🚗 ⚡ 📊 EDA | Electric Vehicle Analytics | By Vivek Chauhan 📊 ⚡ 🚗

In [ ]:
```
# Load all the necessary libraries for the exploratory data analysis

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

In [12]:
```
# Load/Read actual data

data = pd.read_csv("C:/Users/VIVEK CHAUHAN/Desktop/electric_vehicle_analytics.csv")
data
```

Out[12]:

	Vehicle_ID	Make	Model	Year	Region	Vehicle_Type	Battery_Capacity_kWh	Battery_Health_%	Range_km	Charging_Power_I
0	1	Nissan	Leaf	2021	Asia	SUV	101.7	75.5	565	15
1	2	Nissan	Leaf	2020	Australia	Sedan	30.1	99.8	157	15
2	3	Hyundai	Kona Electric	2021	North America	SUV	118.5	84.0	677	17
3	4	Audi	Q4 e-tron	2022	Europe	Hatchback	33.1	97.3	149	16
4	5	Tesla	Model 3	2022	Australia	Truck	81.3	85.6	481	21
...	...	...	...	...	...	...	...	...	...	
2995	2996	Mercedes	EQS	2021	North America	SUV	57.2	84.0	239	10
2996	2997	Ford	Mustang Mach-E	2022	Europe	Hatchback	98.4	83.1	498	16
2997	2998	Kia	Niro EV	2024	Europe	Truck	35.1	82.1	189	1
2998	2999	Mercedes	EQC	2015	North America	Truck	69.4	98.4	336	9
2999	3000	Audi	Q4 e-tron	2023	North America	Hatchback	70.2	82.6	387	23

3000 rows × 25 columns

Data Analysis

In [14]:

```
# Print first 5 data from the dataset

data.head()
```

Out[14]:

	Vehicle_ID	Make	Model	Year	Region	Vehicle_Type	Battery_Capacity_kWh	Battery_Health_%	Range_km	Charging_Power_kW	...
0	1	Nissan	Leaf	2021	Asia	SUV	101.7	75.5	565	153.6	...
1	2	Nissan	Leaf	2020	Australia	Sedan	30.1	99.8	157	157.2	...
2	3	Hyundai	Kona Electric	2021	North America	SUV	118.5	84.0	677	173.6	...
3	4	Audi	Q4 e-tron	2022	Europe	Hatchback	33.1	97.3	149	169.3	...
4	5	Tesla	Model 3	2022	Australia	Truck	81.3	85.6	481	212.8	...

5 rows × 25 columns

In [15]:

```
# print last 5 data from the dataset

data.tail()
```

Out[15]:

	Vehicle_ID	Make	Model	Year	Region	Vehicle_Type	Battery_Capacity_kWh	Battery_Health_%	Range_km	Charging_Power_k
2995	2996	Mercedes	EQS	2021	North America	SUV	57.2	84.0	239	10
2996	2997	Ford	Mustang Mach-E	2022	Europe	Hatchback	98.4	83.1	498	16
2997	2998	Kia	Niro EV	2024	Europe	Truck	35.1	82.1	189	1
2998	2999	Mercedes	EQC	2015	North America	Truck	69.4	98.4	336	9
2999	3000	Audi	Q4 e-tron	2023	North America	Hatchback	70.2	82.6	387	23

5 rows × 25 columns

In [16]:

```
# Print the all the columnn names from the dataset

data.columns
```

Out[16]:

```
Index(['Vehicle_ID', 'Make', 'Model', 'Year', 'Region', 'Vehicle_Type',
       'Battery_Capacity_kWh', 'Battery_Health_%', 'Range_km',
       'Charging_Power_kW', 'Charging_Time_hr', 'Charge_Cycles',
       'Energy_Consumption_kWh_per_100km', 'Mileage_km', 'Avg_Speed_kmh',
       'Max_Speed_kmh', 'Acceleration_0_100_kmh_sec', 'Temperature_C',
       'Usage_Type', 'CO2_Saved_tons', 'Maintenance_Cost_USD',
       'Insurance_Cost_USD', 'Electricity_Cost_USD_per_kWh',
       'Monthly_Charging_Cost_USD', 'Resale_Value_USD'],
      dtype='object')
```

In [17]:

```
# Print the overall statistics from the dataset

data.describe().T
```

Out[17]:

	count	mean	std	min	25%	50%	75%	max
Vehicle_ID	3000.0	1500.500000	866.169729	1.00	750.7500	1500.500	2250.2500	3000.00
Year	3000.0	2019.499667	2.848047	2015.00	2017.0000	2020.000	2022.0000	2024.00
Battery_Capacity_kWh	3000.0	74.810100	25.734079	30.00	53.0000	74.850	96.9000	120.00
Battery_Health_%	3000.0	85.030000	8.589526	70.00	77.7750	85.250	92.3000	100.00
Range_km	3000.0	374.414667	137.184112	121.00	260.0000	371.000	476.2500	713.00
Charging_Power_kW	3000.0	129.301000	68.742745	11.10	70.9000	126.700	187.9750	250.00
Charging_Time_hr	3000.0	1.203570	1.421866	0.14	0.4600	0.720	1.2925	12.14
Charge_Cycles	3000.0	1107.009667	510.834590	200.00	674.7500	1116.000	1535.2500	1997.00
Energy_Consumption_kWh_per_100km	3000.0	18.589740	3.767421	12.00	15.3200	18.700	21.8300	24.99
Mileage_km	3000.0	125209.685667	70465.774772	5046.00	65140.5000	125965.000	184764.7500	249987.00
Avg_Speed_kmh	3000.0	65.674067	20.305364	30.00	48.0000	65.600	83.5250	100.00
Max_Speed_kmh	3000.0	190.678333	35.184232	130.00	159.0000	191.500	222.0000	249.00
Acceleration_0_100_kmh_sec	3000.0	6.707073	1.880355	3.50	5.0800	6.720	8.3100	10.00
Temperature_C	3000.0	14.794133	14.407087	-10.00	2.4000	14.550	27.5000	40.00
CO2_Saved_tons	3000.0	15.025163	8.455850	0.61	7.8175	15.115	22.1700	30.00
Maintenance_Cost_USD	3000.0	1104.199000	521.530356	200.00	652.0000	1109.000	1569.0000	1999.00
Insurance_Cost_USD	3000.0	1495.469333	585.919978	500.00	984.7500	1496.000	2019.2500	2498.00
Electricity_Cost_USD_per_kWh	3000.0	0.216467	0.078383	0.08	0.1500	0.220	0.2800	0.35
Monthly_Charging_Cost_USD	3000.0	418.814683	312.389226	7.99	175.4850	347.285	595.1525	1643.70
Resale_Value_USD	3000.0	22257.038000	5594.979382	8506.00	17813.0000	22154.000	26732.7500	35521.00

In [18]:

```
# Print all the information about the datasets
```

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 25 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Vehicle_ID                     3000 non-null   int64  
 1   Make                           3000 non-null   object 
 2   Model                          3000 non-null   object 
 3   Year                           3000 non-null   int64  
 4   Region                         3000 non-null   object 
 5   Vehicle_Type                   3000 non-null   object 
 6   Battery_Capacity_kWh           3000 non-null   float64
 7   Battery_Health_%               3000 non-null   float64
 8   Range_km                       3000 non-null   int64  
 9   Charging_Power_kW              3000 non-null   float64
 10  Charging_Time_hr               3000 non-null   float64
 11  Charge_Cycles                  3000 non-null   int64  
 12  Energy_Consumption_kWh_per_100km  3000 non-null   float64
 13  Mileage_km                     3000 non-null   int64  
 14  Avg_Speed_kmh                  3000 non-null   float64
 15  Max_Speed_kmh                  3000 non-null   int64  
 16  Acceleration_0_100_kmh_sec     3000 non-null   float64
 17  Temperature_C                  3000 non-null   float64
 18  Usage_Type                     3000 non-null   object 
 19  CO2_Saved_tons                 3000 non-null   float64
 20  Maintenance_Cost_USD           3000 non-null   int64  
 21  Insurance_Cost_USD             3000 non-null   int64  
 22  Electricity_Cost_USD_per_kWh  3000 non-null   float64
 23  Monthly_Charging_Cost_USD      3000 non-null   float64
 24  Resale_Value_USD               3000 non-null   int64  
dtypes: float64(11), int64(9), object(5)
memory usage: 586.1+ KB
```

In [19]:

```python
# Print the shape of the data

data.shape
```

Out[19]: (3000, 25)

In [20]:
```
# Print all the dtypes of the dataset

data.dtypes
```

Out[20]:
```
Vehicle_ID                              int64
Make                                   object
Model                                  object
Year                                    int64
Region                                 object
Vehicle_Type                           object
Battery_Capacity_kWh                  float64
Battery_Health_%                      float64
Range_km                                int64
Charging_Power_kW                     float64
Charging_Time_hr                      float64
Charge_Cycles                           int64
Energy_Consumption_kWh_per_100km      float64
Mileage_km                              int64
Avg_Speed_kmh                         float64
Max_Speed_kmh                           int64
Acceleration_0_100_kmh_sec            float64
Temperature_C                         float64
Usage_Type                             object
CO2_Saved_tons                        float64
Maintenance_Cost_USD                    int64
Insurance_Cost_USD                      int64
Electricity_Cost_USD_per_kWh          float64
Monthly_Charging_Cost_USD             float64
Resale_Value_USD                        int64
dtype: object
```

In [21]:
```
# check is there any null values is present in our dataset or not

data.isnull().sum()
```

Out[21]:
```
Vehicle_ID                      0
Make                            0
Model                           0
Year                            0
Region                          0
Vehicle_Type                    0
Battery_Capacity_kWh            0
Battery_Health_%                0
Range_km                        0
Charging_Power_kW               0
Charging_Time_hr                0
Charge_Cycles                   0
Energy_Consumption_kWh_per_100km    0
Mileage_km                      0
Avg_Speed_kmh                   0
Max_Speed_kmh                   0
Acceleration_0_100_kmh_sec      0
Temperature_C                   0
Usage_Type                      0
CO2_Saved_tons                  0
Maintenance_Cost_USD            0
Insurance_Cost_USD              0
Electricity_Cost_USD_per_kWh    0
Monthly_Charging_Cost_USD       0
Resale_Value_USD                0
dtype: int64
```

