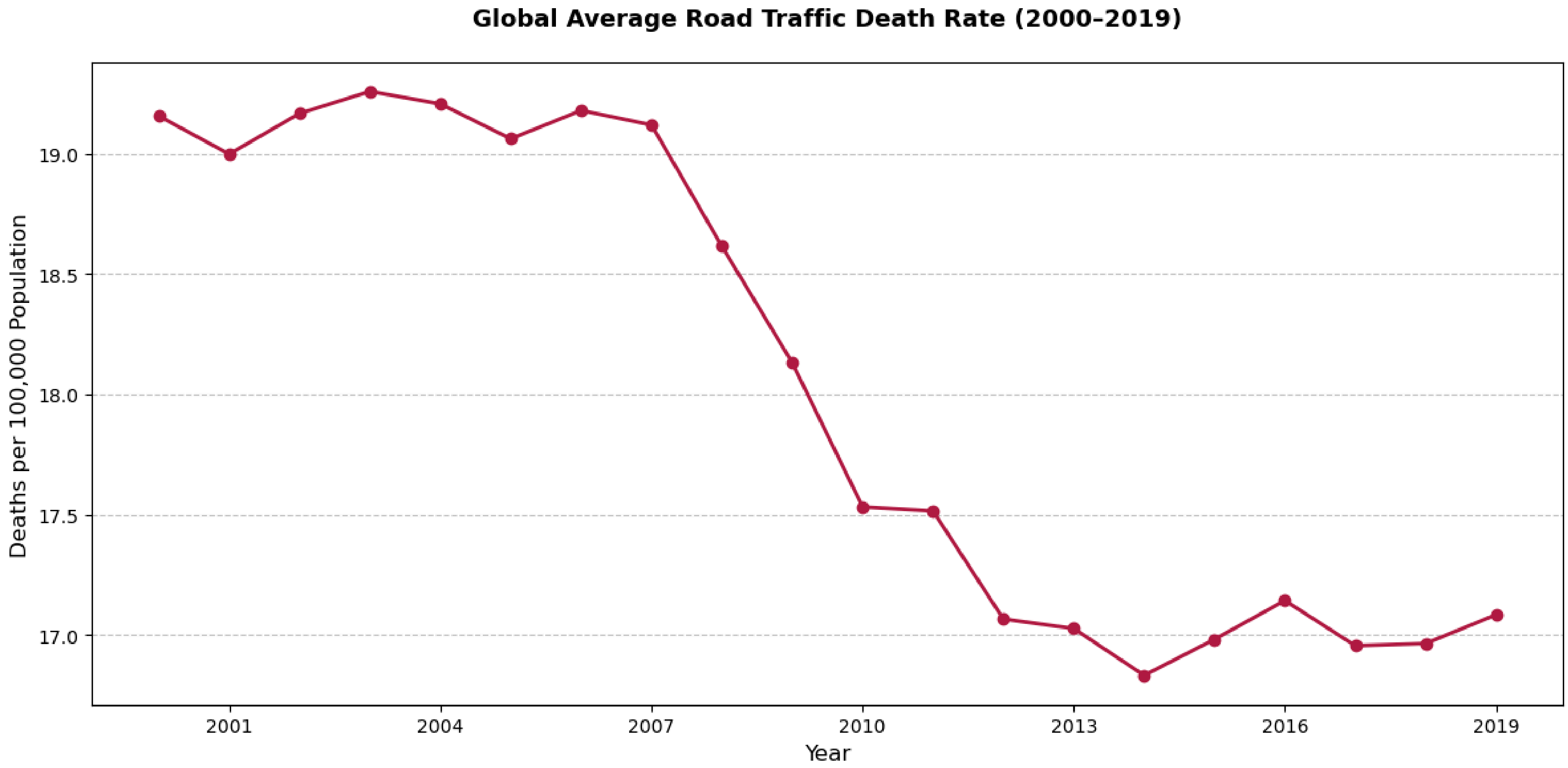
```
# Remove x-axis as instructed
plt.gca().xaxis.set_visible(False)

# Add data labels as percentages inside the bars
for bar in bars:
    width = bar.get_width()
    plt.text(width,                          # x position (right end of bar)
             bar.get_y() + bar.get_height()/2,  # y position (center of bar)
             f'{width:.0f}%',                # text (percentage with 0 decimal)
             ha='left',                      # horizontal alignment
             va='center',                    # vertical alignment
             color='#0f0101',                  # white text for visibility
             fontweight='bold')              # bold text

# Remove spines (top, right, and bottom as x-axis is removed)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)

# Add a title
plt.title('Percentage Distribution of Road Deaths by Continent from 2000 to 2019', pad=20, fontsize=13, weight='bold')

# Set background colors (ADD THESE LINES)
plt.gca().set_facecolor('#f5f5f5')  # Light gray axes background
plt.gcf().set_facecolor('#f5f5f5')  # Light gray axes background

# Invert y-axis to show highest at top
plt.gca().invert_yaxis()

# Adjust layout to prevent clipping
plt.tight_layout()

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



In [25]:
```
road_death.index
```

Out[25]:
```
DatetimeIndex(['2019-01-01', '2019-01-01', '2019-01-01', '2019-01-01',
               '2019-01-01', '2019-01-01', '2019-01-01', '2019-01-01',
               '2019-01-01', '2019-01-01',
               ...
               '2000-01-01', '2000-01-01', '2000-01-01', '2000-01-01',
               '2000-01-01', '2000-01-01', '2000-01-01', '2000-01-01',
               '2000-01-01', '2000-01-01'],
              dtype='datetime64[ns]', name='year', length=3660, freq=None)
```

In [26]:
```
# Extract year from the DatetimeIndex and group by it
yearly_avg = road_death.groupby(road_death.index.year)['road traffic death rate'].mean()

# Plotting
plt.figure(figsize=(12, 6))
plt.plot(yearly_avg.index, yearly_avg.values,
         marker='o', linestyle='-', color='#B31942', linewidth=2, label='Death Rate')

# Formatting
plt.title('Global Average Road Traffic Death Rate (2000–2019)', pad=20, fontsize=13, weight='bold')
plt.xlabel('Year', fontsize=12)
plt.ylabel('Deaths per 100,000 Population', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show all years as integer ticks
plt.gca().xaxis.set_major_locator(plt.MaxNLocator(integer=True))

plt.tight_layout()
plt.show()
```



Analyzing the Road accidents and incidents data (Nairobi, Kenya) by worldbank

In [27]:
```
# load the road death data by Kamau Munyori in kaggle
Nairobi_Road_Crashes = pd.read_csv(r"Data\Nairobi-Road-crashes.csv")
Nairobi_Road_Crashes.head()
```

Out[27]:

	crash_id	crash_datetime	crash_date	latitude	longitude	n_crash_reports	contains_fatality_words	contains_pedestrian_words	contains
0	1	06/06/2018 20:39	06/06/2018	-1.263030	36.764374	1	0	0	
1	2	17/08/2018 06:15	17/08/2018	-0.829710	37.037820	1	1	0	
2	3	25/05/2018 17:51	25/05/2018	-1.125301	37.003297	1	0	0	
3	4	25/05/2018 18:11	25/05/2018	-1.740958	37.129025	1	0	0	
4	5	25/05/2018 21:59	25/05/2018	-1.259392	36.842321	1	1	0	

In [28]:

```
Nairobi_Road_Crashes.dtypes
```

Out[28]:
```
crash_id                        int64
crash_datetime                 object
crash_date                     object
latitude                      float64
longitude                     float64
n_crash_reports                 int64
contains_fatality_words         int64
contains_pedestrian_words       int64
contains_matatu_words           int64
contains_motorcycle_words       int64
location                       object
dtype: object
```

In [29]:
```
Nairobi_Road_Crashes[['crash_datetime', 'crash_date']].head()
```

Out[29]:

	crash_datetime	crash_date
0	06/06/2018 20:39	06/06/2018
1	17/08/2018 06:15	17/08/2018
2	25/05/2018 17:51	25/05/2018
3	25/05/2018 18:11	25/05/2018
4	25/05/2018 21:59	25/05/2018

In [30]:
```
# Convert crash_datetime (contains both date and time)
Nairobi_Road_Crashes['crash_datetime'] = pd.to_datetime(
    Nairobi_Road_Crashes['crash_datetime'],
    format='%d/%m/%Y %H:%M',  # Matches "06/06/2018 20:39" format
    errors='coerce'  # Converts invalid entries to NaT
)

