In [56]:

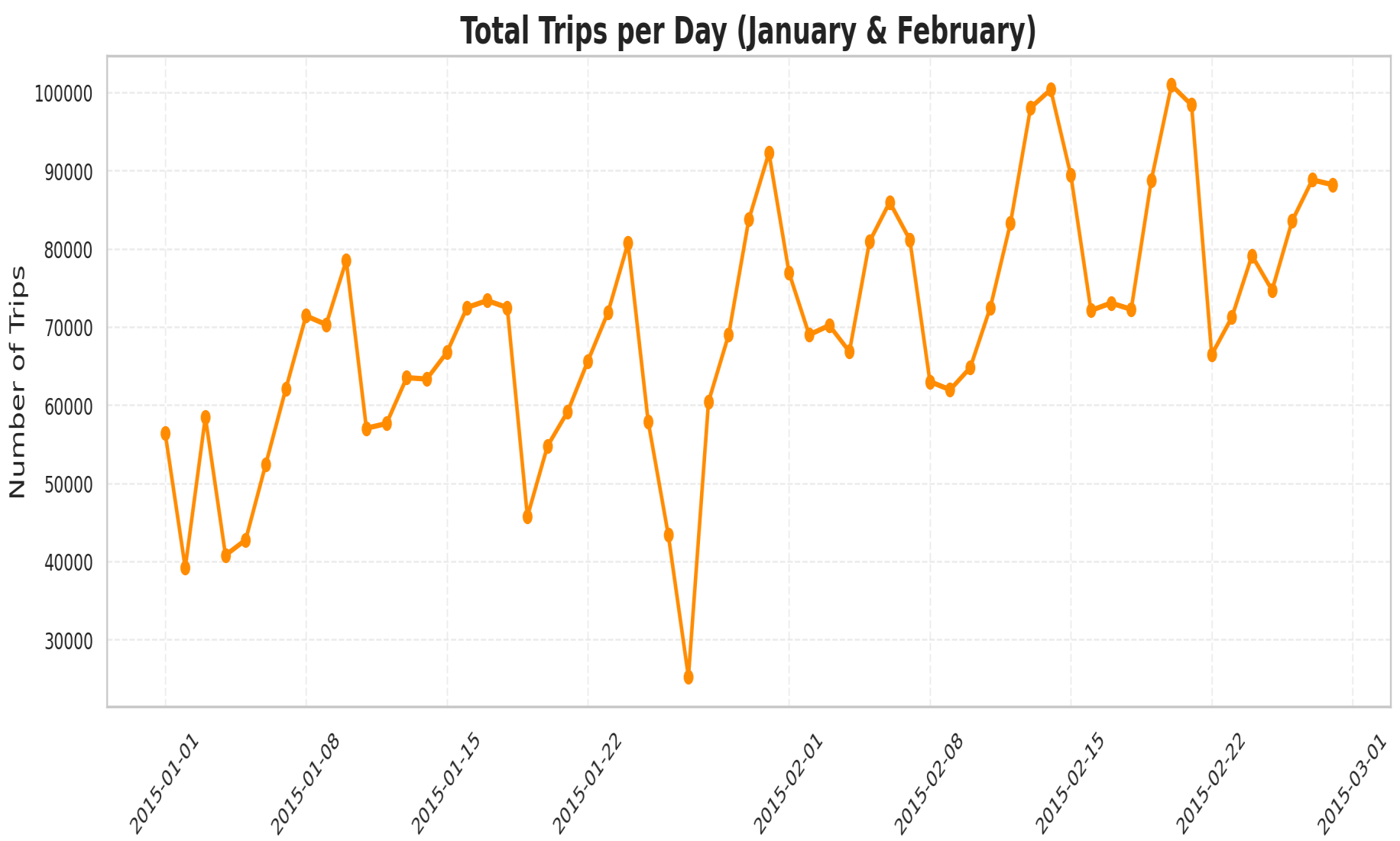
```
#     Name : Abhay Arun Chavan

#     ID:    UMID25062546328
#     course_name: Data Analytics Intership-(1 Month )
#     email : abhaychavan7700@gmail.com




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


plt.rcParams['font.family'] = 'sans-serif'


%config InlineBackend.figure_format = 'retina'

# Load the dataset
df = pd.read_csv('/content/Uber-Jan-Feb-FOIL.csv')


df['date'] = pd.to_datetime(df['date'])

# Extract day, month, weekday for analysis
df['day'] = df['date'].dt.day
df['month'] = df['date'].dt.month_name()
df['weekday'] = df['date'].dt.day_name()

# Display first few rows
df.head()
```