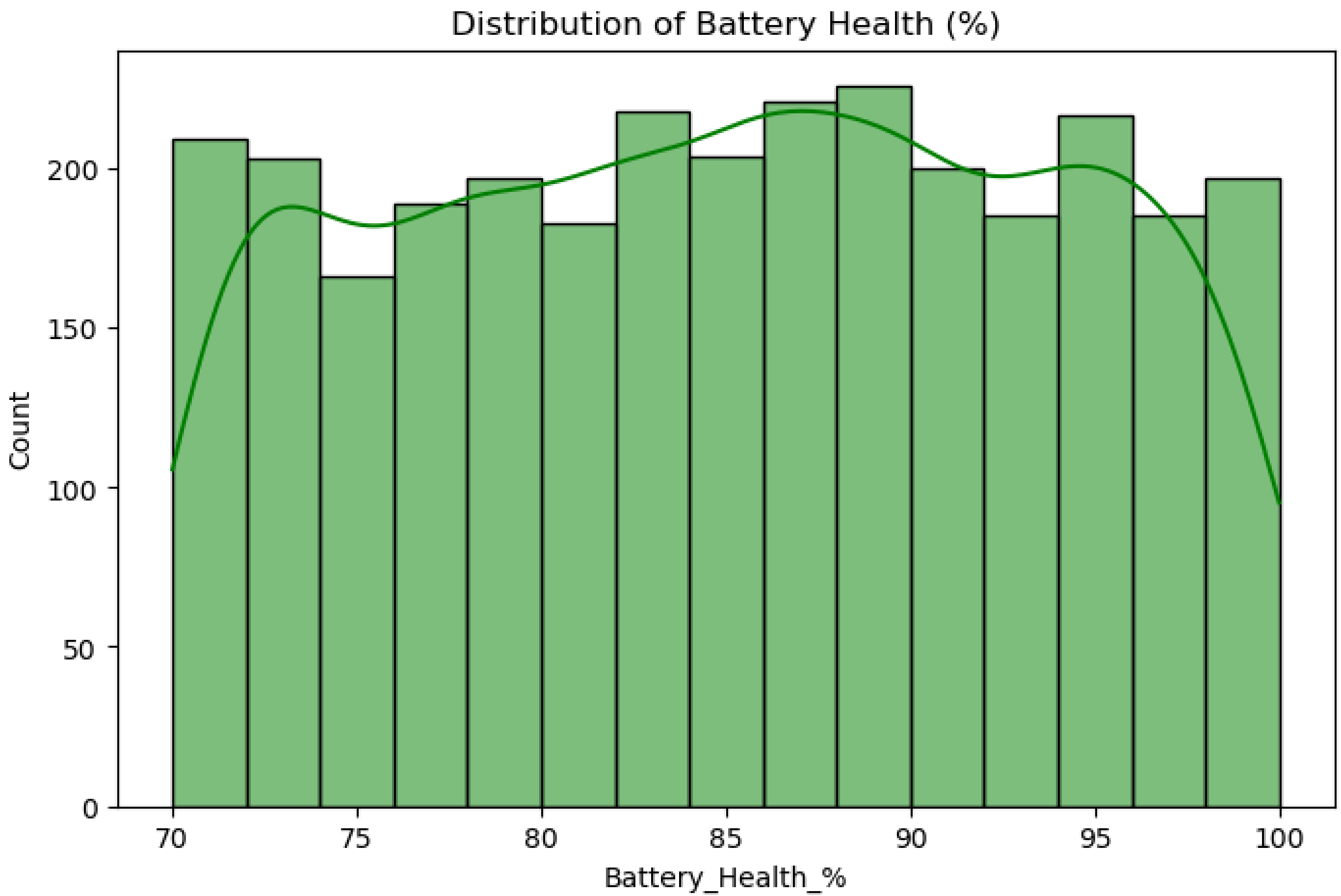
In [22]:
```python
# check is there any duplicate values is present in our dataset or not

data.duplicated().sum()
```

Out[22]:
```
0
```

Uni-Variate Analysis

In [24]:
```python
# Distribution of Battery Health
plt.figure(figsize=(8,5))
sns.histplot(data['Battery_Health_%'], kde=True, color="green")
```

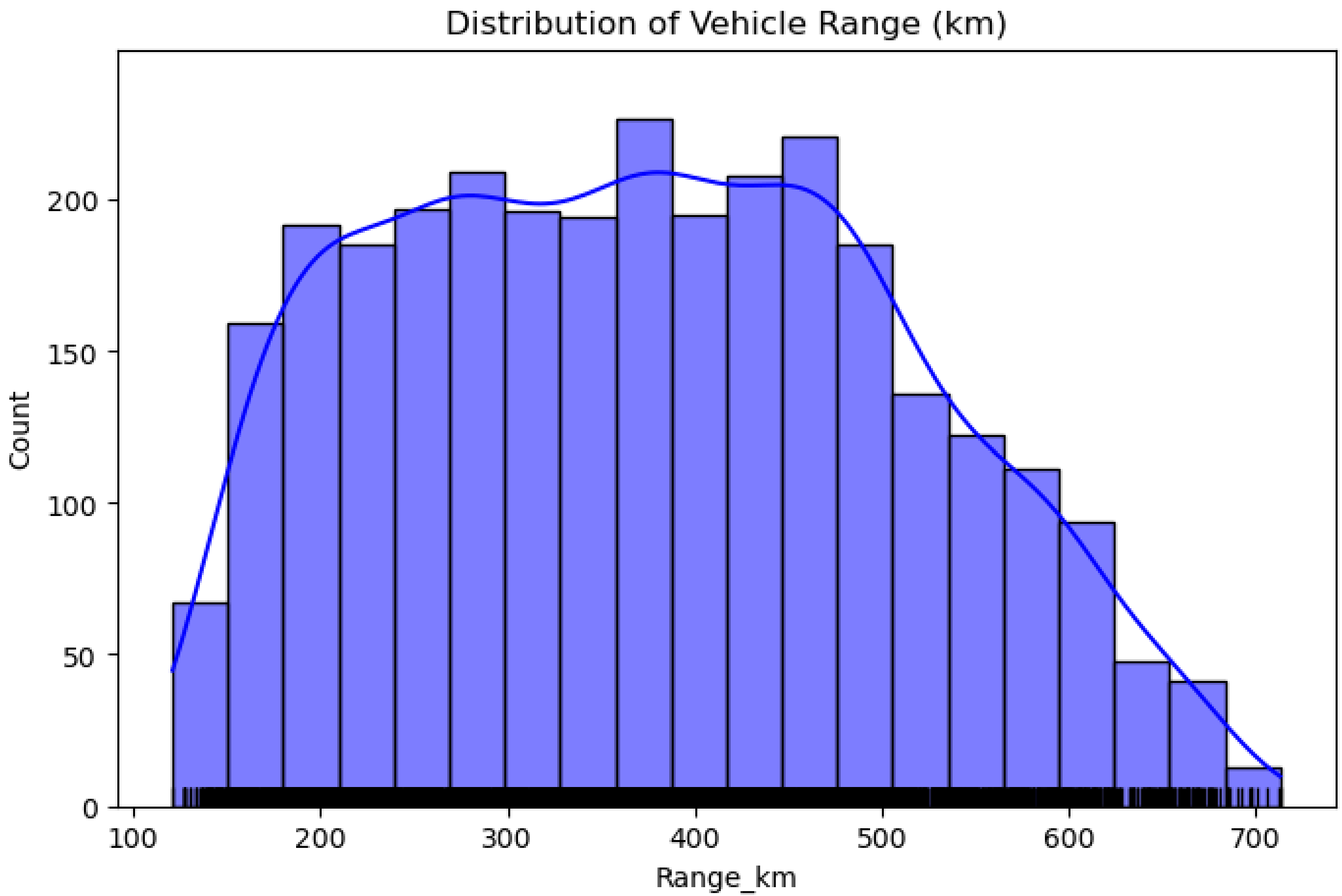
```
plt.title("Distribution of Battery Health (%)")
plt.show()
```



The distribution of Battery Health (%) deviates from a normal bell-shaped curve, indicating possible skewness or irregular wear patterns across vehicles.

In [26]:
```
# Range distribution with rug plot
plt.figure(figsize=(8,5))
```

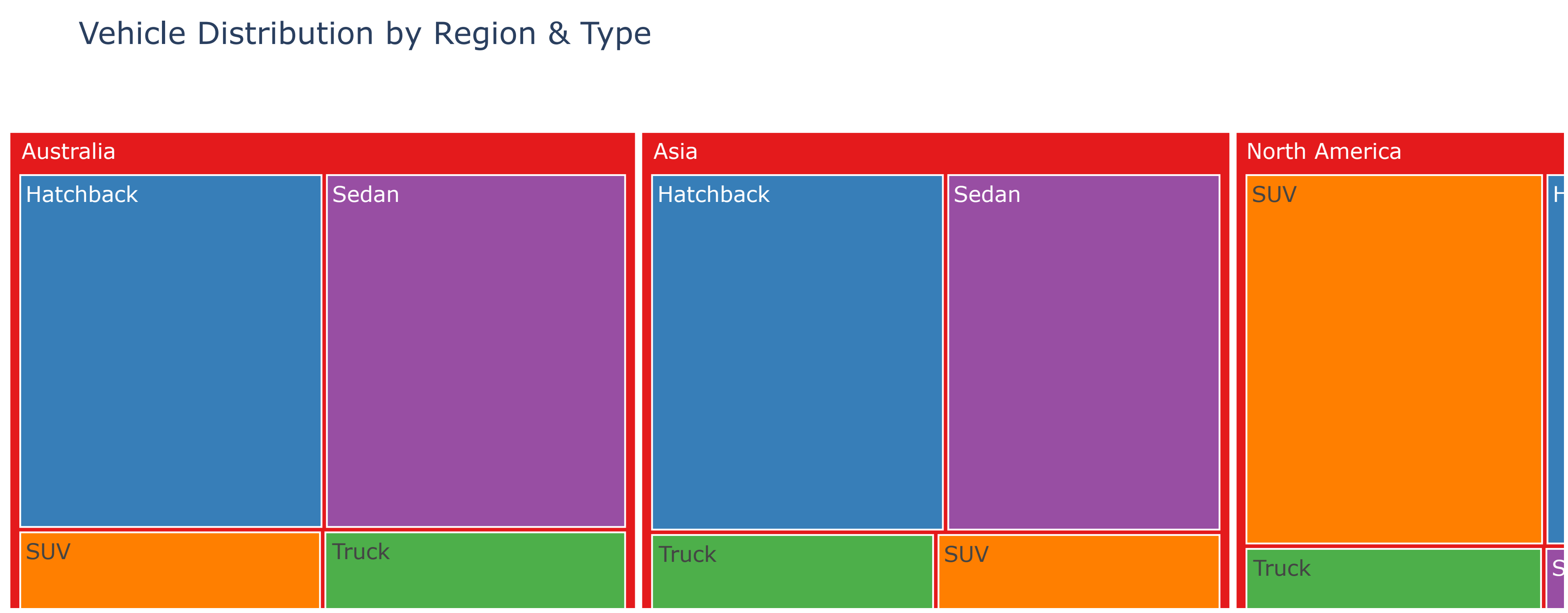
```python
sns.histplot(data['Range_km'], kde=True, color="blue")
sns.rugplot(data['Range_km'], color="black")
plt.title("Distribution of Vehicle Range (km)")
plt.show()
```



The distribution of Vehicle Range is left-skewed, with higher frequencies on the lower end and a gradual decline towards longer ranges, suggesting most vehicles have shorter range capabilities.