# Convert crash_date (date only)
Nairobi_Road_Crashes['crash_date'] = pd.to_datetime(
    Nairobi_Road_Crashes['crash_date'],
    format='%d/%m/%Y',  # Matches "06/06/2018" format
    errors='coerce'
)
```

```
# Verify the changes
Nairobi_Road_Crashes.head()
```

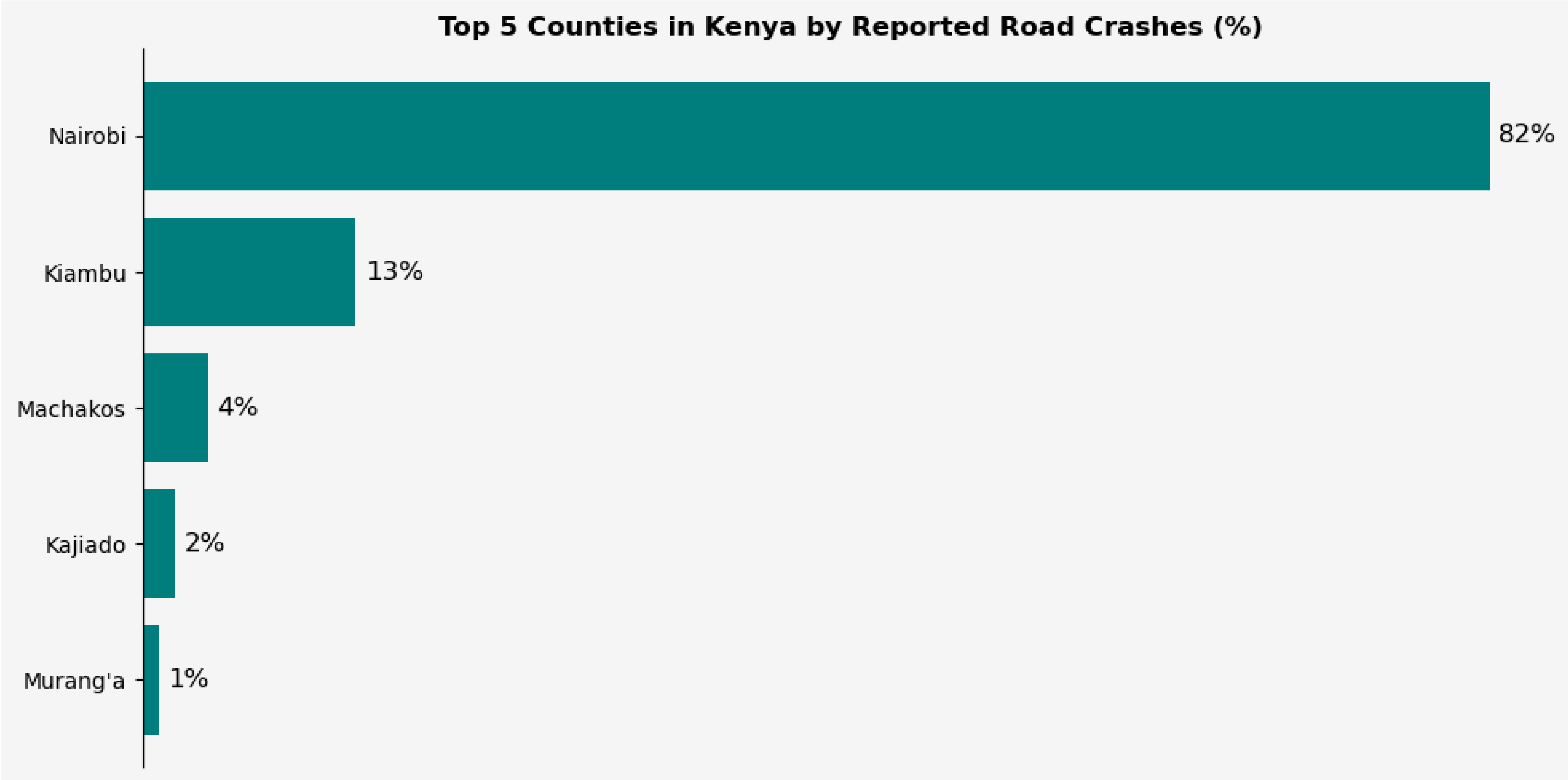
Out[30]:

	crash_id	crash_datetime	crash_date	latitude	longitude	n_crash_reports	contains_fatality_words	contains_pedestrian_words	contains
0	1	2018-06-06 20:39:00	2018-06-06	-1.263030	36.764374	1	0	0	
1	2	2018-08-17 06:15:00	2018-08-17	-0.829710	37.037820	1	1	0	
2	3	2018-05-25 17:51:00	2018-05-25	-1.125301	37.003297	1	0	0	
3	4	2018-05-25 18:11:00	2018-05-25	-1.740958	37.129025	1	0	0	
4	5	2018-05-25 21:59:00	2018-05-25	-1.259392	36.842321	1	1	0	

2. How has the frequency and severity of road crashes evolved over time across Kenyan counties, and are specific regions becoming increasingly dangerous?

Road traffic accidents in Kenya have shown both an increasing trend in frequency and severity over time, with some regions experiencing more pronounced increases. While there have been recent reductions in accident frequency, some counties still face a high burden of road accidents and injuries.

In [31]:
```
from extract_county import extract_county

# Apply the function to your DataFrame
Nairobi_Road_Crashes['county'] = Nairobi_Road_Crashes['location'].apply(extract_county)
county_crashes = pd.DataFrame(Nairobi_Road_Crashes['county'].value_counts())
county_crashes.head(5)
# Step 1: Get the top 5 counties by count
county_counts = Nairobi_Road_Crashes['county'].value_counts().nlargest(5)
total = county_counts.sum()

# Step 2: Convert to percentages and round up
county_percentages = np.ceil((county_counts / total) * 100)

# Step 3: Plot
plt.figure(figsize=(10, 5), facecolor="#f5f5f5")
bars = plt.barh(county_percentages.index, county_percentages.values, color='teal')

# Step 4: Add data labels
for bar in bars:
    width = bar.get_width()
    plt.text(width + 0.5, bar.get_y() + bar.get_height() / 2,
             f"{int(width)}%", va='center', fontsize=12)

# Step 5: Formatting
ax = plt.gca()
ax.set_facecolor("#f5f5f5")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.invert_yaxis()  # Largest on top
ax.set_xticks([])  # Remove x-axis ticks