Out [56]:

	dispatching_base_number	date	active_vehicles	trips	day	month	weekday
0	B02512	2015-01-01	190	1132	1	January	Thursday
1	B02765	2015-01-01	225	1765	1	January	Thursday
2	B02764	2015-01-01	3427	29421	1	January	Thursday
3	B02682	2015-01-01	945	7679	1	January	Thursday
4	B02617	2015-01-01	1228	9537	1	January	Thursday

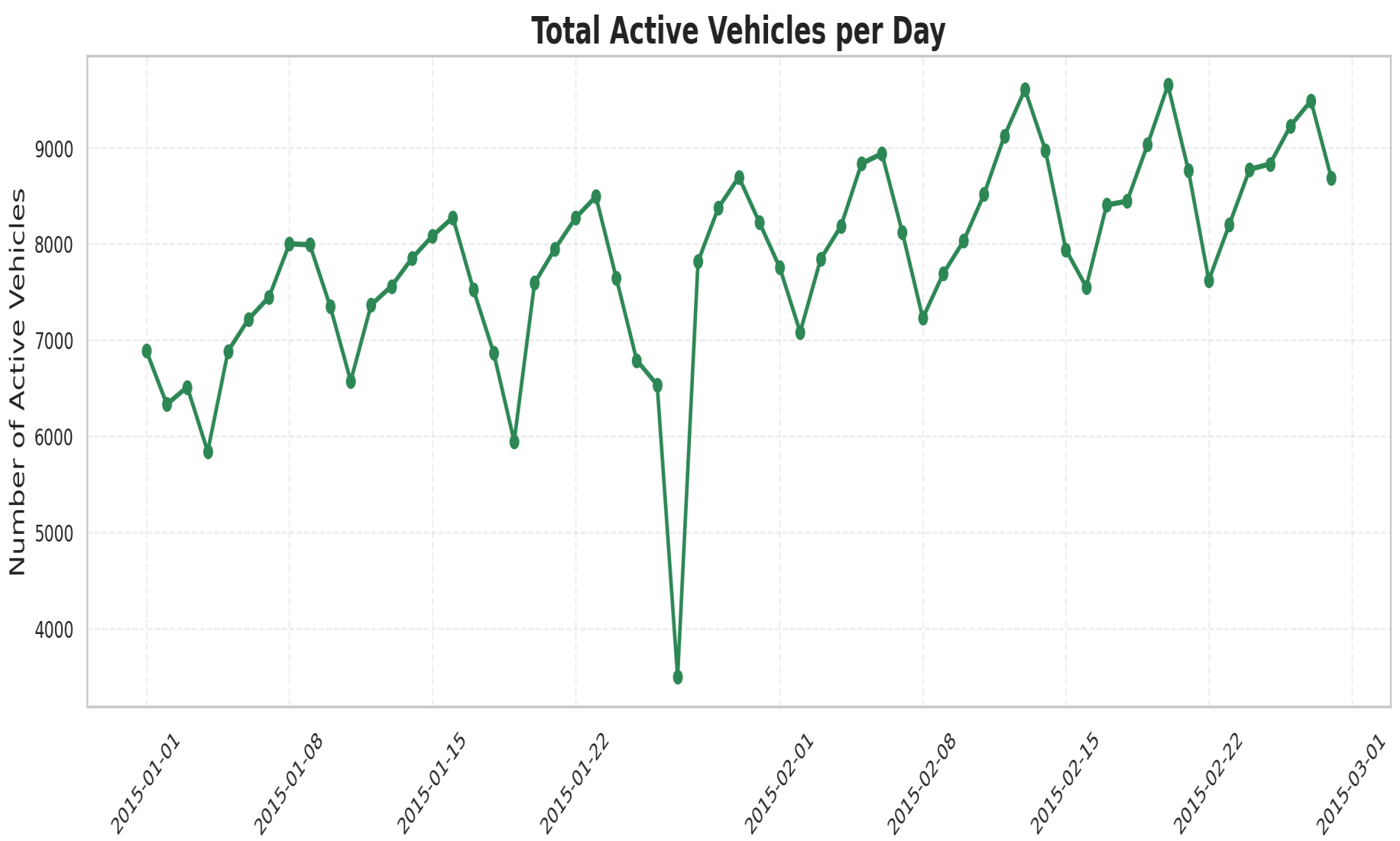
In [57]:

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

sns.set_theme(style="whitegrid")

plt.figure(figsize=(14,6))
daily_trips = df.groupby('date')['trips'].sum()

plt.plot(daily_trips.index, daily_trips.values, color='darkorange', linewidth=2.5, marker='o')

plt.title('Total Trips per Day (January & February)', fontsize=18, fontweight='bold')
plt.xlabel('Date', fontsize=14)
plt.ylabel('Number of Trips', fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```



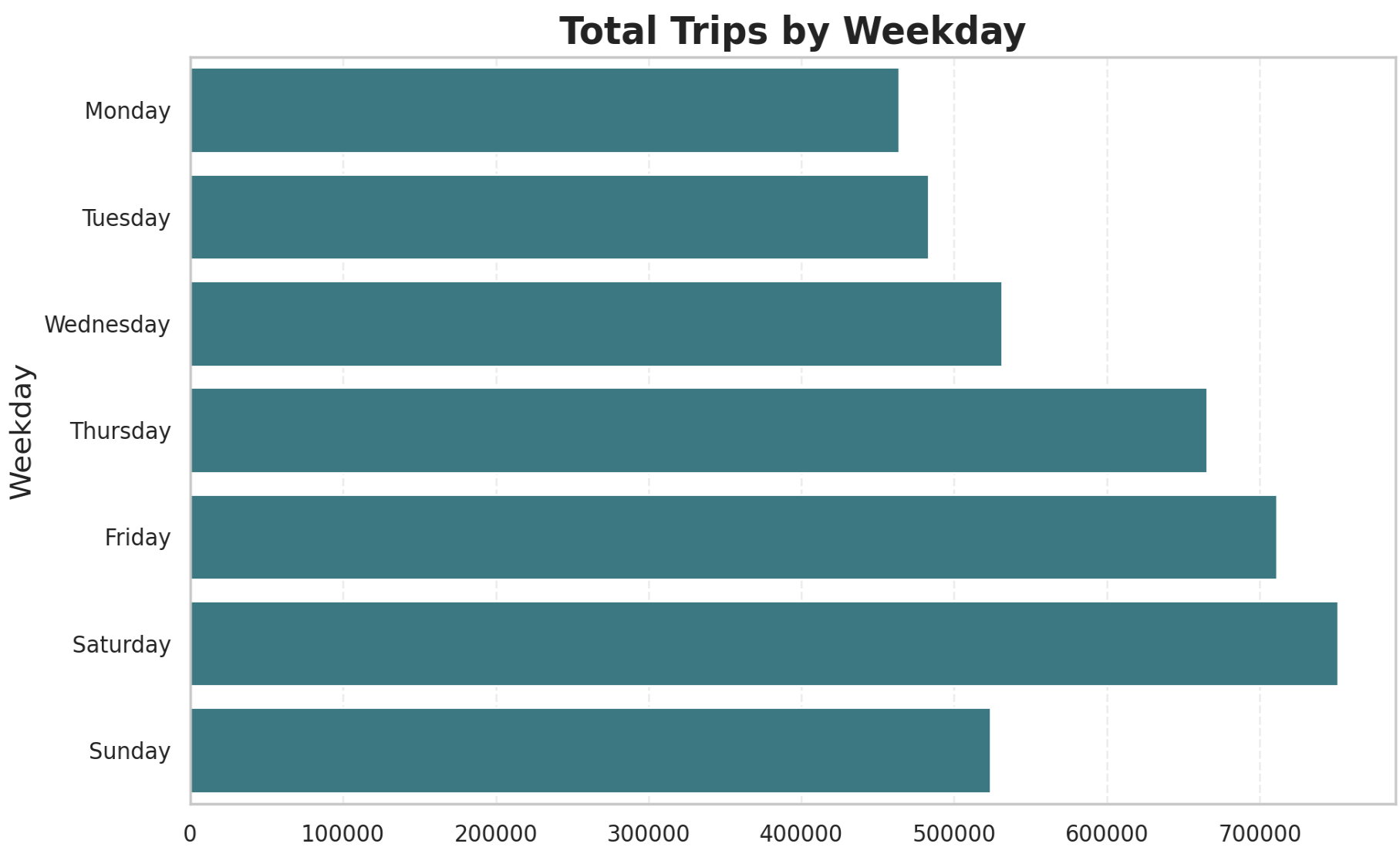
In [58]:

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


sns.set_theme(style="whitegrid")
plt.rcParams['font.family'] = 'sans-serif'


daily_vehicles = df.groupby('date')['active_vehicles'].sum()
plt.figure(figsize=(14,6))
plt.plot(daily_vehicles.index, daily_vehicles.values, color='seagreen', linewidth=2.5, marker='o')


plt.title('Total Active Vehicles per Day', fontsize=18, fontweight='bold')
plt.xlabel('Date', fontsize=14)
plt.ylabel('Number of Active Vehicles', fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```



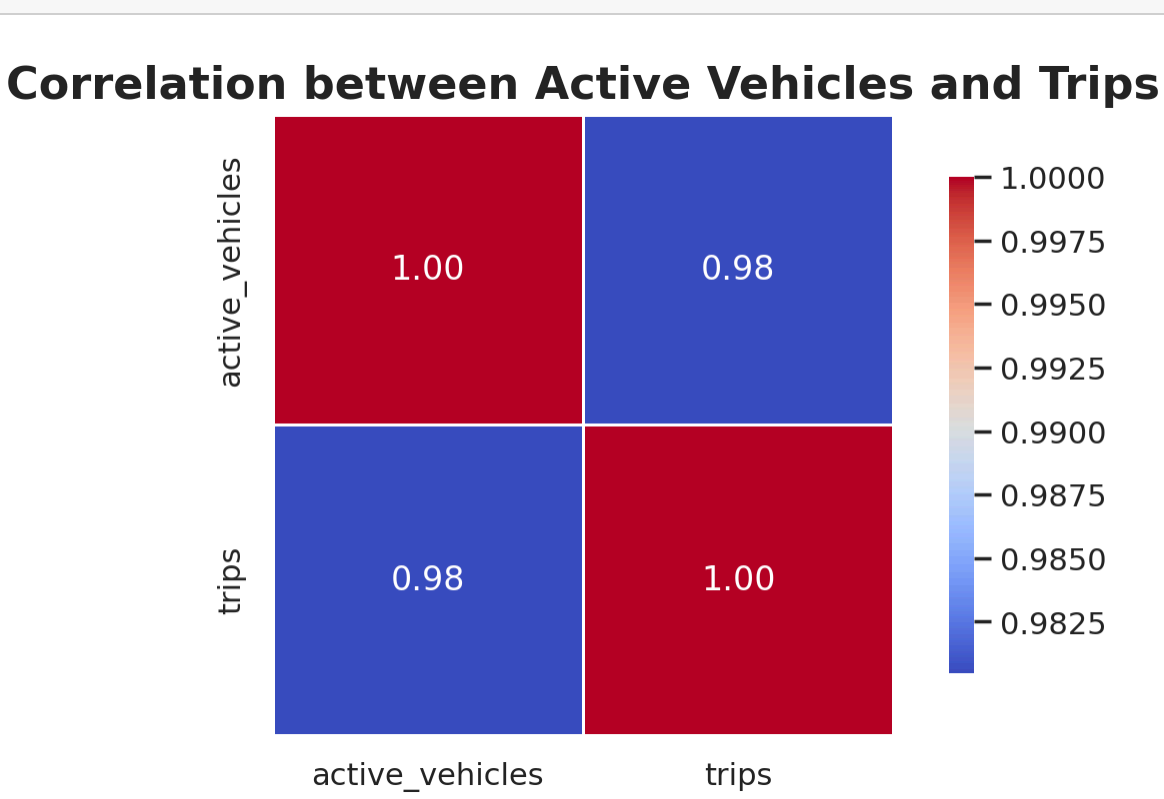
In [59]:

```
# Add a fake column 'Trip Count' to allow hue usage for color palette
weekday_trips_df = weekday_trips.reset_index()
weekday_trips_df.columns = ['Weekday', 'Total Trips']
weekday_trips_df['Trip Count'] = 'All Trips'  # dummy hue

# Plot horizontal bar chart with hue assigned
sns.set_style("whitegrid")
plt.figure(figsize=(10, 6))
sns.barplot(
    data=weekday_trips_df,
        x='Total Trips',
            y='Weekday',
                hue='Trip Count',
                    dodge=False,
                        palette='crest'
                        )

plt.title('Total Trips by Weekday', fontsize=18, fontweight='bold')
plt.xlabel('Total Trips', fontsize=14)
plt.ylabel('Weekday', fontsize=14)
plt.grid(True, axis='x', linestyle='--', alpha=0.3)
plt.tight_layout()
plt.legend().remove()  # hide legend since 'hue' is artificial
plt.show()
```



