M Koushik

2211CS010343

Group-4

DATASET DESCRIPTION:

The **Hubli Route Division dataset** is a collection of travel routes connecting different places, with Hubballi serving as a major hub. It includes details of 360 routes, showing where each journey starts (`From`), where it ends (`To`), and how far the two locations are (`ROUTE Length`). The shortest routes are around 20 km, while the longest stretches over 200 km.

This dataset gives a clear picture of how Hubballi is connected to various towns and cities. Since there are no missing values, it's a reliable source for analyzing travel patterns, identifying frequently used routes, and even planning better transportation options. Whether you're looking to understand the region's connectivity or optimize travel routes, this data provides valuable insights into how people move around this network.

In [15]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [16]:
```
df = pd.read_csv(r"C:\Users\Saiko\Downloads\Hubli_division_route_details.csv")
```

In [17]:
```
df
```

Out[17]:

	Sl no	From	To	ROUTE Length
0	1	Hubballi	Annigeri	38
1	2	Hubballi	Alagawadi	52
2	3	Hubballi	Aralikatti	20
3	4	Hubballi	Belgam	100
4	5	Hubballi	Bagadageri	34
...	...	...	...	...
355	357	Mundagod	HLY	66
356	358	Nandigatti	HRBS	50
357	359	SIRSI	HUBLI	105
358	360	Hubballi	Panaji	204
359	361	Kalaghatagi	Mundagod	32

360 rows × 4 columns

In [18]:
```
df.info
```

Out[18]:
```
<bound method DataFrame.info of      Sl no        From         To  ROUTE Length
0        1    Hubballi    Annigeri            38
1        2    Hubballi   Alagawadi            52
2        3    Hubballi  Aralikatti            20
3        4    Hubballi      Belgam           100
4        5    Hubballi  Bagadageri            34
..     ...         ...         ...           ...
355    357    Mundagod         HLY            66
356    358  Nandigatti        HRBS            50
357    359       SIRSI       HUBLI           105
358    360    Hubballi      Panaji           204
359    361  Kalaghatagi    Mundagod            32

[360 rows x 4 columns]>
```

Checking the null values

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
Sl no           0
From            0
To              0
ROUTE Length    0
dtype: int64
```

Dropping the null values

In [24]:
```
df = df.dropna()
```

Filling the null values using the median method

In [26]:
```
df['ROUTE Length'] = df['ROUTE Length'].fillna(df['ROUTE Length'].median())
```

Checking for the duplicate entries and dropping the Duplicate entries in the data set

In [27]:
```
print(df.duplicated().sum())
df = df.drop_duplicates()
```

```
0
```

Head of the dataset

In [28]:
```
df.head()
```

Out[28]:

	Sl no	From	To	ROUTE Length
0	1	Hubballi	Annigeri	38
1	2	Hubballi	Alagawadi	52
2	3	Hubballi	Aralikatti	20
3	4	Hubballi	Belgam	100
4	5	Hubballi	Bagadageri	34

Tail of the dataset

In [30]:
```
df.tail()
```

Out[30]:

	Sl no	From	To	ROUTE Length
355	357	Mundagod	HLY	66
356	358	Nandigatti	HRBS	50
357	359	SIRSI	HUBLI	105
358	360	Hubballi	Panaji	204
359	361	Kalaghatagi	Mundagod	32

Checking the datatypes of the dataset if all the data types are correct or not

In [33]:
```
print(df.dtypes)
```

```
Sl no            int64
From            object
To              object
ROUTE Length     int64
dtype: object
```

Describing the dataset with various factors and all checkpoints

In [34]:
```
print(df.describe())
```

```
             Sl no  ROUTE Length
count  360.000000    360.000000
mean   180.888889     96.013889
std    104.479817    128.550792
min      1.000000      5.000000
25%     90.750000     28.000000
50%    180.500000     43.000000
75%    271.250000     92.500000
max    361.000000    658.000000
```

Coverting the kms into the miles

In [37]:
```
df['ROUTE Length (miles)'] = df['ROUTE Length'] * 0.621
print(df.head())
```