plt.title("Top 5 Counties in Kenya by Reported Road Crashes (%)", fontsize=12, weight='bold')
plt.tight_layout()
plt.show()
```

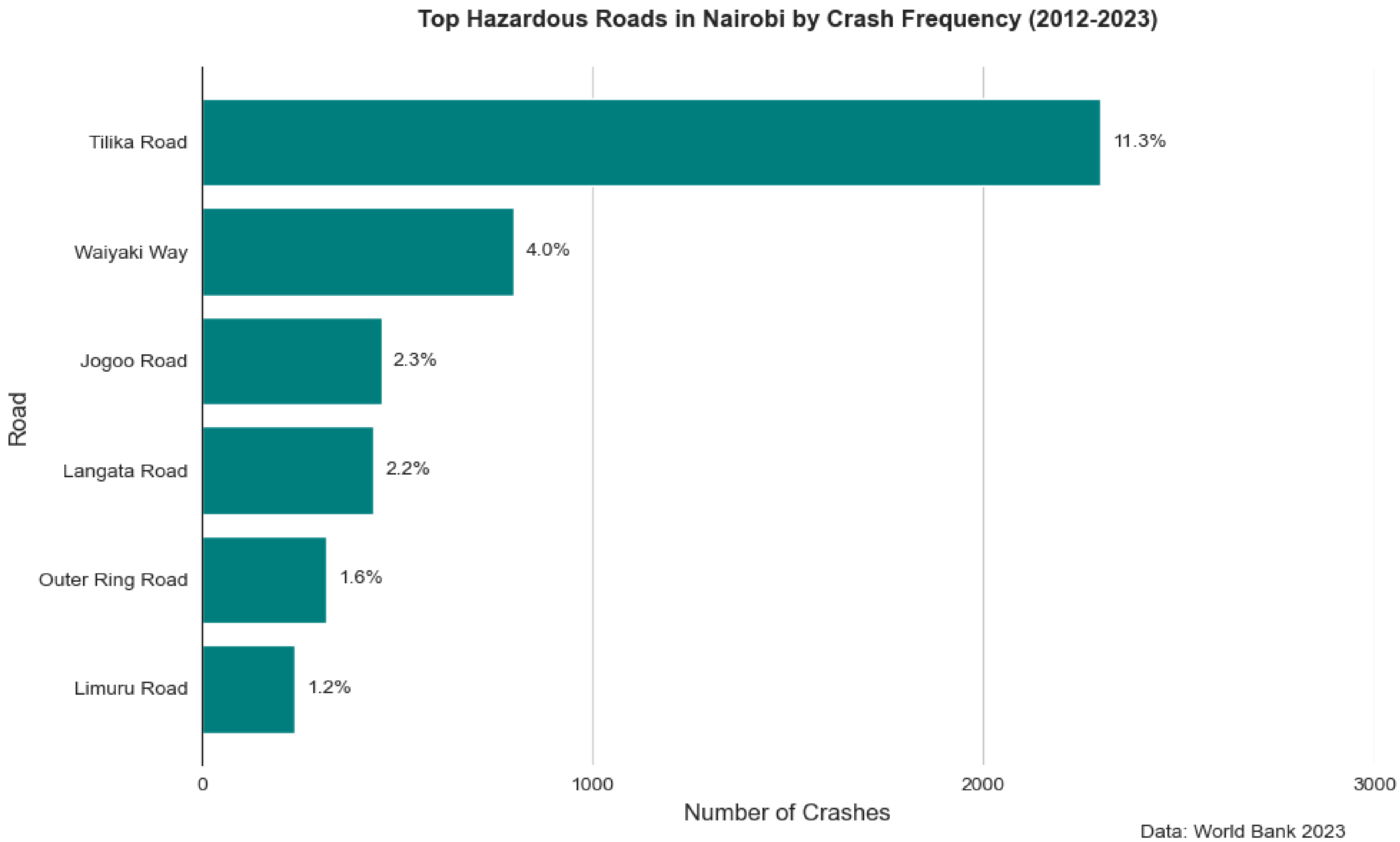


3. What are the spatiotemporal patterns of road crashes within Nairobi and its environs?

Nairobi accounts for about a quarter of national crash deaths . Crashes are highly concentrated: one analysis found ~10 locations (clusters) accounted for 10% of Nairobi crashes, and ~100 clusters for 50% . Major Nairobi corridors (Thika Superhighway, Airport North Road, Eastern Bypass, Jogoo Road, Mombasa Road) are repeatedly identified as blackspots. Temporal spikes occur on weekends and evenings. A study of Nairobi hospital data (2011) saw ~42% of pedestrian crashes on weekends (25.5% on Saturdays, 16.7% Sundays). NTSA confirms national peaks: 2024 had the most fatalities on Saturdays (855), and evening hours (7–8 pm) were riskiest. Seasonally, December is deadliest (466 deaths in 2024). Pedestrians dominate Nairobi crash severity: they made up ~59% of severe injuries in 2011 and about 65–74% of fatalities in 2015–16. Most were struck while crossing roads (70%). Nairobi’s hotspots coincide with high pedestrian exposure and heavy traffic zones, especially at night and on weekends.

In [57]:
```
# Data preparation
data = {
    'Road': ["Tilika Road", "Waiyaki Way", "Jogoo Road", "Langata Road",
             "Outer Ring Road", "Limuru Road"],
    'Percentage': [11.3, 4.0, 2.3, 2.2, 1.6, 1.2],
    'Crashes': [2300, 800, 460, 440, 320, 240]
}

# Create DataFrame and sort by Crashes (descending)
df = pd.DataFrame(data).sort_values('Crashes', ascending=True)  # Note: ascending=True for horizontal bars

# Set style
sns.set_style("white")
plt.figure(figsize=(10, 6))
ax = plt.gca()

# Create horizontal bar plot
bars = ax.barh(df['Road'], df['Crashes'], color='teal')

# Add percentage labels at the end of each bar
for bar, percentage in zip(bars, df['Percentage']):
    width = bar.get_width()
    ax.text(width + 30, bar.get_y() + bar.get_height()/2,
            f'{percentage}%',
            va='center', fontsize=10)

# Customize chart
plt.title('Top Hazardous Roads in Nairobi by Crash Frequency (2012-2023)',
          fontsize=12, fontweight='bold', pad=20)
plt.xlabel('Number of Crashes', fontsize=12)
plt.ylabel('Road', fontsize=12)

# Set x-axis ticks and enable only horizontal grid lines (vertical in horizontal bar chart)
plt.xticks([0, 1000, 2000, 3000])
ax.xaxis.grid(True)   # Enable x-axis (horizontal) grid lines
ax.yaxis.grid(False)  # Disable y-axis (vertical) grid lines

# Remove unnecessary spines
for spine in ['top', 'right']:
```

```
    ax.spines[spine].set_visible(False)

ax.spines['bottom'].set_visible(False)

# Add data source at bottom right
plt.figtext(0.95, 0.01, 'Data: World Bank 2023', ha='right', fontsize=10)

# Adjust layout
plt.tight_layout()
plt.show()
```

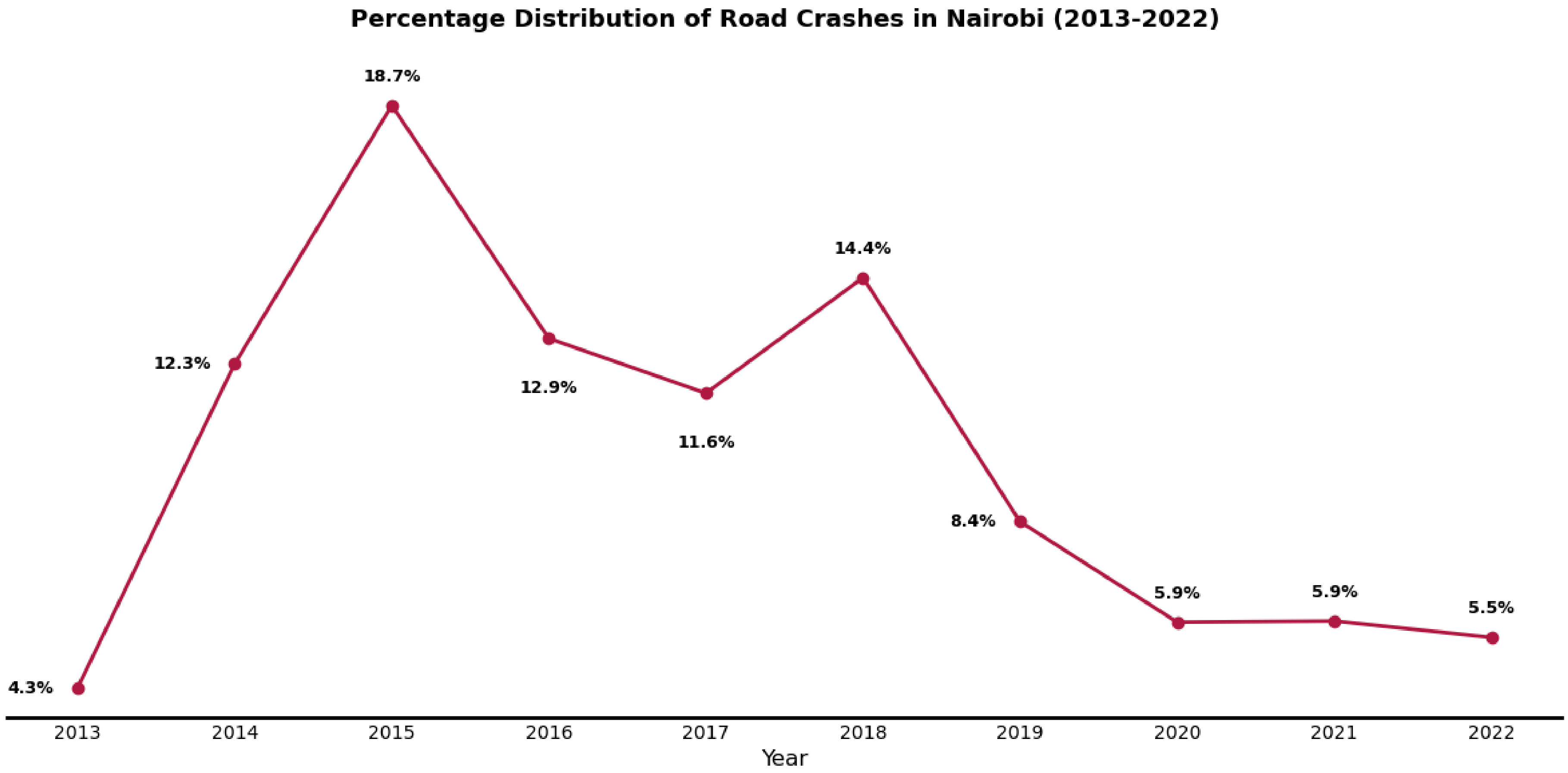


4. How has the distribution of road crashes in Nairobi changed across years, and are there identifiable demographic patterns (e.g., age, gender, victim type)?

In Nairobi, road crash distributions have shown changes over time, with identifiable demographic patterns. Data indicates a higher frequency of crashes on specific roads like Thika Road, Waiyaki Way, and Mombasa Road. Demographically, road crashes tend to affect men, with a significant portion occurring during their most reproductive years (15-64 years). Additionally, studies suggest that men and individuals within the most economically productive age groups are more likely to be involved in crashes.

In [32]:
```
# Extract year from crash_date
Nairobi_Road_Crashes['year'] = Nairobi_Road_Crashes['crash_date'].dt.year