In [28]:

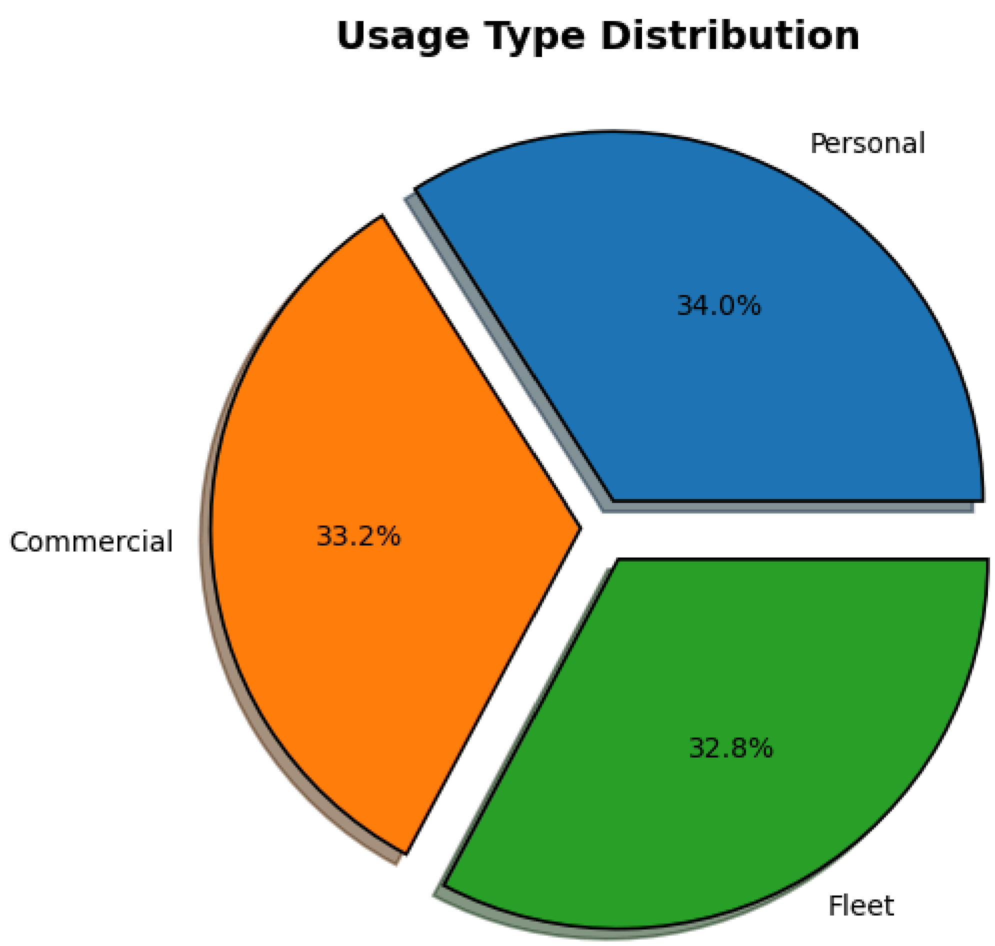
```python
import plotly.express as px

fig = px.treemap(
    data,
    path=['Region','Vehicle_Type'],
    title="Vehicle Distribution by Region & Type",
    color='Vehicle_Type',            # color by vehicle type
    color_discrete_sequence=px.colors.qualitative.Set1  # attractive color palette
)
fig.update_layout(margin=dict(t=50, l=25, r=25, b=25))
fig.show()
```



The treemap highlights regional vehicle trends: Hatchbacks are most prevalent in Asia and Australia, while SUVs dominate in North America, reflected by the larger tiles in these regions.

In [30]:

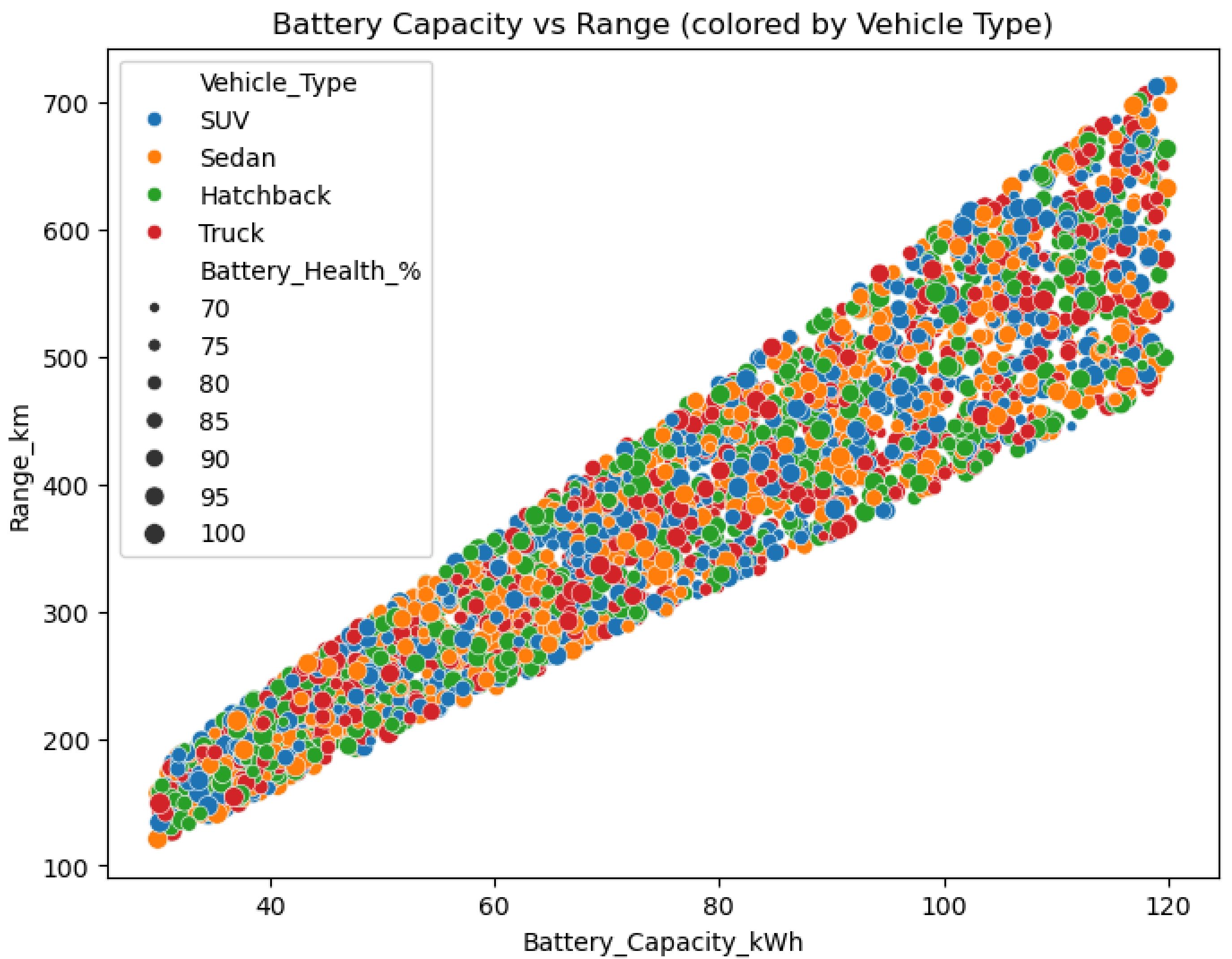
```
data['Usage_Type'].value_counts().plot.pie(
    autopct="%1.1f%%",
    figsize=(6,6),
    colors=sns.color_palette("tab10"),   # bold vibrant colors
    shadow=True,
    explode=[0.08,0.05,0.1,0.07][:data['Usage_Type'].nunique()],  # uneven explode for “noise” effect
    wedgeprops={'edgecolor':'black','linewidth':1.2}
)
plt.title("Usage Type Distribution", fontsize=14, fontweight='bold')
plt.ylabel("")
plt.show()
```



The pie chart indicates that Personal usage constitutes the largest segment of the dataset, followed by Commercial and Fleet categories, highlighting the predominance of individual vehicle ownership over business or fleet operations.

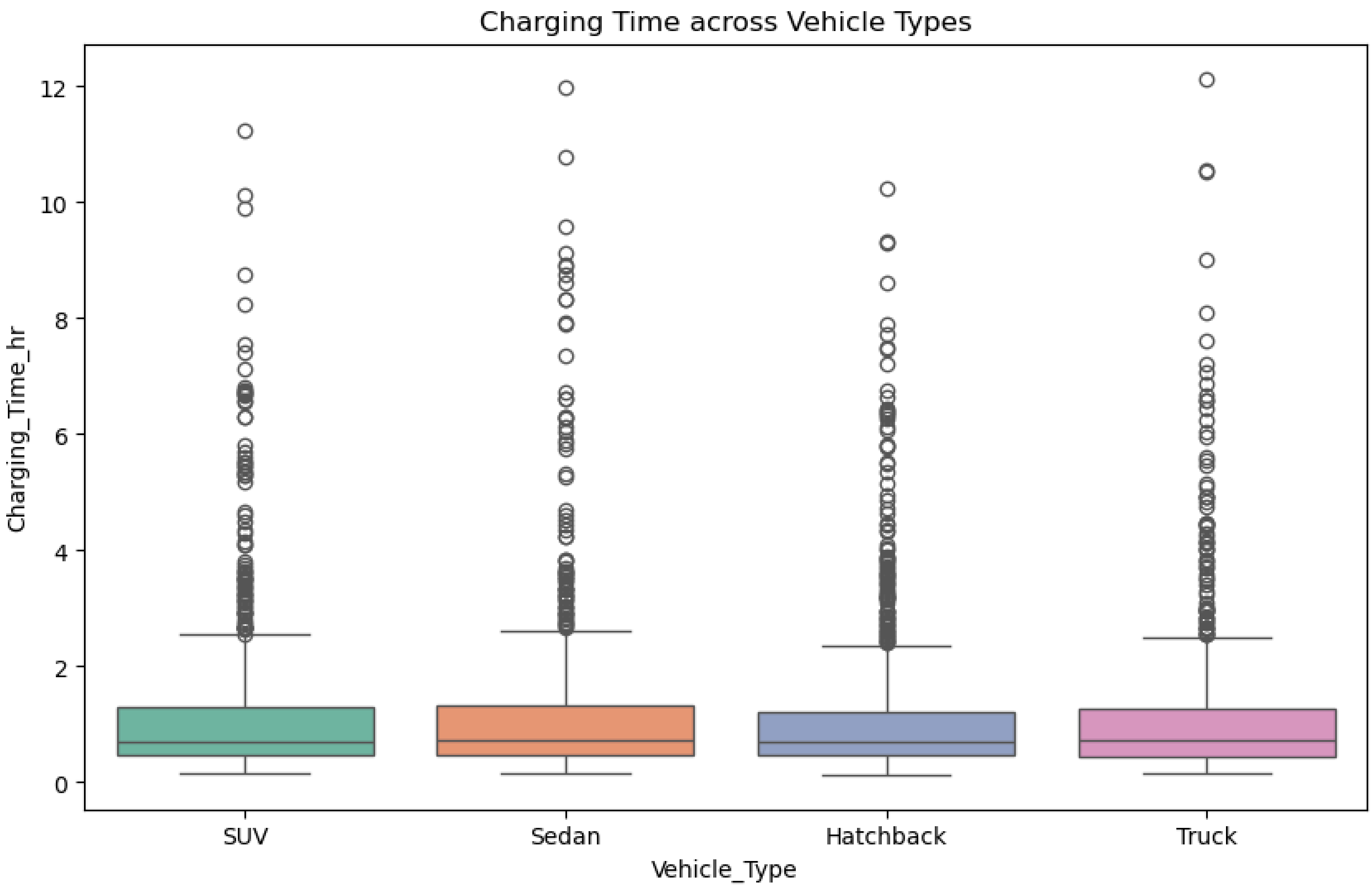
Bi-Variate analysis

In [33]:
```
# Battery Capacity vs Range
plt.figure(figsize=(8,6))
sns.scatterplot(x="Battery_Capacity_kWh", y="Range_km", hue="Vehicle_Type", size="Battery_Health_%", data=data)
plt.title("Battery Capacity vs Range (colored by Vehicle Type)")
plt.show()
```



The scatter plot demonstrates a positive relationship between Battery Capacity and Vehicle Range, indicating that vehicles with higher battery capacities generally achieve longer ranges.