```
   Sl no      From         To  ROUTE Length  ROUTE Length (miles)
0      1  Hubballi    Annigeri            38                23.598
1      2  Hubballi   Alagawadi            52                32.292
2      3  Hubballi  Aralikatti            20                12.420
3      4  Hubballi      Belgam           100                62.100
4      5  Hubballi  Bagadageri            34                21.114
```

In [38]:
```
df
```

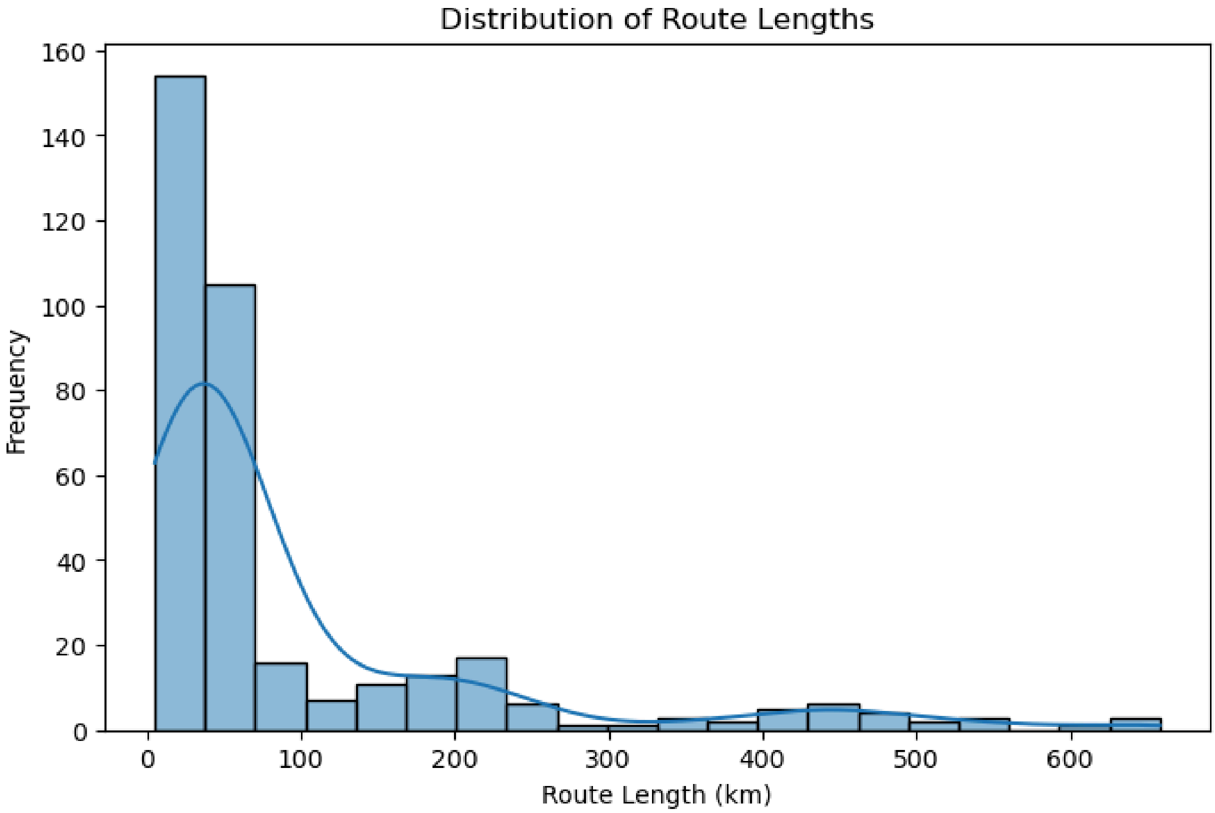
Out[38]:

	Sl no	From	To	ROUTE Length	ROUTE Length (miles)
0	1	Hubballi	Annigeri	38	23.598
1	2	Hubballi	Alagawadi	52	32.292
2	3	Hubballi	Aralikatti	20	12.420
3	4	Hubballi	Belgam	100	62.100
4	5	Hubballi	Bagadageri	34	21.114