# Filter out 2012 and 2023 records
filtered_data = Nairobi_Road_Crashes[
    (Nairobi_Road_Crashes['year'] >= 2013) &
    (Nairobi_Road_Crashes['year'] <= 2022)
]

# Calculate yearly counts and convert to percentages (rounded to integers)
yearly_counts = filtered_data['year'].value_counts().sort_index()
percentages = ((yearly_counts / yearly_counts.sum()) * 100)

# Create the line plot
plt.figure(figsize=(12, 6))
plt.plot(percentages.index, percentages.values,
         marker='o', linestyle='-', color='#B31942', linewidth=2)

# Add data labels with custom positioning
for year, pct in zip(percentages.index, percentages.values):
    # Custom positioning based on year
    if year in [2013, 2014, 2019]:
        # Place to the left of the point
        plt.text(year-0.15, pct, f'{pct:.1f}%',
                 ha='right', va='center',
                 fontsize=9, fontweight='bold')
    elif year in [2016, 2017]:
        # Place below the point
        plt.text(year, pct-1, f'{pct:.1f}%',
                 ha='center', va='top',
                 fontsize=9, fontweight='bold')
    else:
        # Default position (above the point)
        plt.text(year, pct+0.5, f'{pct:.1f}%',
                 ha='center', va='bottom',
                 fontsize=9, fontweight='bold')
```

```python
# Formatting
title = plt.title('Percentage Distribution of Road Crashes in Nairobi (2013-2022)',
                 pad=20, fontsize=13,
                 loc='center',
                 y=1.02,
                 fontweight='bold')

plt.xlabel('Year', fontsize=12)

# Custom x-axis styling
ax = plt.gca()
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.yaxis.set_visible(False)
plt.grid(False)

# Enhanced x-axis line
ax.spines['bottom'].set_linewidth(2)  # Thicker baseline
ax.spines['bottom'].set_linestyle("-")  # continuous line
ax.spines['bottom'].set_color("black")   # Black color

# Set x-axis ticks for each year with equal spacing
plt.xticks(percentages.index)
ax.xaxis.set_minor_locator(plt.FixedLocator(percentages.index))
ax.tick_params(axis='x', which='both', length=0)
ax.set_xticklabels(percentages.index, ha='center')  # Center aligned labels

plt.tight_layout()
plt.show()
```



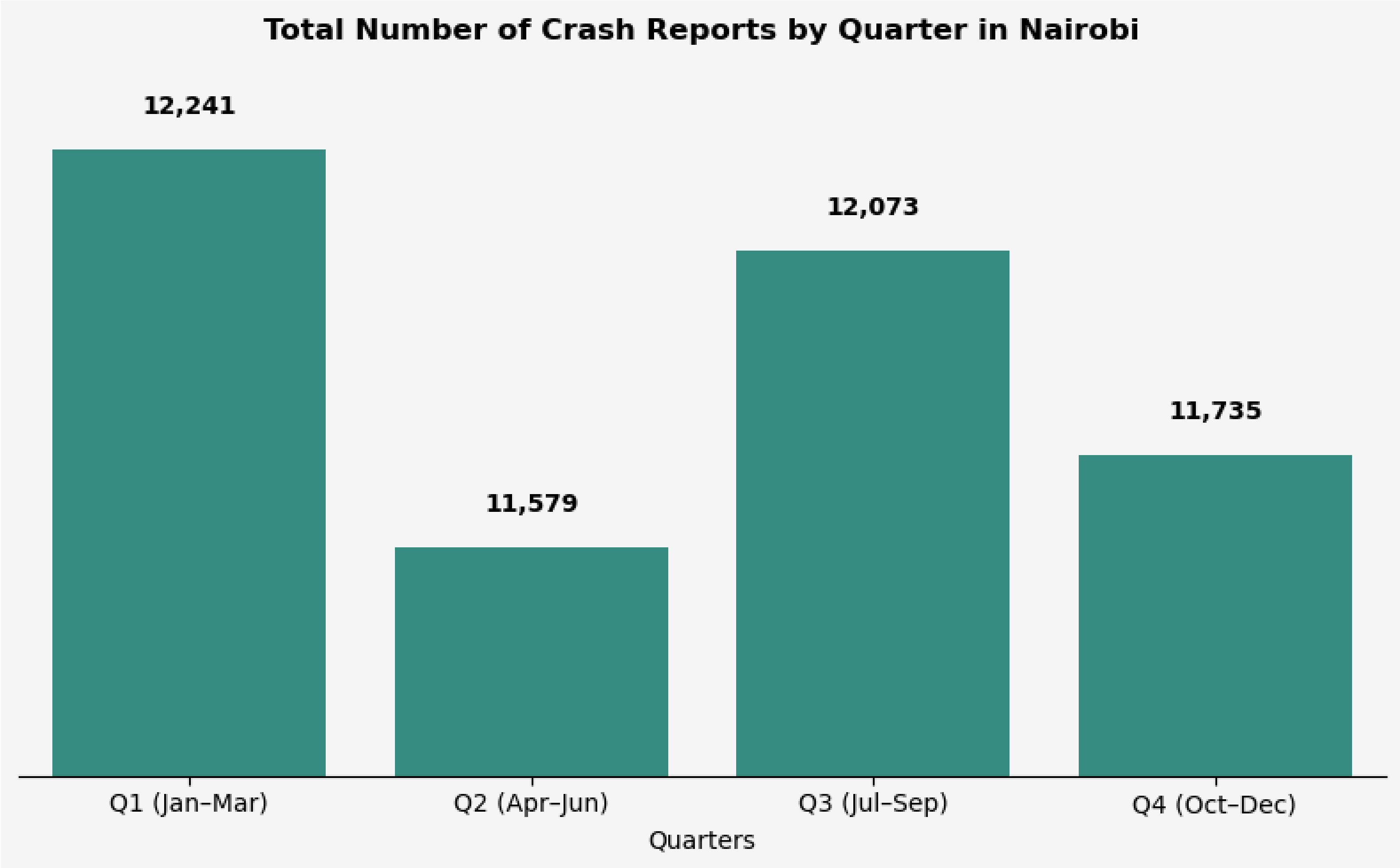
5. How do road crashes in Nairobi vary by month and quarter, and what seasonal trends can be observed?

Road accidents in Nairobi exhibit seasonal trends, with an increase during rainy seasons and festive periods. Monthly, the number of accidents can vary, with some months, like November, showing higher averages than others, like January. Quarter-wise, the fourth quarter often sees the highest average number of reported accidents.

In [33]:
```
# Step 1: Extract quarter and label
Nairobi_Road_Crashes['quarter'] = Nairobi_Road_Crashes['crash_datetime'].dt.quarter
Nairobi_Road_Crashes['quarter_label'] = Nairobi_Road_Crashes['quarter'].apply(lambda x: f'Q{x}')
```