In [60]:

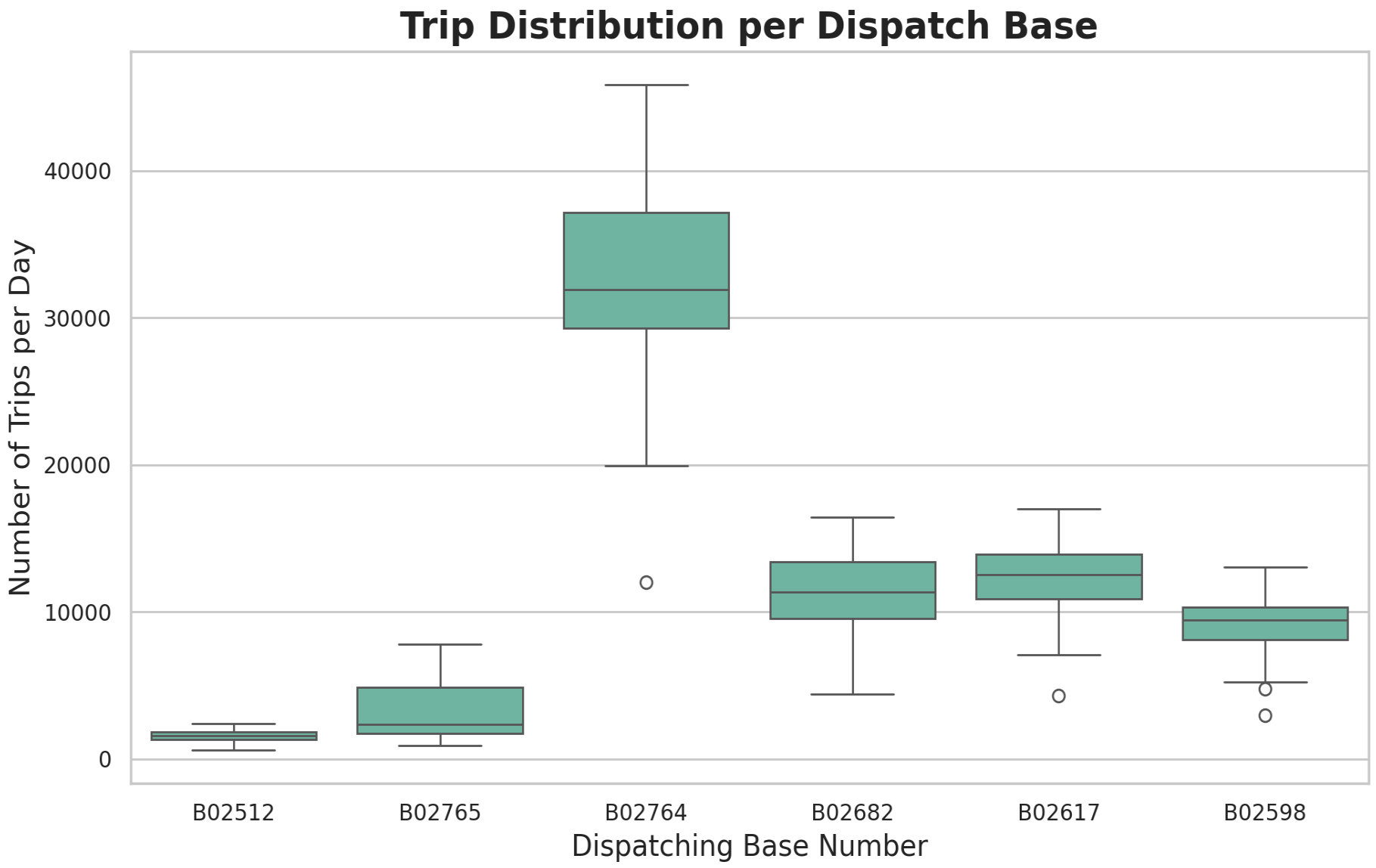
```
import matplotlib.pyplot as plt
import seaborn as sns


sns.set_theme(style="whitegrid")
plt.rcParams['font.family'] = 'sans-serif'


corr = df[['active_vehicles', 'trips']].corr()

# Plot heatmap
plt.figure(figsize=(6,4))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', square=True, linewidths=0.5,
            cbar_kws={"shrink": .8})


plt.title('Correlation between Active Vehicles and Trips', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
```