...	...	...	...	...	...
355	357	Mundagod	HLY	66	40.986
356	358	Nandigatti	HRBS	50	31.050
357	359	SIRSI	HUBLI	105	65.205
358	360	Hubballi	Panaji	204	126.684
359	361	Kalaghatagi	Mundagod	32	19.872

360 rows × 5 columns

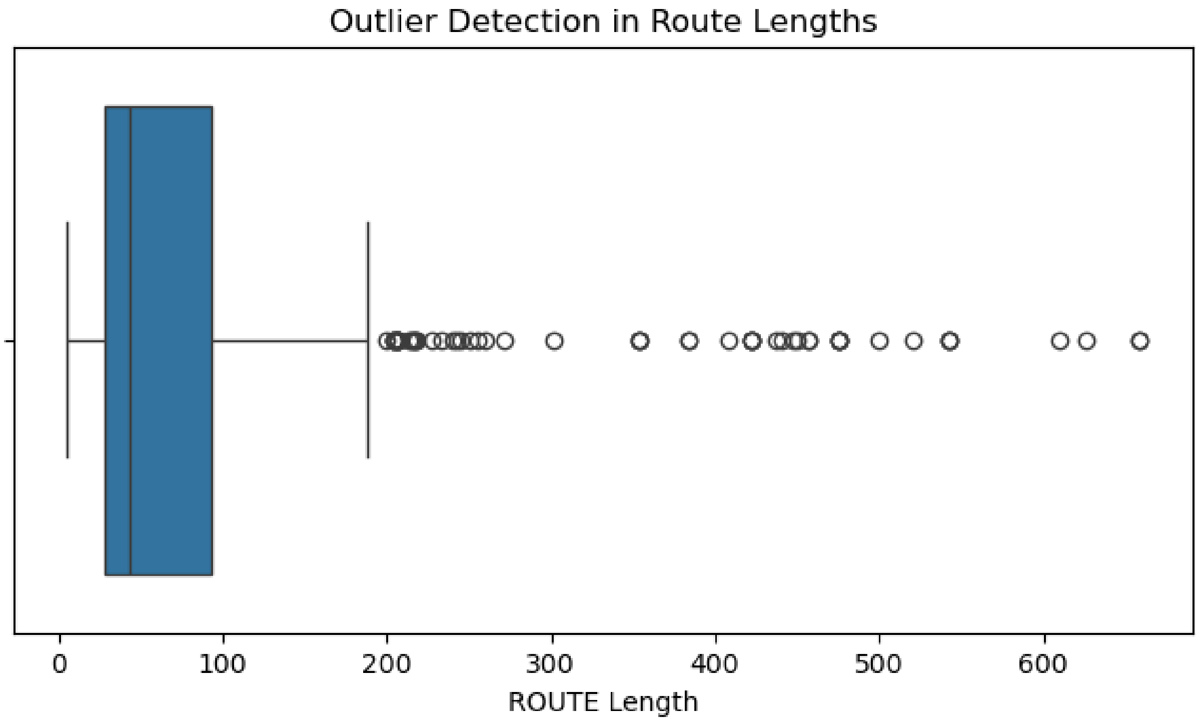
Showing the route length across the dataset

In [39]:
```
plt.figure(figsize=(8, 5))
sns.histplot(df['ROUTE Length'], bins=20, kde=True)
plt.title("Distribution of Route Lengths")
plt.xlabel("Route Length (km)")
plt.ylabel("Frequency")
plt.show()
```



Box plot for the Identifies extreme route lengths that may indicate anomalies or unusual travel patterns.

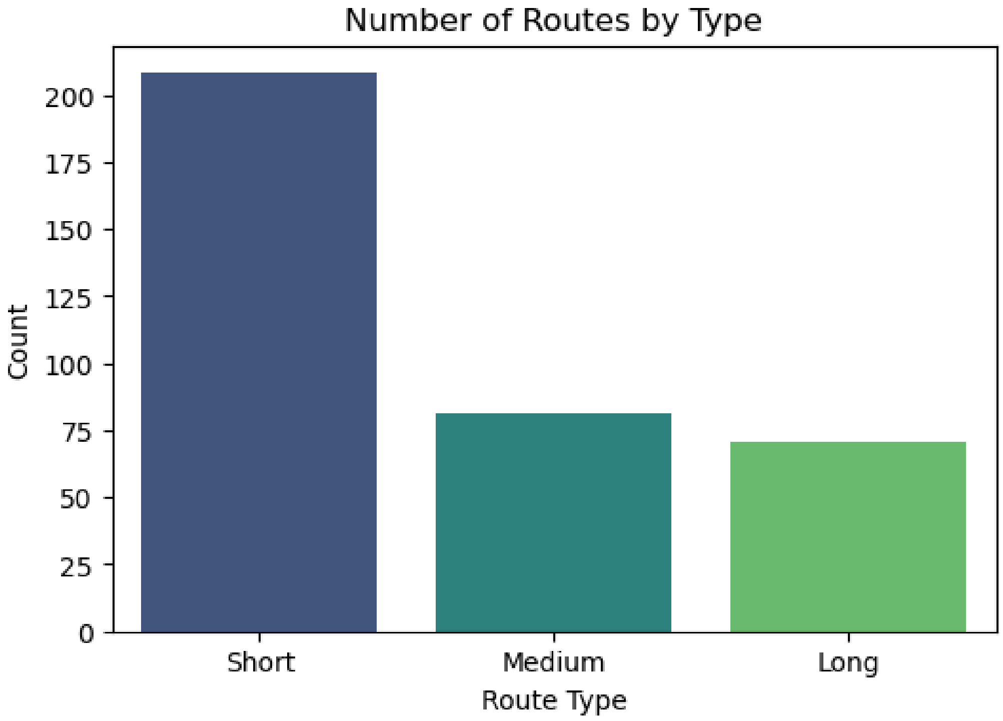
```
In [40]: plt.figure(figsize=(8, 4))