```
# Step 2: Define detailed quarter labels
quarter_name_mapping = {
    'Q1': 'Q1 (Jan-Mar)',
    'Q2': 'Q2 (Apr-Jun)',
    'Q3': 'Q3 (Jul-Sep)',
    'Q4': 'Q4 (Oct-Dec)'
}
ordered_quarters = ['Q1', 'Q2', 'Q3', 'Q4']
Quarters = [quarter_name_mapping[q] for q in ordered_quarters]

# Step 3: Group by quarter and sum crash reports
total_crashes_by_quarter = (
    Nairobi_Road_Crashes
    .groupby('quarter_label')['n_crash_reports']
    .sum()
    .reindex(ordered_quarters)
    .reset_index()
)
total_crashes_by_quarter['Quarters'] = Quarters

# Step 4: Plot bar chart with specified modifications
plt.figure(figsize=(8, 5), facecolor='#f5f5f5')
constant_color = '#2a9d8f'  # Set your preferred color here

ax = sns.barplot(
    data=total_crashes_by_quarter,
    x='Quarters',
    y='n_crash_reports',
    color=constant_color
)

# Set background color
ax.set_facecolor('#f5f5f5')

# Remove gridlines and spines
ax.grid(False)
sns.despine(top=True, right=True, left=True)

# Hide y-axis
ax.set_ylabel("")
```

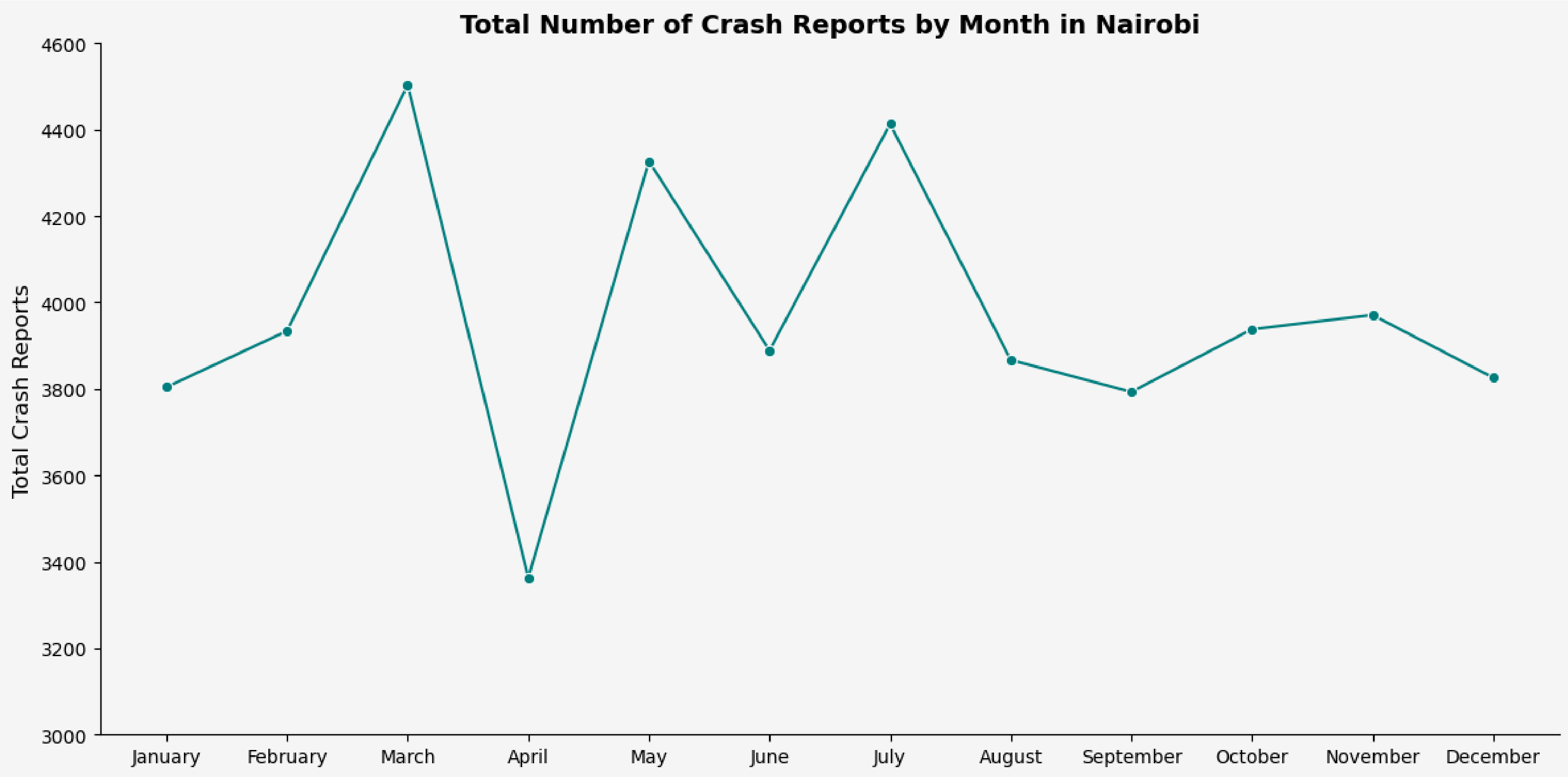
```
ax.set_yticks([])

# Set custom y-axis limits
ax.set_ylim(11200, 12400)

# Add data labels on top of each bar with thousands separator
for index, row in total_crashes_by_quarter.iterrows():
    ax.text(index, row['n_crash_reports'] + 50, f"{int(row['n_crash_reports']):,}",
            ha='center', va='bottom', fontsize=10, weight='bold')

# Title
ax.set_title("Total Number of Crash Reports by Quarter in Nairobi", fontsize=12, weight='bold')

# Layout adjustment
plt.tight_layout()
plt.show()
```



In [38]:
```
# Step 1: Extract month name
Nairobi_Road_Crashes['month'] = Nairobi_Road_Crashes['crash_datetime'].dt.month_name()

# Step 2: Define the order of months
ordered_months = [
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
]

# Step 3: Group by month and sum crash reports
total_crashes_by_month = (
```

```
    Nairobi_Road_Crashes
    .groupby('month')['n_crash_reports']
    .sum()
    .reindex(ordered_months)
    .reset_index()
)

# Step 4: Plot
plt.figure(figsize=(12, 6), facecolor='#f5f5f5')
ax = sns.lineplot(data=total_crashes_by_month, x='month', y='n_crash_reports',
                  marker='o', color='teal')

# Set background color
ax.set_facecolor('#f5f5f5')

# Remove gridlines
ax.grid(False)

# Set custom y-axis limits
ax.set_ylim(3000, 4600)

# Remove top and right spines
sns.despine(top=True, right=True)

# Set labels and title
ax.set_title("Total Number of Crash Reports by Month in Nairobi", fontsize=14, weight='bold')
ax.set_xlabel("")  # No need to label x-axis explicitly
ax.set_ylabel("Total Crash Reports", fontsize=12)