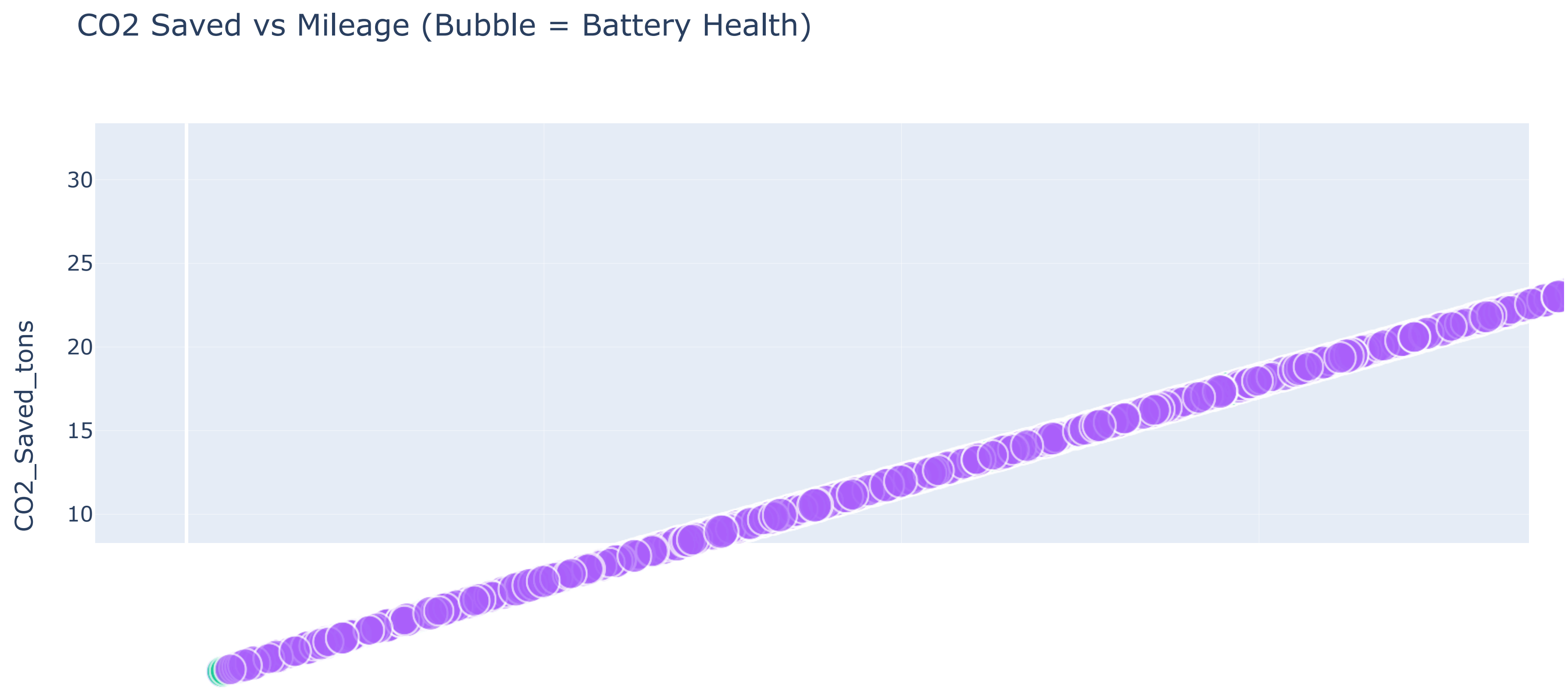
In [35]:
```
# Charging Time by Vehicle Type
plt.figure(figsize=(10,6))
sns.boxplot(x="Vehicle_Type", y="Charging_Time_hr", data=data, palette="Set2")
plt.title("Charging Time across Vehicle Types")
plt.show()
```



The boxplot shows that Charging Time is relatively consistent across different Vehicle Types, with a similar range of variation observed in each category.

In [37]:

```
# Bubble Chart: CO2 Saved vs Mileage with Battery Health as Size
fig = px.scatter(data, x="Mileage_km", y="CO2_Saved_tons",
                 size="Battery_Health_%", color="Region",
                 hover_data=["Make","Model"],
                 title="CO2 Saved vs Mileage (Bubble = Battery Health)")
fig.show()
```

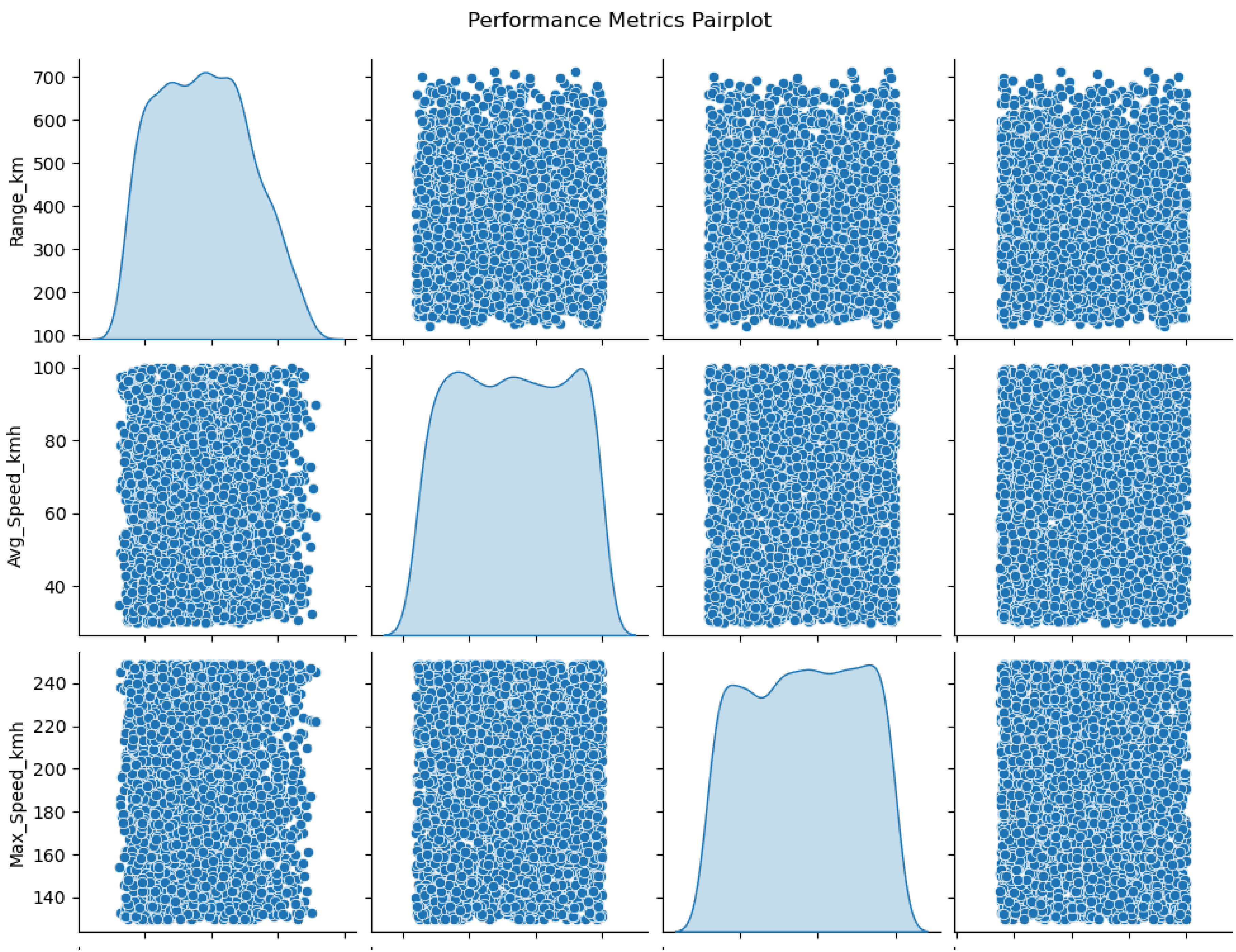


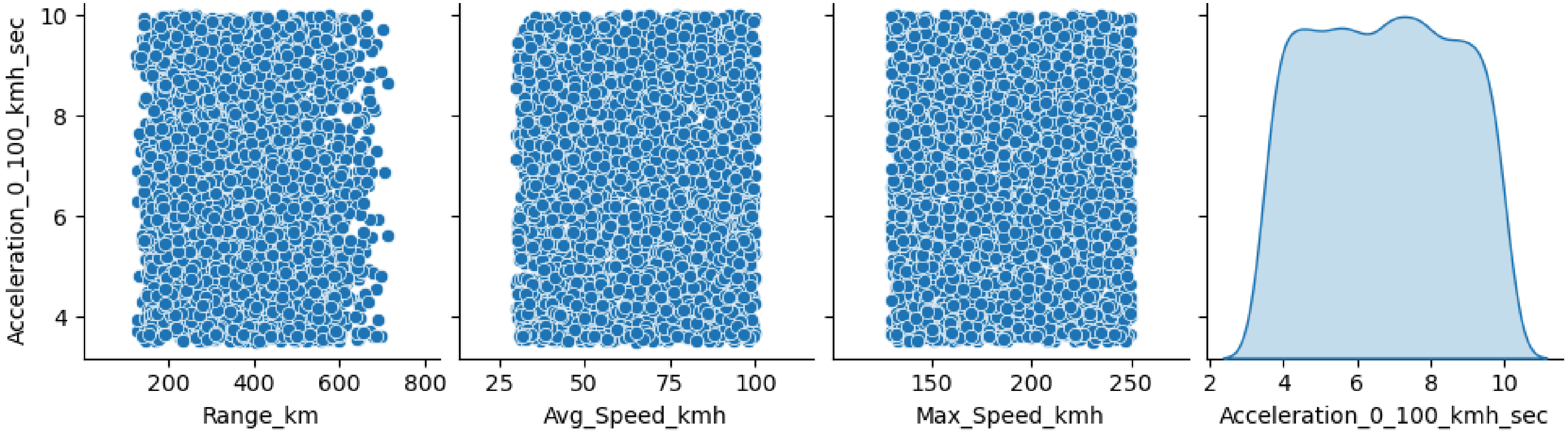
The bubble chart illustrates a positive relationship between Mileage and CO2 Saved, showing that vehicles with higher mileage tend to save more CO2. The lavender-colored points indicate the regional distribution, highlighting concentration in specific regions.