         sns.boxplot(x=df['ROUTE Length'])
         plt.title("Outlier Detection in Route Lengths")
         plt.show()
```



Visualizes the number of short, medium, and long routes to understand network structure.

In [42]:
```
df['Route Type'] = df['ROUTE Length'].apply(lambda x: "Short" if x < 50 else "Medium"
plt.figure(figsize=(6, 4))
sns.countplot(x=df['Route Type'], palette="viridis")
plt.title("Number of Routes by Type")
plt.xlabel("Route Type")
plt.ylabel("Count")
plt.show()
```

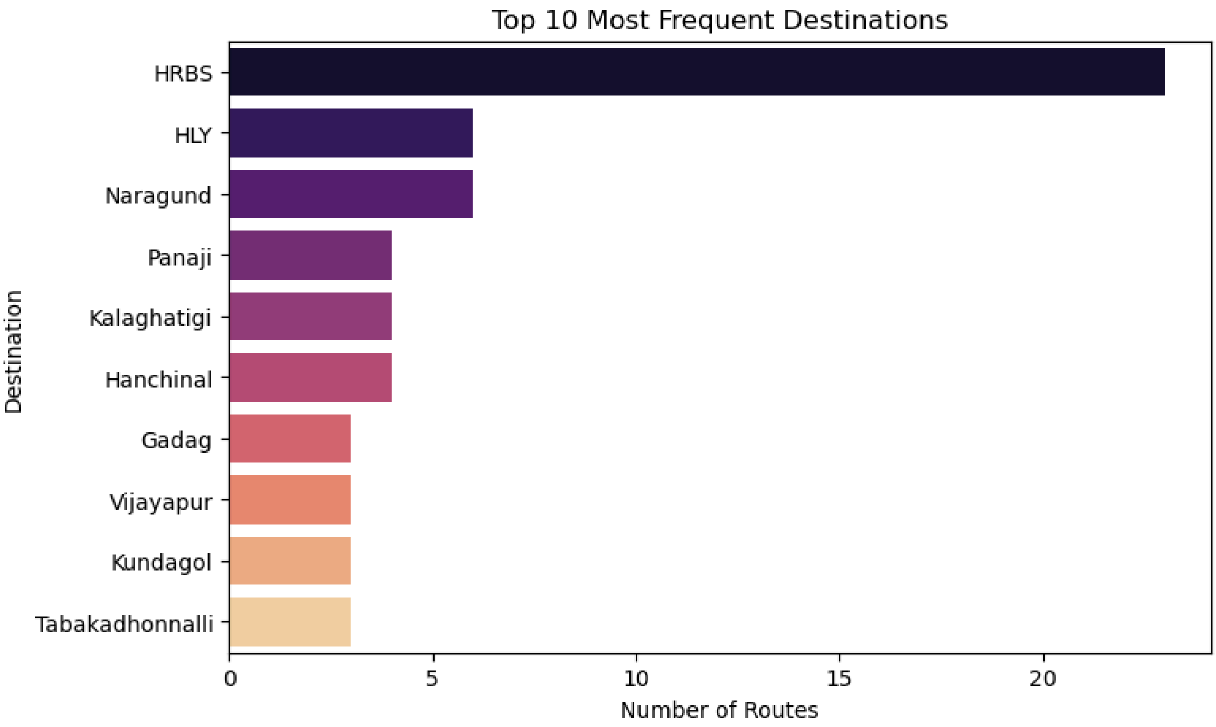
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\1781673237.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=df['Route Type'], palette="viridis")
```