# Layout adjustment
plt.tight_layout()
plt.show()
```

```
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



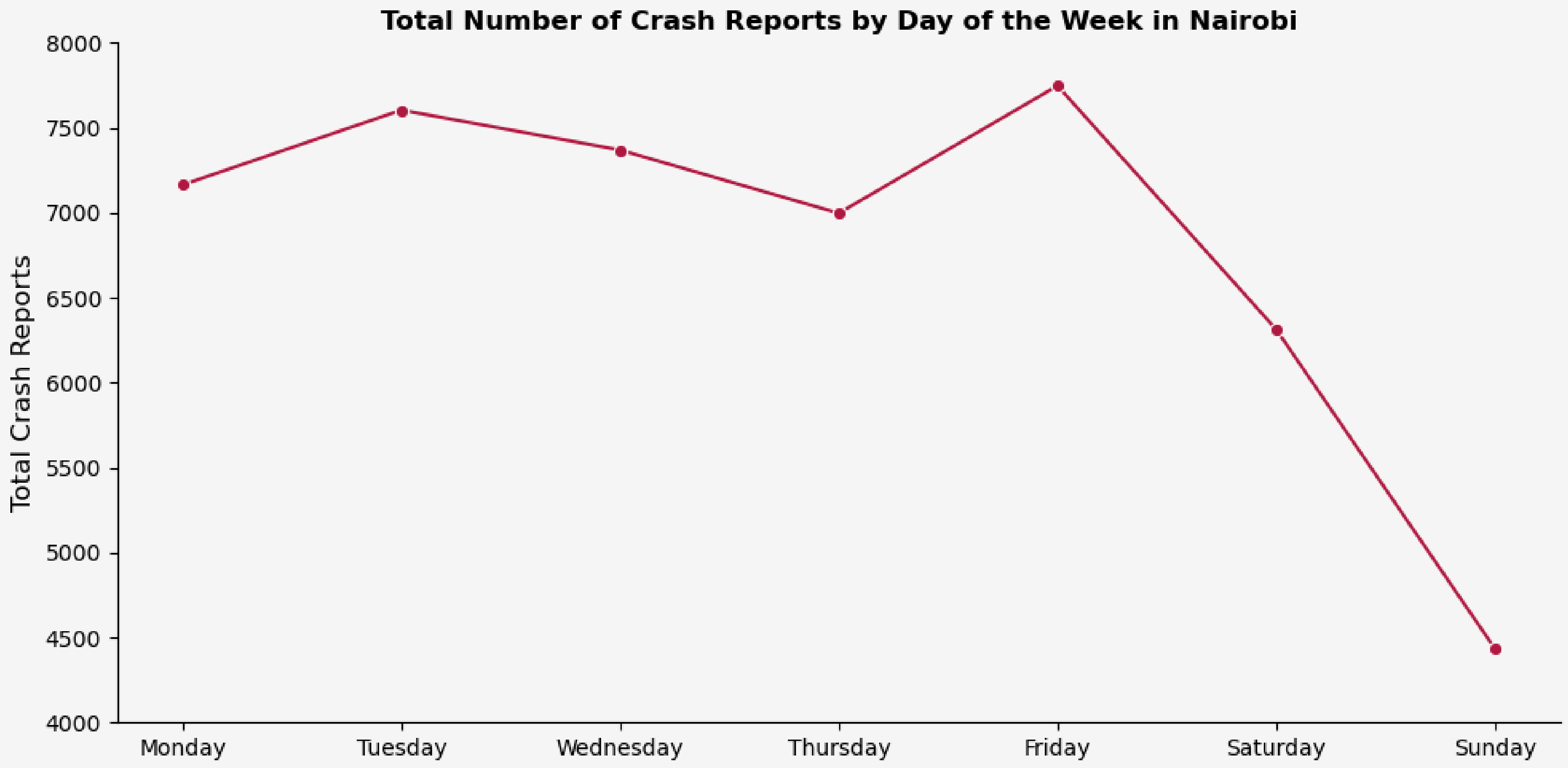
6. How do crash characteristics differ by day of the week, and what are the weekday vs weekend variations in fatality and pedestrian involvement?

In Nairobi, crash characteristics and pedestrian involvement vary significantly between weekdays and weekends. Weekends see more fatal crashes, particularly on Saturdays, and a higher proportion of pedestrian crashes. While weekday crashes are more likely to occur during congested periods, weekend crashes tend to happen in free-flowing traffic.

In [35]:
```
# Step 1: Extract day of the week
Nairobi_Road_Crashes['day_of_week'] = Nairobi_Road_Crashes['crash_datetime'].dt.day_name()

# Step 2: Define custom order of days
```

```
ordered_days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

# Step 3: Group by day and sum crash reports
total_crashes_by_day = (
    Nairobi_Road_Crashes
    .groupby('day_of_week')['n_crash_reports']
    .sum()
    .reindex(ordered_days)
    .reset_index()
)

# Step 4: Plot
plt.figure(figsize=(10, 5), facecolor='#f5f5f5')
ax = sns.lineplot(data=total_crashes_by_day, x='day_of_week', y='n_crash_reports',
                  marker='o', color='#B31942')

# Set background color
ax.set_facecolor('#f5f5f5')

# Set custom y-axis limits
ax.set_ylim(4000, 8000)

# Remove gridlines
ax.grid(False)

# Remove top and right spines
sns.despine(top=True, right=True)

# Set labels and title
ax.set_title("Total Number of Crash Reports by Day of the Week in Nairobi", fontsize=12, weight='bold')
ax.set_xlabel("")  # No need to label x-axis explicitly
ax.set_ylabel("Total Crash Reports", fontsize=12)

# Layout adjustment
plt.tight_layout()
plt.show()
```

```
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



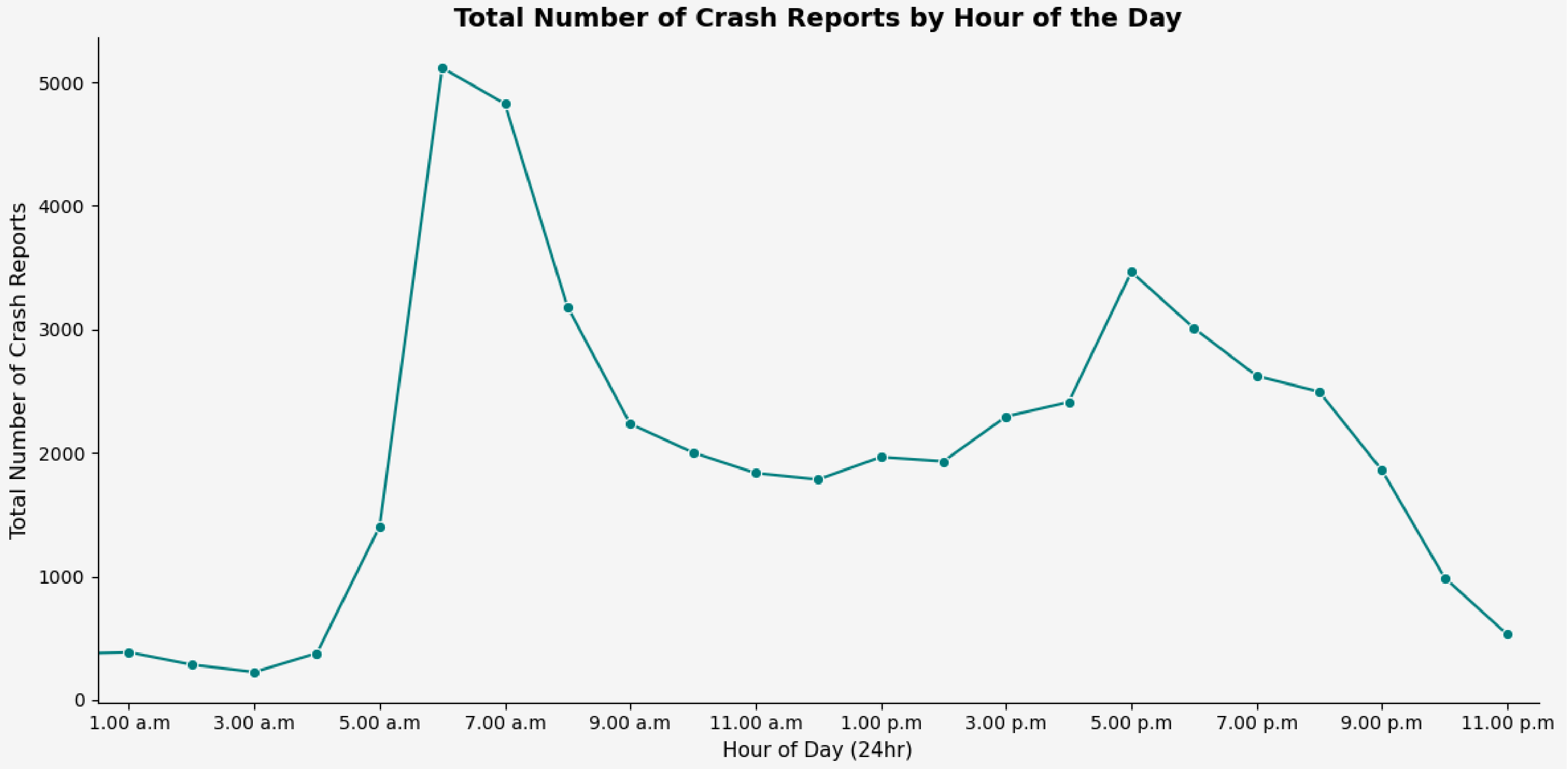
7. How does crash frequency and severity vary by time of day, and are specific hours (e.g., peak commuting times) associated with higher crash risk?

Crash frequency and severity tend to vary throughout the day, with peak commuting times and nighttime driving often associated with higher crash risk. Specific hours like morning and evening rush hours (e.g., 6-9 AM and 4-7 PM) are frequently cited as accident-prone due to increased traffic volume and potential for congestion, while nighttime driving is linked to reduced visibility and increased risky behaviors, leading to higher fatality rates.

In [36]:
```
# Step 1: Extract hour
Nairobi_Road_Crashes['hour'] = Nairobi_Road_Crashes['crash_datetime'].dt.hour