Multi-Variate Analysis

In [40]:

```
# Pairplot for performance metrics
sns.pairplot(data[["Range_km","Avg_Speed_kmh","Max_Speed_kmh","Acceleration_0_100_kmh_sec"]], diag_kind="kde")
plt.suptitle("Performance Metrics Pairplot", y=1.02)
plt.show()
```

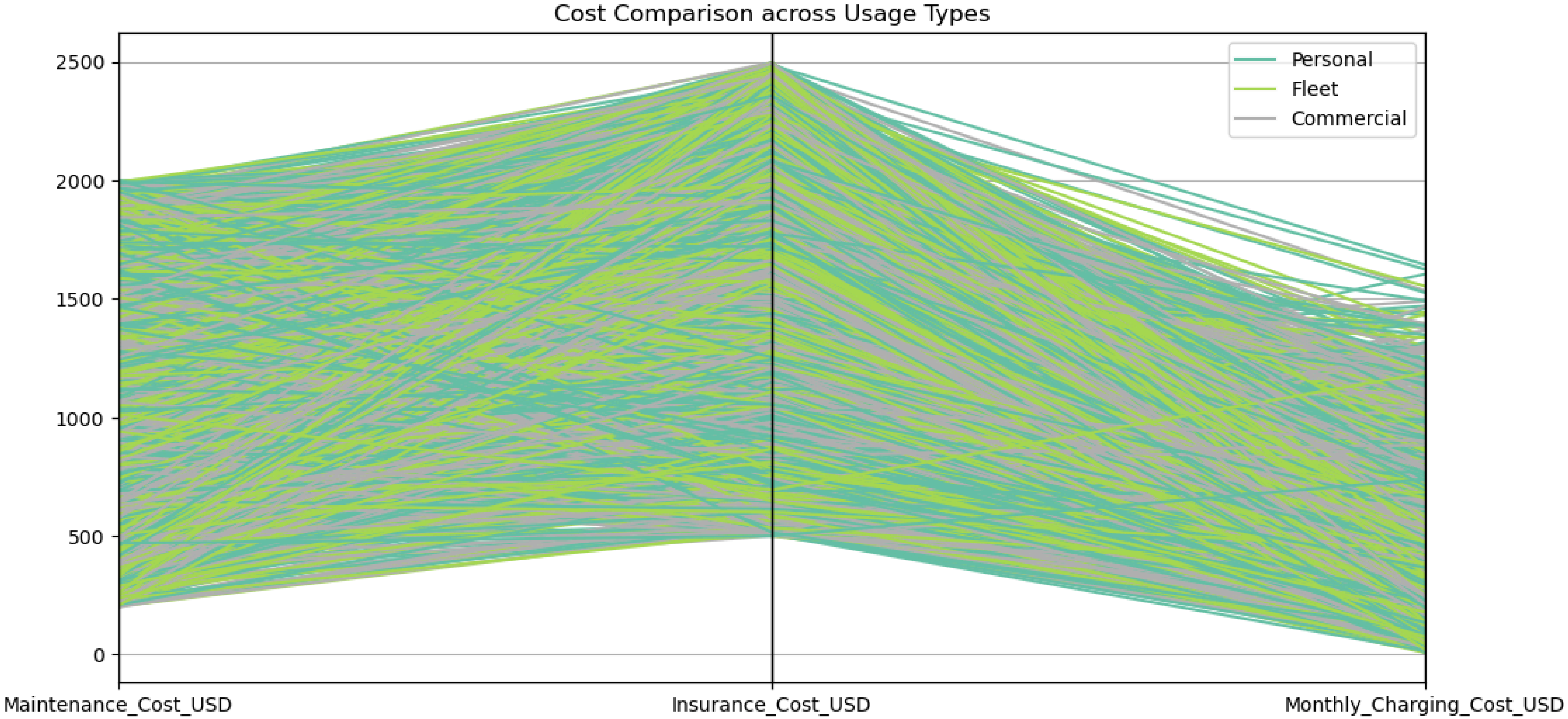

Performance Metrics Pairplot
Range_km
Avg_Speed_kmh
Max_Speed_kmh
100
200
300
400
500
600
700
40
60
80
100
140
160
180
200
220
240



The pairplot of performance metrics shows consistent relationships across Range, Average Speed, Maximum Speed, and Acceleration, indicating similar distribution patterns among the vehicles for these key performance indicators.

In [42]:

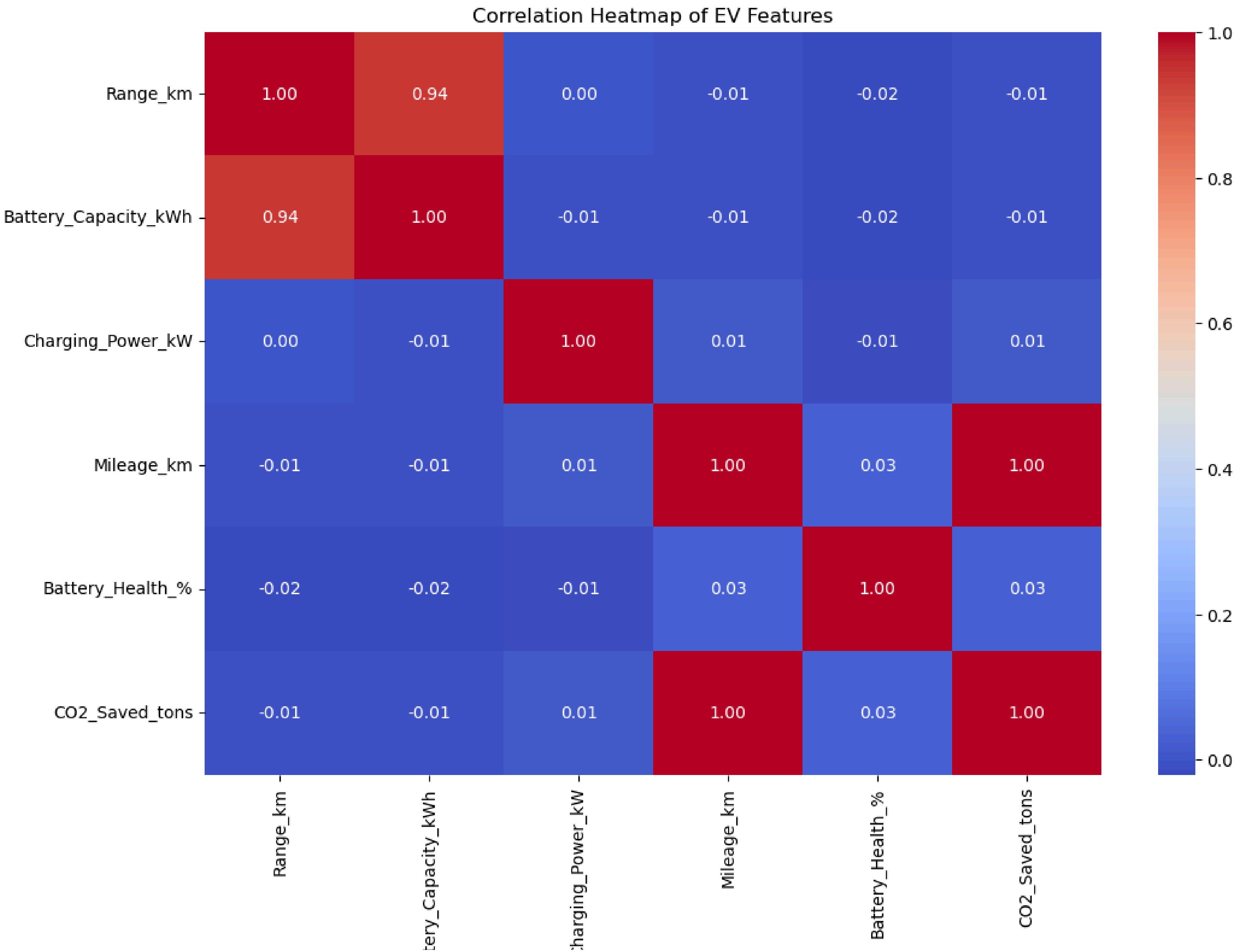
```
# Parallel Coordinates for key cost factors
from pandas.plotting import parallel_coordinates
subset = data[["Usage_Type","Maintenance_Cost_USD","Insurance_Cost_USD","Monthly_Charging_Cost_USD"]].dropna()
plt.figure(figsize=(12,6))
parallel_coordinates(subset, class_column="Usage_Type", colormap=plt.get_cmap("Set2"))
plt.title("Cost Comparison across Usage Types")
plt.show()
```



The parallel coordinates plot indicates that Insurance Costs are generally higher than Maintenance Costs, while Monthly Charging Costs tend to decrease, highlighting the relative scale of key expenses across vehicles.