In [61]:

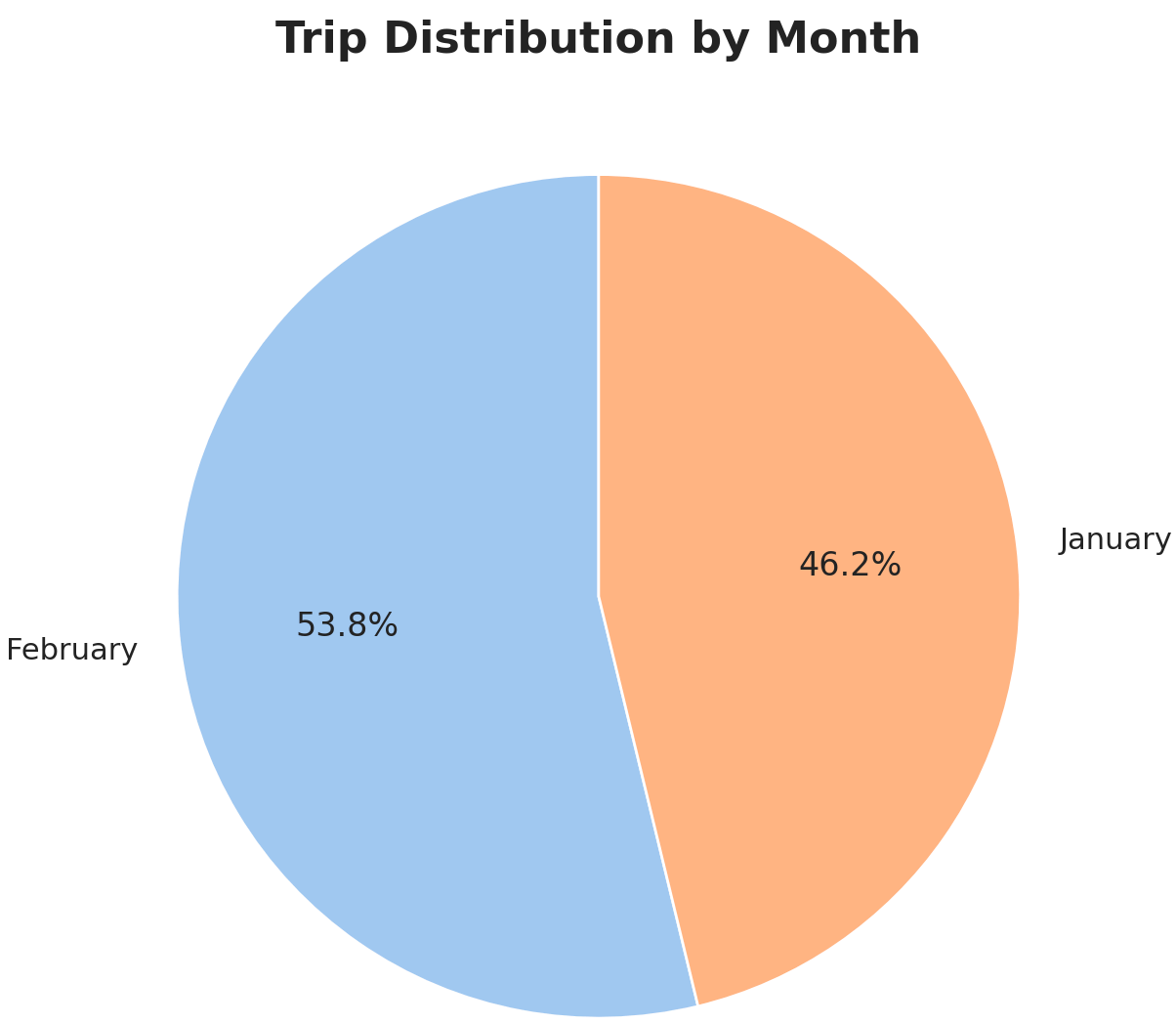
```
# Create dummy hue for color palette
df['hue_dummy'] = 'All Dispatch Bases'

# Set theme
sns.set_theme(style="whitegrid")
plt.rcParams['font.family'] = 'sans-serif'
```

```
# Plot boxplot with hue
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='dispatching_base_number', y='trips', hue='hue_dummy', dodge=False, palette='Set2')

# Labels and title
plt.title('Trip Distribution per Dispatch Base', fontsize=18, fontweight='bold')
plt.xlabel('Dispatching Base Number', fontsize=14)
plt.ylabel('Number of Trips per Day', fontsize=14)
plt.tight_layout()
plt.legend().remove()  # Remove legend for dummy hue
plt.show()
```