# Step 2: Group by hour and average
# avg_crashes_by_hour = Nairobi_Road_Crashes.groupby('hour')['n_crash_reports'].mean().reset_index()
total_crashes_by_hour = Nairobi_Road_Crashes.groupby('hour')['n_crash_reports'].sum().reset_index()


# Step 3: Define custom x-tick labels
custom_ticks = list(range(1, 24, 2))  # 1, 3, 5, ..., 23
custom_labels = [f"{h}.00 a.m" if h < 12 else (f"{h-12}.00 p.m" if h > 12 else "12.00 p.m") for h in custom_ticks]

# Step 4: Plot
plt.figure(figsize=(12, 6), facecolor='#f5f5f5')
# ax = sns.lineplot(data=avg_crashes_by_hour, x='hour', y='n_crash_reports', marker='o', color='teal')
ax = sns.lineplot(data=total_crashes_by_hour, x='hour', y='n_crash_reports', marker='o', color='teal')

# Set background color
ax.set_facecolor('#f5f5f5')

# Remove gridlines
ax.grid(False)

# Set custom x-ticks and labels
ax.set_xticks(custom_ticks)
ax.set_xticklabels(custom_labels)

# Add padding to x-axis (left and right)
ax.set_xlim(0.5, 23.5)

# Remove top and right spines
sns.despine(top=True, right=True)

# Set labels and title
ax.set_title("Total Number of Crash Reports by Hour of the Day", fontsize=14, weight='bold')
ax.set_xlabel("Hour of Day (24hr)", fontsize=11)
ax.set_ylabel("Total Number of Crash Reports", fontsize=12)
```

```
# Layout adjustment
plt.tight_layout()
plt.show()
```

```
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

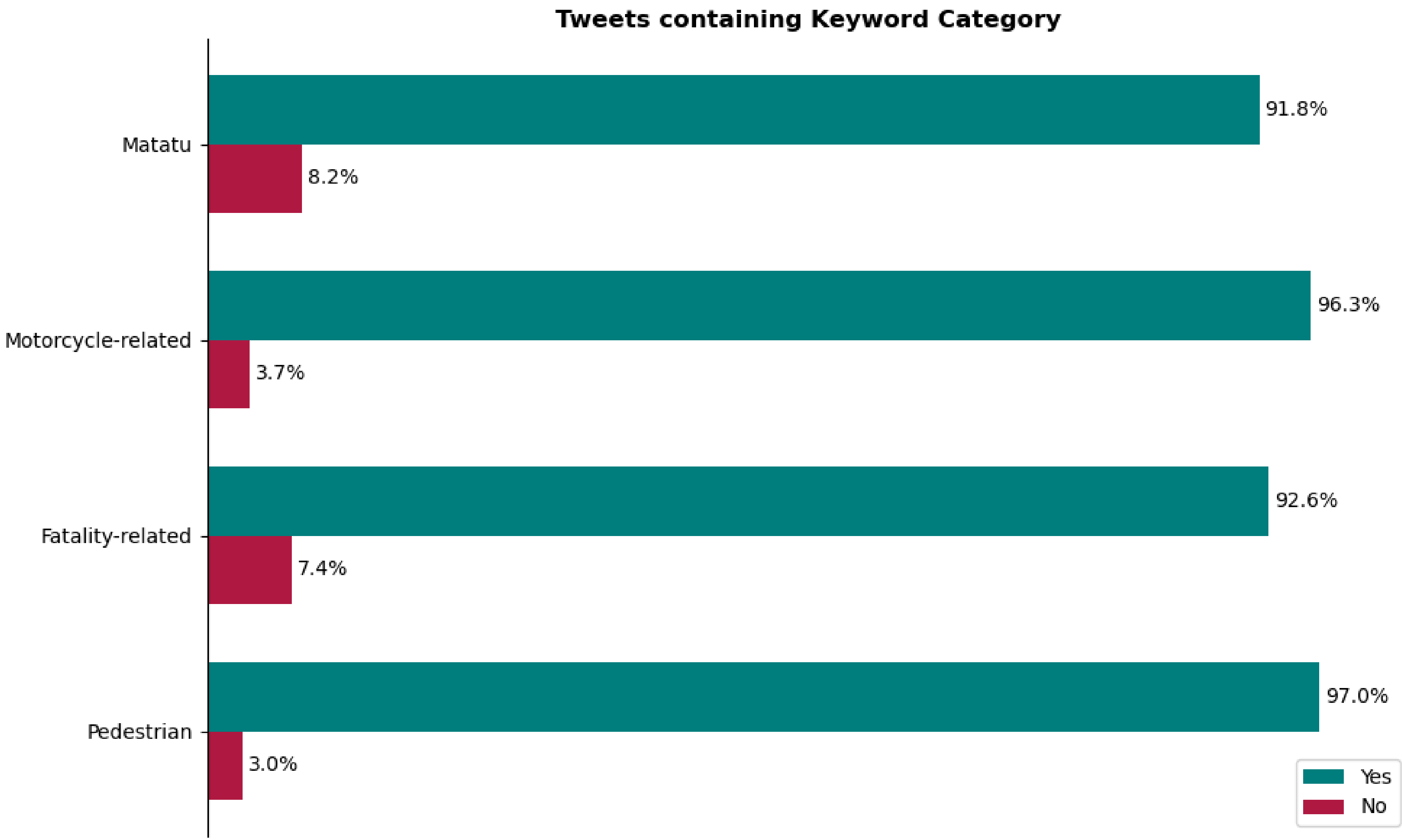


8. To what extent do matatus and motorcycles feature in road crash reports, and how does their presence relate to severity across time (hour, day, month)?