In [44]:
```
# Correlation heatmap of important features
plt.figure(figsize=(12,8))
sns.heatmap(data[["Range_km","Battery_Capacity_kWh","Charging_Power_kW",
                  "Mileage_km","Battery_Health_%","CO2_Saved_tons"]].corr(),
            annot=True, cmap="coolwarm", fmt=".2f")
```

```
plt.title("Correlation Heatmap of EV Features")
plt.show()
```

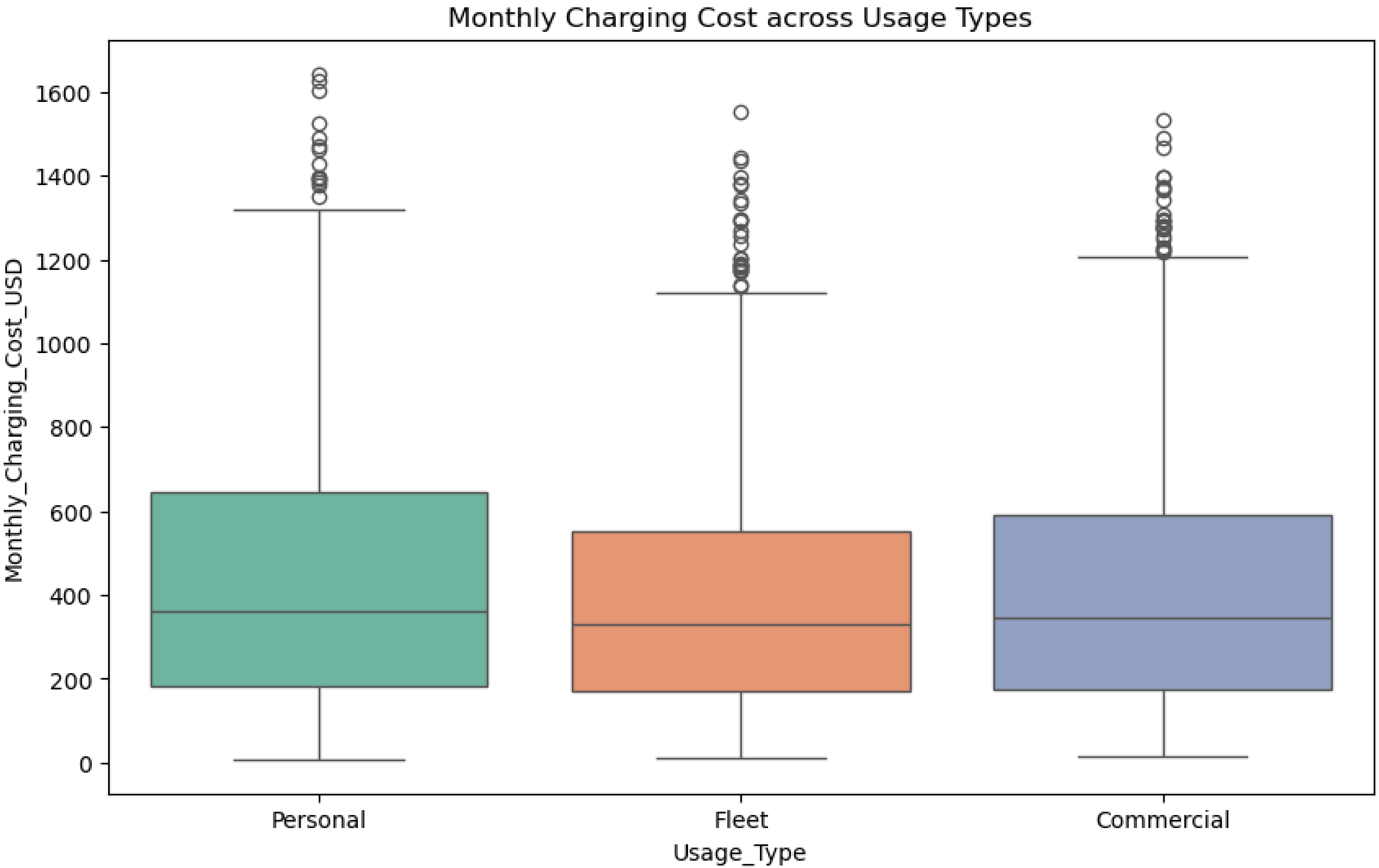


Correlation Heatmap of EV Features
Range_km
Battery_Capacity_kWh
Charging_Power_kW
Mileage_km
Battery_Health_%
CO2_Saved_tons
1.00
0.94
0.00
-0.01
-0.02
0.01
0.03
1.0
0.8
0.6
0.4
0.2
0.0

The correlation heatmap reveals a strong positive correlation (0.94) between Battery Capacity and Range, indicating that vehicles with larger batteries tend to achieve longer ranges. Other features show relatively weaker correlations.

In [46]:

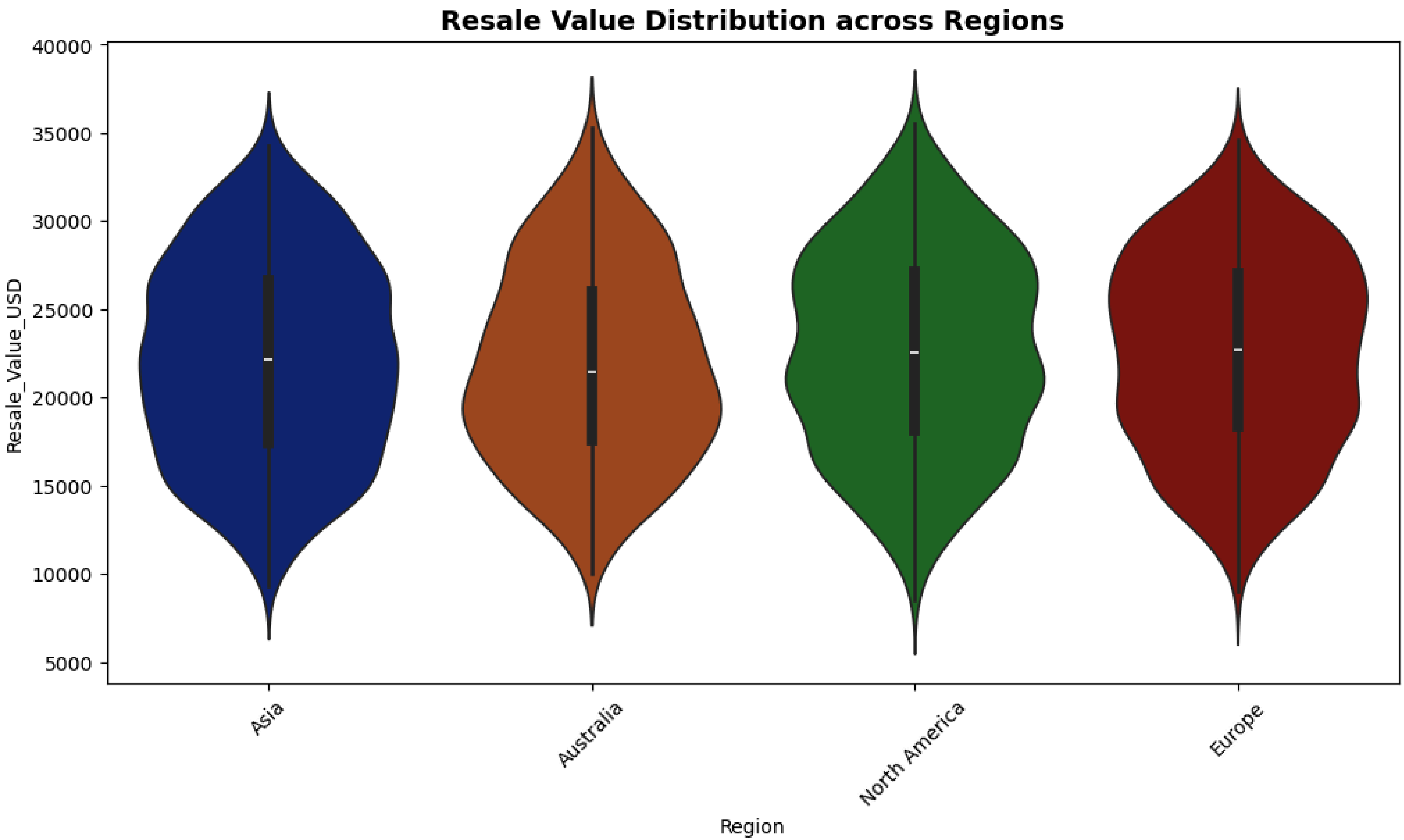
```
# Boxplot for Monthly Charging Cost by Usage Type
plt.figure(figsize=(10,6))
sns.boxplot(x="Usage_Type", y="Monthly_Charging_Cost_USD", data=data, palette="Set2")
plt.title("Monthly Charging Cost across Usage Types")
plt.show()
```



The boxplot shows that Personal vehicles generally incur higher Monthly Charging Costs compared to Commercial and Fleet vehicles.

In [48]:

```
plt.figure(figsize=(12,6))
sns.violinplot(
    x="Region",
    y="Resale_Value_USD",
    data=data,
    palette="dark",      # darker shades
    inner="box"          # dark box inside instead of quartile lines
)
plt.title("Resale Value Distribution across Regions", fontsize=14, fontweight="bold")
plt.xticks(rotation=45)
plt.show()
```

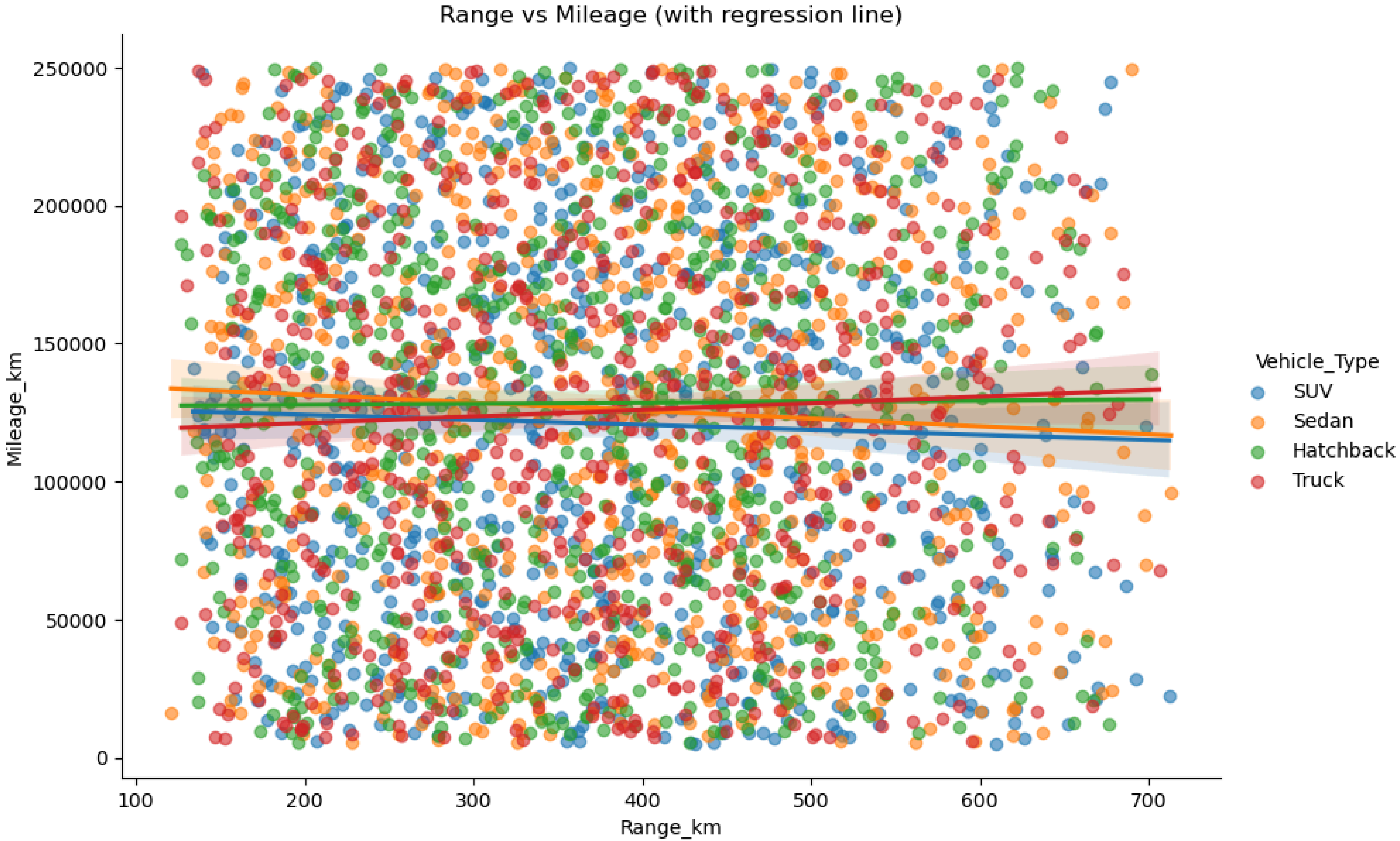


The violin plot indicates that Resale Values are fairly consistent across all regions, with similar distributions observed for each, suggesting limited regional variation in vehicle resale prices.