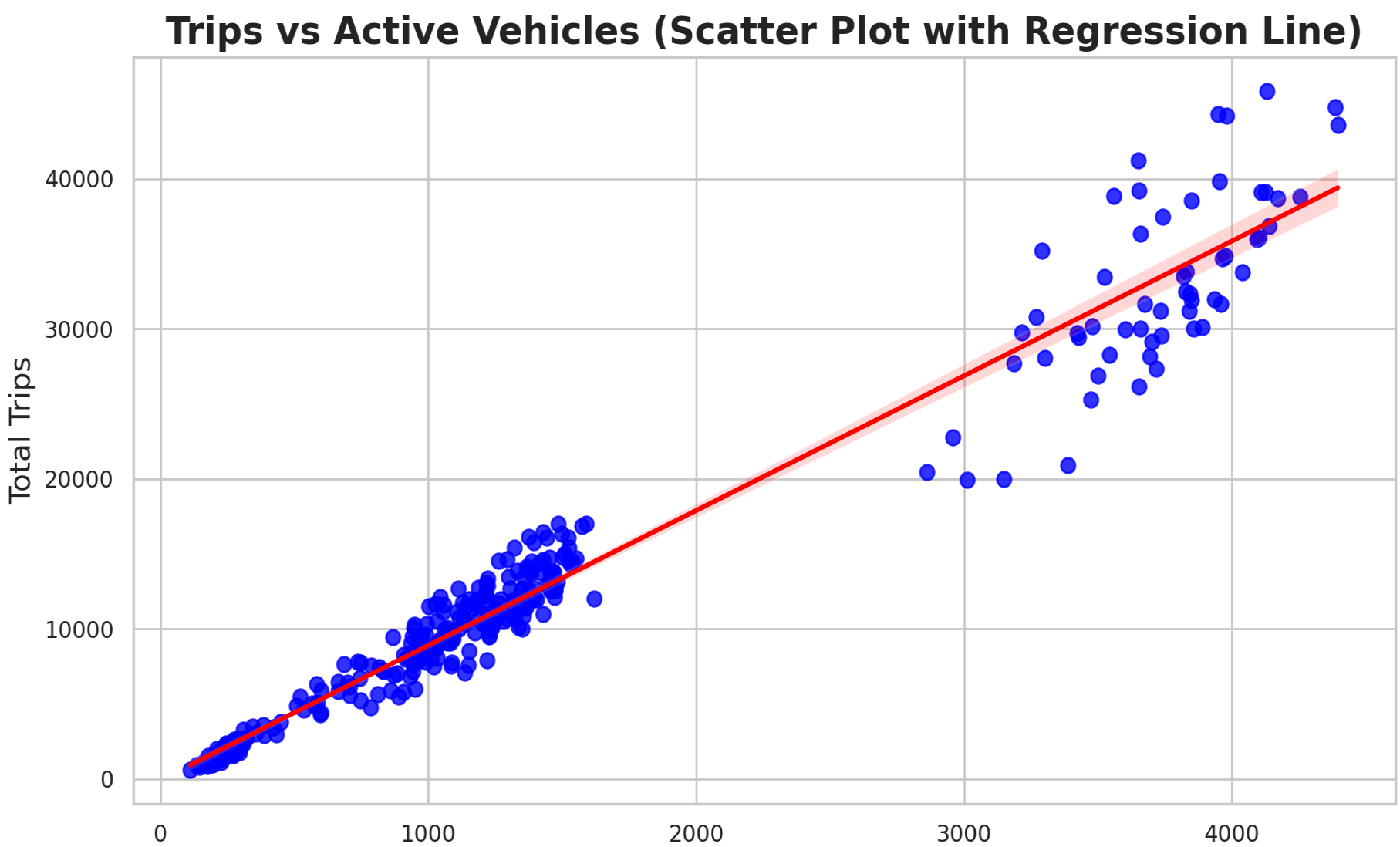
In [62]:

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


sns.set_theme(style="whitegrid")
plt.rcParams['font.family'] = 'sans-serif'

month_trips = df.groupby('month')['trips'].sum()


colors = sns.color_palette('pastel')

plt.figure(figsize=(8,6))
plt.pie(month_trips, labels=month_trips.index, autopct='%1.1f%%', colors=colors, startangle=90)


plt.title('Trip Distribution by Month', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
```



In [63]:

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


sns.set_theme(style="whitegrid")
plt.rcParams['font.family'] = 'sans-serif'

# Scatter plot with regression line
plt.figure(figsize=(10,6))
sns.regplot(x='active_vehicles', y='trips', data=df, scatter_kws={'color':'blue', 's':50},
            line_kws={'color':'red'})


plt.title('Trips vs Active Vehicles (Scatter Plot with Regression Line)', fontsize=18, fontweight='bold')
plt.xlabel('Active Vehicles', fontsize=14)
plt.ylabel('Total Trips', fontsize=14)
plt.tight_layout()
plt.show()
```



In [64]:
```
# FINAL SUMMARY: Data-Driven Insights

# 1. Total Trips
total_trips = df['trips'].sum()

# 2. Most Active Base
base_trips = df.groupby('dispatching_base_number')['trips'].sum()
top_base = base_trips.idxmax()
top_base_trips = base_trips.max()

# 3. Busiest Day
day_trips = df.groupby('date')['trips'].sum()
busiest_day = day_trips.idxmax()
busiest_day_trips = day_trips.max()

# 4. Most Active Weekday
weekday_trips = df.groupby('weekday')['trips'].sum()
most_active_weekday = weekday_trips.idxmax()
weekday_trip_count = weekday_trips.max()

# 5. Average Vehicles Per Day
avg_vehicles = df.groupby('date')['active_vehicles'].sum().mean()

# Display Summary
print("📊 UBER DATA ANALYTICS SUMMARY (JAN - FEB 2015)\n")
print(f"• Total Trips Recorded: {total_trips:,}")
print(f"• Most Active Dispatch Base: {top_base} with {top_base_trips:,} trips")
print(f"• Busiest Day: {busiest_day.date()} with {busiest_day_trips:,} trips")
print(f"• Peak Weekday: {most_active_weekday} ({weekday_trip_count:,} trips)")
print(f"• Average Number of Active Vehicles Per Day: {avg_vehicles:.0f}")
```

```
📊 UBER DATA ANALYTICS SUMMARY (JAN - FEB 2015)

• Total Trips Recorded: 4,130,230
• Most Active Dispatch Base: B02764 with 1,914,449 trips
• Busiest Day: 2015-02-20 with 100,915 trips
• Peak Weekday: Saturday (751,325 trips)
• Average Number of Active Vehicles Per Day: 7845
```