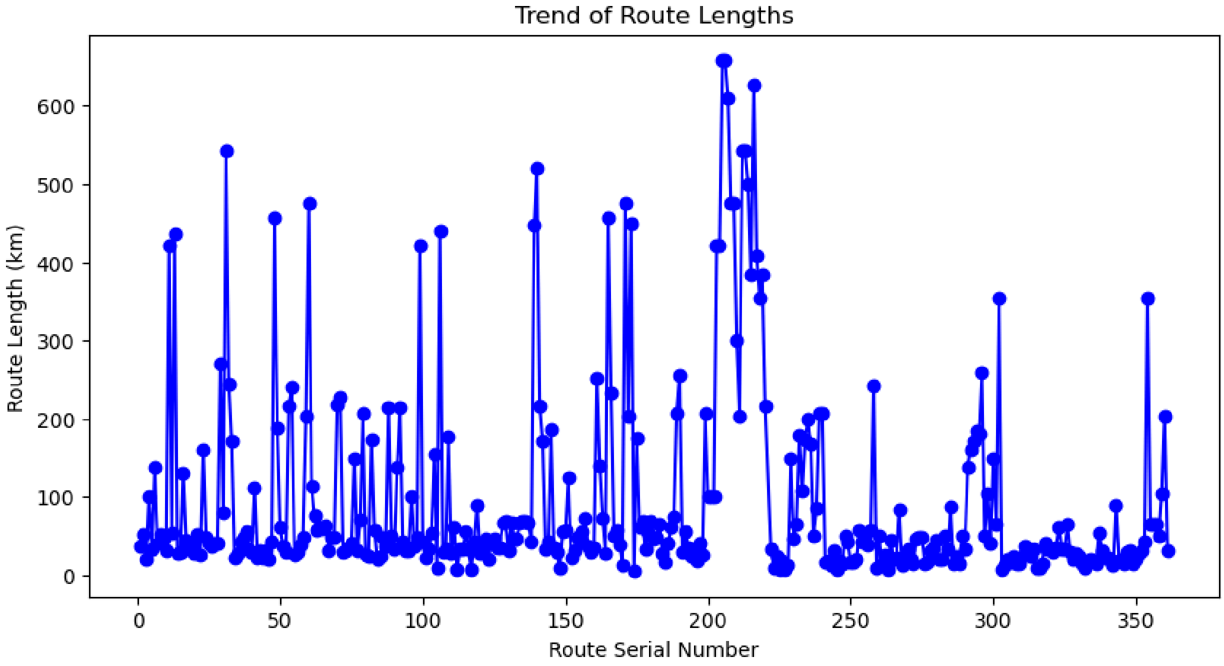
Top 10 most frequent destinations visited