In road crash reports, matatus are significantly more involved than motorcycles. Matatus are involved in over 50% of crashes, while motorcycles account for around 25%. Severity and time (hour, day, month) likely influence the specific circumstances of these crashes. For example, excessive speeding, loss of control, and overcorrecting during turns are common in motorcycle crashes. Speeding is also a major factor in matatu crashes

In [37]:
```
# Labels for categories
variables = ["Pedestrian", "Fatality-related", "Motorcycle-related", "Matatu"]

# Data: [Yes, No] percentages for each variable
data = [
    [97.0, 3.0],     # Pedestrian
    [92.6, 7.4],     # Fatality-related
    [96.3, 3.7],     # Motorcycle-related
    [91.8, 8.2],     # Matatu
]

# Set up bar positions
y = np.arange(len(variables))  # base y positions
bar_height = 0.35

fig, ax = plt.subplots(figsize=(10, 6), facecolor='white')

# Colors
colors = {'Yes': 'teal', 'No': '#B31942'}

# Draw bars with Yes on top
for i, (yes, no) in enumerate(data):
    # Yes bar (upper bar visually)
    ax.barh(y[i] + bar_height / 2, yes, height=bar_height, color=colors['Yes'], label='Yes' if i == 0 else "")
    # No bar (lower bar visually)
    ax.barh(y[i] - bar_height / 2, no, height=bar_height, color=colors['No'], label='No' if i == 0 else "")

    # Labels
    ax.text(yes + 0.5, y[i] + bar_height / 2, f'{yes:.1f}%', va='center', color='black')
    ax.text(no + 0.5, y[i] - bar_height / 2, f'{no:.1f}%', va='center', color='black')
```

```
# Remove spines (top, right, and bottom as x-axis is removed)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)

# Y-axis
ax.set_yticks(y)
ax.set_yticklabels(variables)

# Clean visuals
ax.grid(False)
ax.set_xticks([])
ax.set_xlim(0, 105)
ax.set_facecolor('white')

# Title and legend
ax.set_title('Tweets containing Keyword Category', fontsize=12, weight='bold')
ax.legend(loc='lower right')

plt.tight_layout()
plt.show()
```

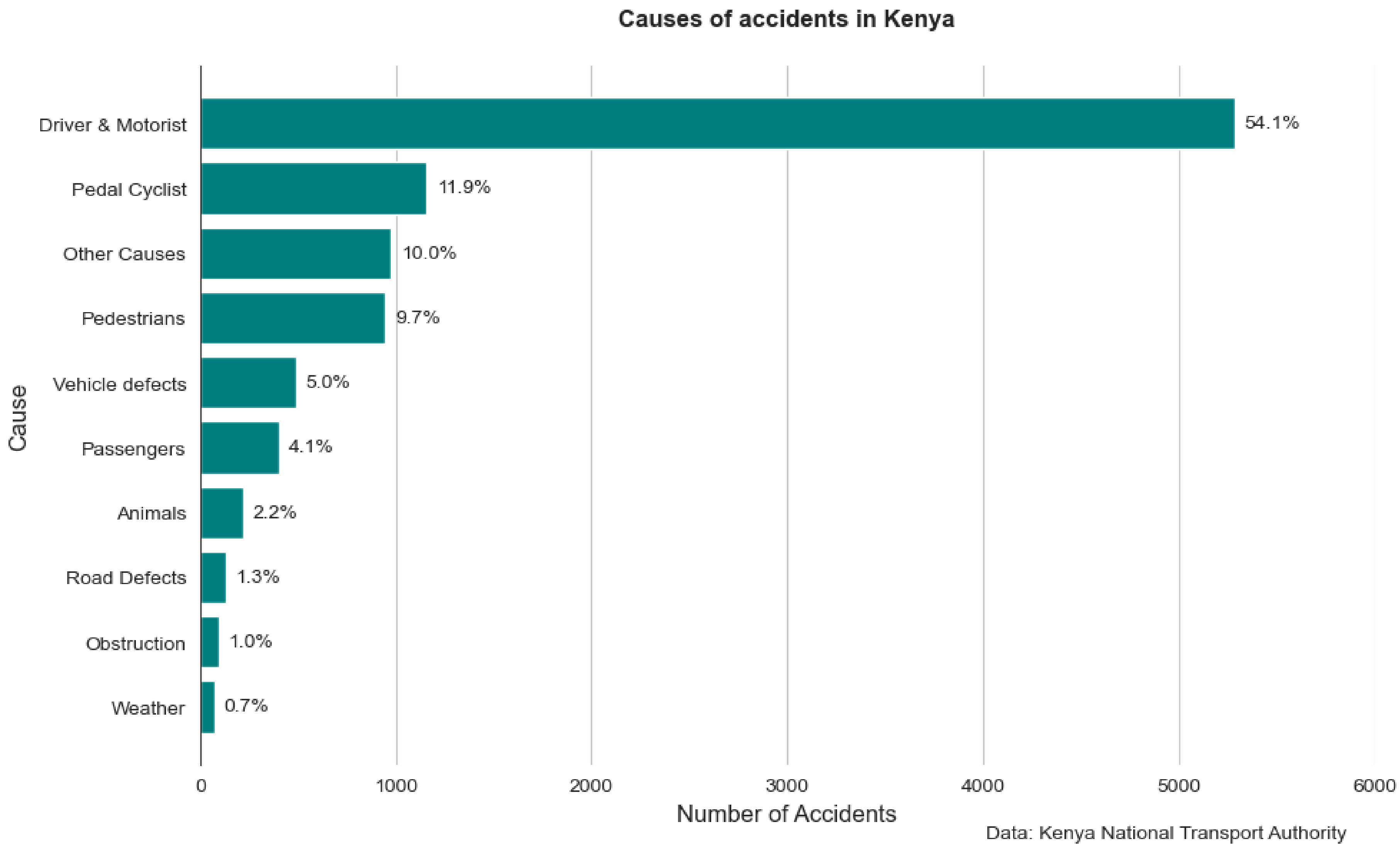


9. What are the primary causes of road traffic accidents globally, in Kenya, and specifically in Nairobi over time?

Globally, the primary causes of road traffic accidents are human error, including speeding, drunk driving, distracted driving, and reckless behavior. In Kenya and specifically in Nairobi, the same factors are dominant, with a significant percentage of accidents attributed to speeding and dangerous overtaking. Over time, Nairobi's road accident trends have been influenced by increased urbanization and traffic congestion, with factors like poor road infrastructure and inadequate traffic management also playing a role.

In [64]:
```
# Data preparation
causes = [
    "Driver & Motorist", "Pedal Cyclist", "Pedestrians", "Passengers",
    "Animals", "Obstruction", "Vehicle defects", "Road Defects",
    "Weather", "Other Causes"
]

values = [5284, 1159, 947, 401, 217, 94, 489, 131, 73, 976]  # Values from the image

# Calculate percentages
total = sum(values)
percentages = [(value/total)*100 for value in values]

# Create DataFrame and sort by value (descending)
df = pd.DataFrame({
    'Cause': causes,
    'Value': values,
    'Percentage': percentages
}).sort_values('Value', ascending=True)  # ascending=True for horizontal bars

# Set style
sns.set_style("white")
plt.figure(figsize=(10, 6))
ax = plt.gca()

# Create horizontal bar plot with teal color
bars = ax.barh(df['Cause'], df['Value'], color='teal')

# Add percentage labels at the end of each bar
for bar, percentage in zip(bars, df['Percentage']):
    width = bar.get_width()
    ax.text(width + 50, bar.get_y() + bar.get_height()/2,
            f'{percentage:.1f}%',  # Format to 1 decimal place
            va='center', fontsize=10)

# Customize chart
plt.title('Causes of accidents in Kenya',
```

```
          fontsize=12, fontweight='bold', pad=20)
plt.xlabel('Number of Accidents', fontsize=12)
plt.ylabel('Cause', fontsize=12)

# Set x-axis ticks and enable only horizontal grid lines
plt.xticks(np.arange(0, 6001, 1000))  # From 0 to 10000 in steps of 1000
ax.xaxis.grid(True)   # Enable x-axis (horizontal) grid lines
ax.yaxis.grid(False)  # Disable y-axis (vertical) grid lines

# Remove unnecessary spines
for spine in ['top', 'right', 'bottom']:
    ax.spines[spine].set_visible(False)

# Add data source at bottom right
plt.figtext(0.95, 0.01, 'Data: Kenya National Transport Authority', ha='right', fontsize=10)

# Adjust layout
plt.tight_layout()
plt.show()
```



10. How do emergency response time and proximity to healthcare facilities influence fatality outcomes in Nairobi crashes?

In Nairobi, both emergency response time and proximity to healthcare facilities significantly impact fatality outcomes in road crashes. Longer response times, coupled with limited access to timely care, contribute to increased mortality rates, especially in trauma cases.

In [ ]:

```

```