Advanced Analysis

In [51]:
```
# Scatter with regression line
plt.figure(figsize=(10,6))
sns.lmplot(x="Range_km", y="Mileage_km", hue="Vehicle_Type", data=data,
           aspect=1.5, height=6, scatter_kws={'alpha':0.6})
plt.title("Range vs Mileage (with regression line)")
plt.show()
```

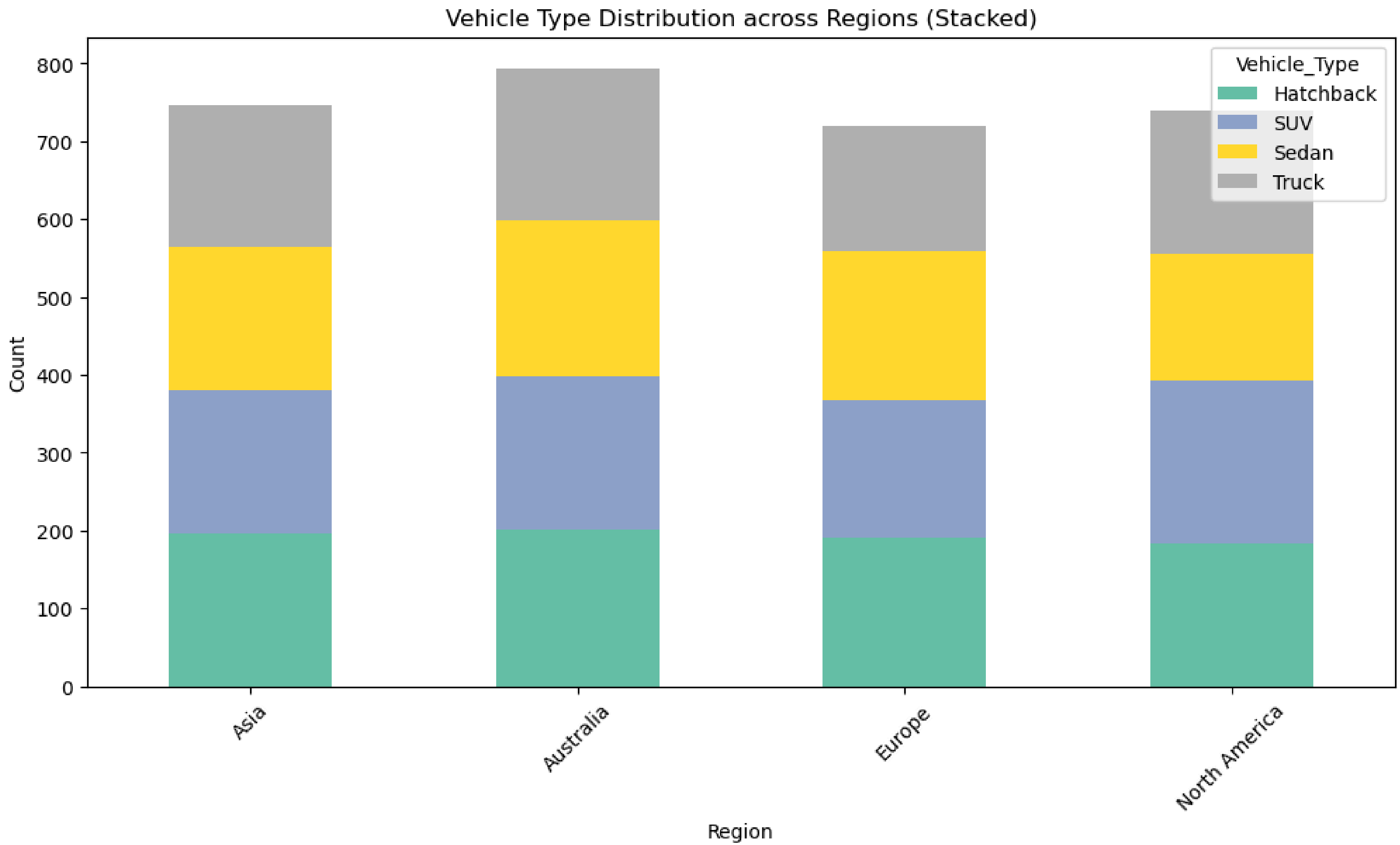
```
<Figure size 1000x600 with 0 Axes>
```



The scatter plot with regression lines shows that Trucks exhibit a positive trend, where higher Range corresponds to higher Mileage, while other Vehicle Types display a slight decreasing trend.

In [53]:

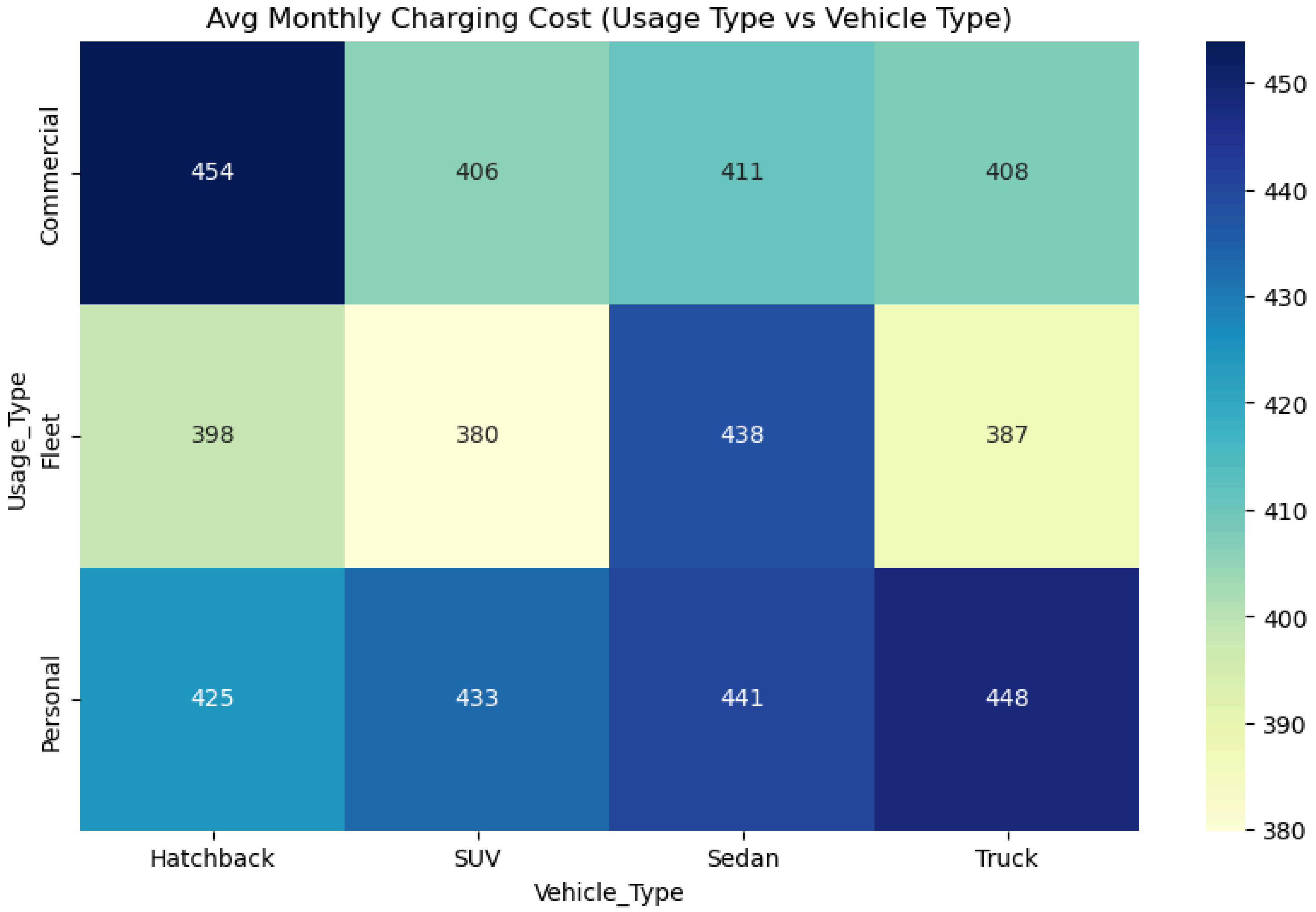
```python
# Stacked bar plot of Vehicle Type by Region
cross_tab = pd.crosstab(data["Region"], data["Vehicle_Type"])
cross_tab.plot(kind="bar", stacked=True, figsize=(12,6), colormap="Set2")
plt.title("Vehicle Type Distribution across Regions (Stacked)")
plt.xlabel("Region")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()
```



The stacked bar plot shows that Australia has a higher count across most Vehicle Types compared to other regions, indicating a more diverse vehicle presence in that region.

In [55]:

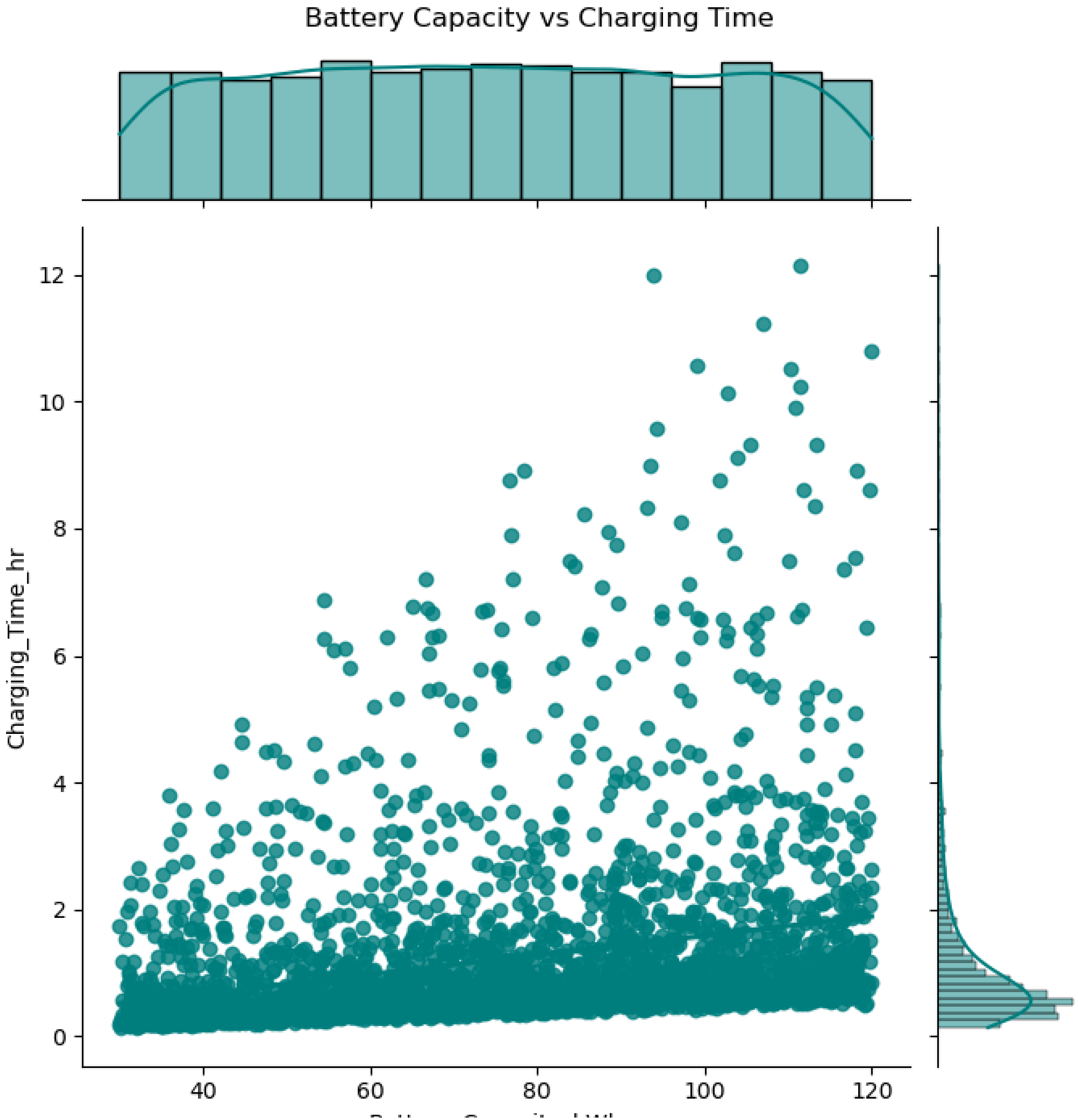
```python
# Heatmap of average monthly charging cost by Usage Type & Vehicle Type
pivot_table = data.pivot_table(values="Monthly_Charging_Cost_USD",
                               index="Usage_Type", columns="Vehicle_Type", aggfunc="mean")
plt.figure(figsize=(10,6))
sns.heatmap(pivot_table, annot=True, fmt=".0f", cmap="YlGnBu")
plt.title("Avg Monthly Charging Cost (Usage Type vs Vehicle Type)")
plt.show()
```



The heatmap of average monthly charging costs shows that Commercial vehicles incur the highest cost with Hatchbacks (≈454), followed by Sedans, Trucks, and SUVs. For Fleet usage, Sedans show the highest cost (≈438). In Personal usage, Trucks have the highest cost (≈448), while Hatchbacks incur the lowest (≈425).

In [57]:

```python
# Jointplot with regression
sns.jointplot(x="Battery_Capacity_kWh", y="Charging_Time_hr",
              data=data, kind="reg", height=7, color="teal")
plt.suptitle("Battery Capacity vs Charging Time", y=1.02)
plt.show()
```

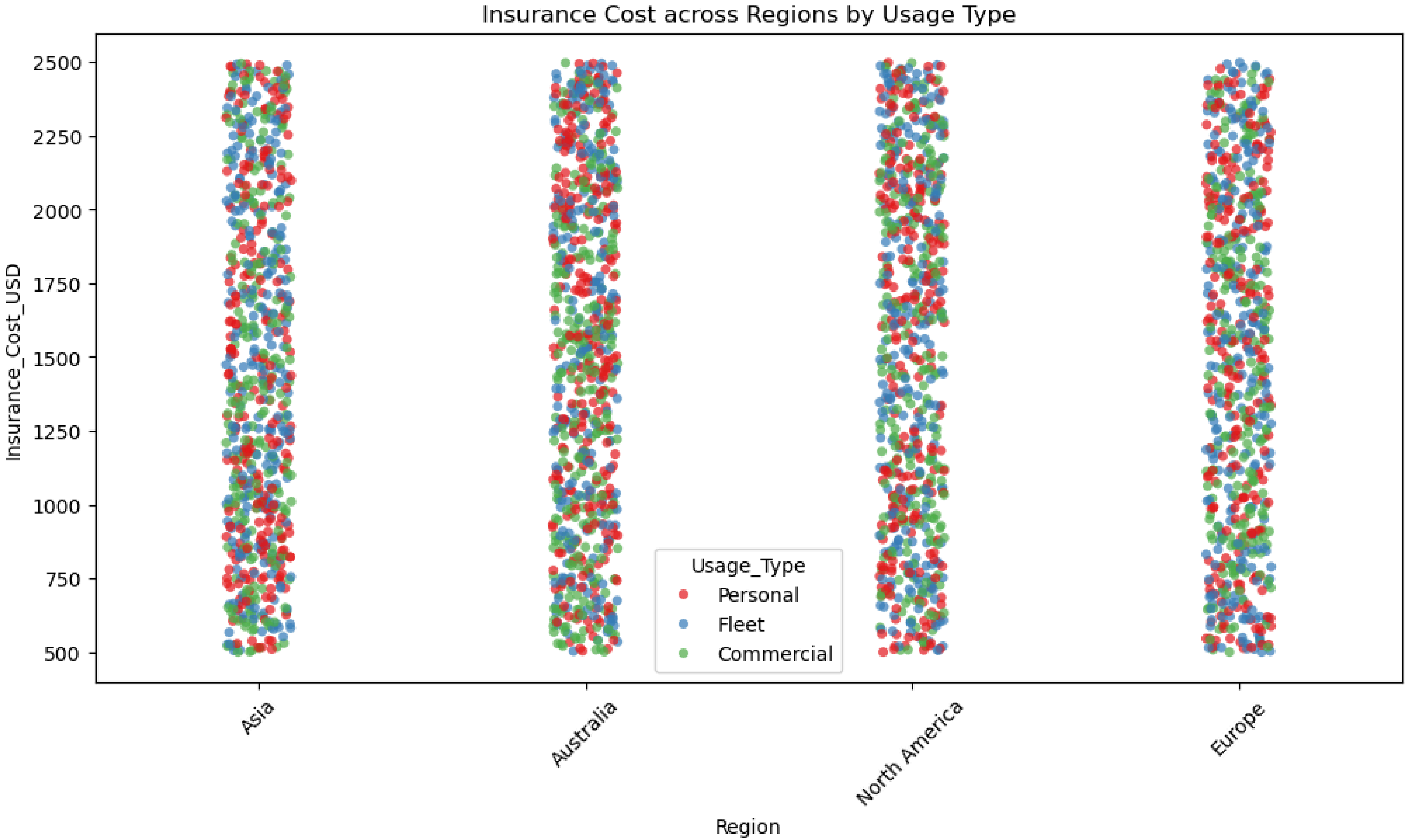


Battery Capacity vs Charging Time
Charging_Time_hr
0
2
4
6
8
10
12
40
60
80
100
120

Battery_Capacity_kWh

The jointplot shows that most vehicles have a Charging Time between 0 to 2 hours, and as Battery Capacity increases, the Charging Time also tends to rise.

In [59]:
```
# Stripplot
plt.figure(figsize=(12,6))
sns.stripplot(x="Region", y="Insurance_Cost_USD", hue="Usage_Type",
              data=data, jitter=True, alpha=0.7, palette="Set1")
plt.title("Insurance Cost across Regions by Usage Type")
plt.xticks(rotation=45)
plt.show()
```



The stripplot indicates that Insurance Costs are fairly similar in scale across regions such as Asia, Australia, North America, and Europe, with no major regional variations observed.

Insights

📊 Key Insights from Electric Vehicle Data Analysis

🔋 Battery & Range Insights

- Vehicles with **higher battery capacities achieve longer ranges**, showing a strong positive correlation (**0.94**).
- **Battery Health (%)** distribution is not bell-shaped, suggesting varying wear patterns.
- **Trucks** show a clear positive trend between range and mileage, while other vehicle types show a slight decline.

🚗 Vehicle Type & Regional Trends

- **Australia leads** with higher counts across almost all vehicle types.
- **SUVs dominate in North America**, while **Hatchbacks dominate in Asia and Australia**.
- **Resale values are fairly consistent** across regions, with no major differences.

⚡ Charging & Performance Insights

- Most vehicles complete charging in **0–2 hours**, with charging time increasing as battery capacity grows.
- **Charging Time** remains relatively uniform across vehicle types.
- **Performance metrics** (range, speed, acceleration) show similar scatter patterns, indicating comparable characteristics across models.

🌍 Usage Type Insights

- **Personal usage dominates**, followed by **Commercial** and **Fleet** vehicles.
- **Personal vehicles incur higher monthly charging costs**.
- **Commercial Hatchbacks** show the **highest average charging cost (~454 USD)**.

- **Fleet Sedans** show the **highest charging cost (~438 USD)**.

💰 Cost & Efficiency Insights

- **Insurance costs are higher** than maintenance costs, while charging costs remain relatively lower.
- Insurance costs remain **consistent across regions**.
- Higher mileage leads to **greater CO_2 savings**, underlining the environmental impact of EVs.

Recommendations

✅ Recommendations for Electric Vehicle Strategy

🔋 Battery & Range

- Invest in **larger battery capacities** to improve range, as capacity and range show a strong positive correlation.
- Focus on **battery health monitoring programs** to minimize performance drops and resale value impact.

🚗 Vehicle Type & Regional Focus

- **Expand SUV offerings in North America**, where demand is strong.
- **Promote hatchbacks in Asia and Australia**, aligning with consumer preferences.
- Maintain **consistent resale strategies** across regions, as resale values show minimal variation.

⚡ Charging & Infrastructure

- Develop **fast-charging solutions**, since most vehicles already charge within 0–2 hours, but higher-capacity batteries need faster support.
- Encourage **standardized charging networks** across regions to reduce variability in user experience.

🌍 Usage Patterns & Costs

- Design **personal-use focused EV models**, as personal vehicles dominate usage.
- Optimize **charging costs for commercial fleets** (especially hatchbacks and sedans) to attract business buyers.
- Offer **insurance + maintenance bundles** to provide cost predictability and attract cost-sensitive customers.

💡 Strategic Outlook

- Highlight **environmental benefits** (CO_2 savings) in marketing to strengthen EV adoption.
- Leverage **data-driven insights** to tailor product portfolios region-wise and usage-wise.
- Continue analyzing **fleet vehicle costs** to identify opportunities for efficiency improvements.