In [43]:
```
top_destinations = df['To'].value_counts().head(10)

plt.figure(figsize=(8, 5))
sns.barplot(x=top_destinations.values, y=top_destinations.index, palette="magma")
plt.title("Top 10 Most Frequent Destinations")
plt.xlabel("Number of Routes")
plt.ylabel("Destination")
plt.show()
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\4233659809.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_destinations.values, y=top_destinations.index, palette="magma")
```

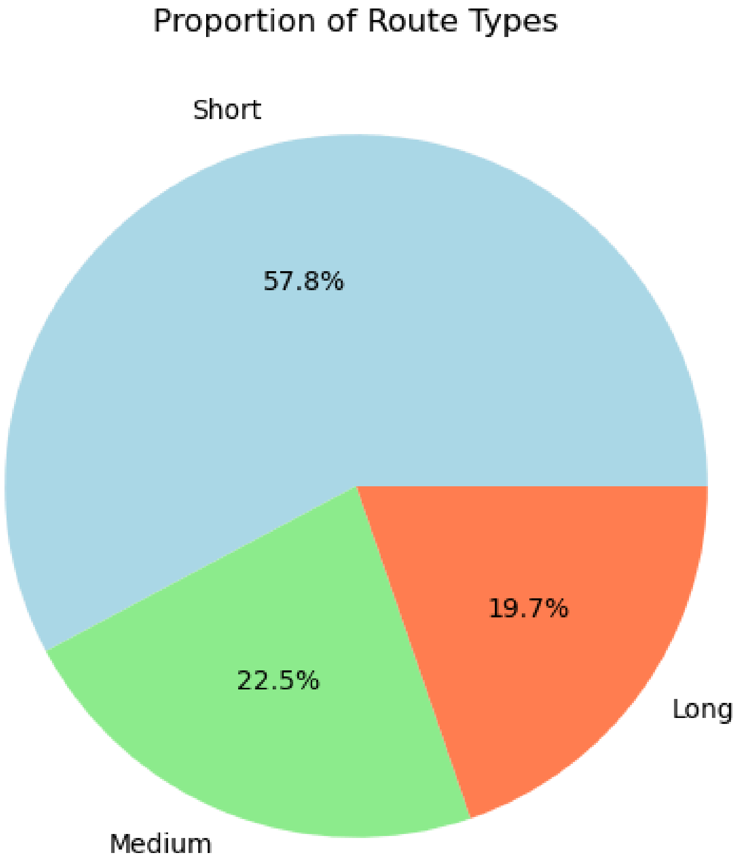


```
In [44]: plt.figure(figsize=(10, 5))
         plt.plot(df['Sl no'], df['ROUTE Length'], marker='o', linestyle='-', color='b')
         plt.title("Trend of Route Lengths")
         plt.xlabel("Route Serial Number")
         plt.ylabel("Route Length (km)")
         plt.show()
```



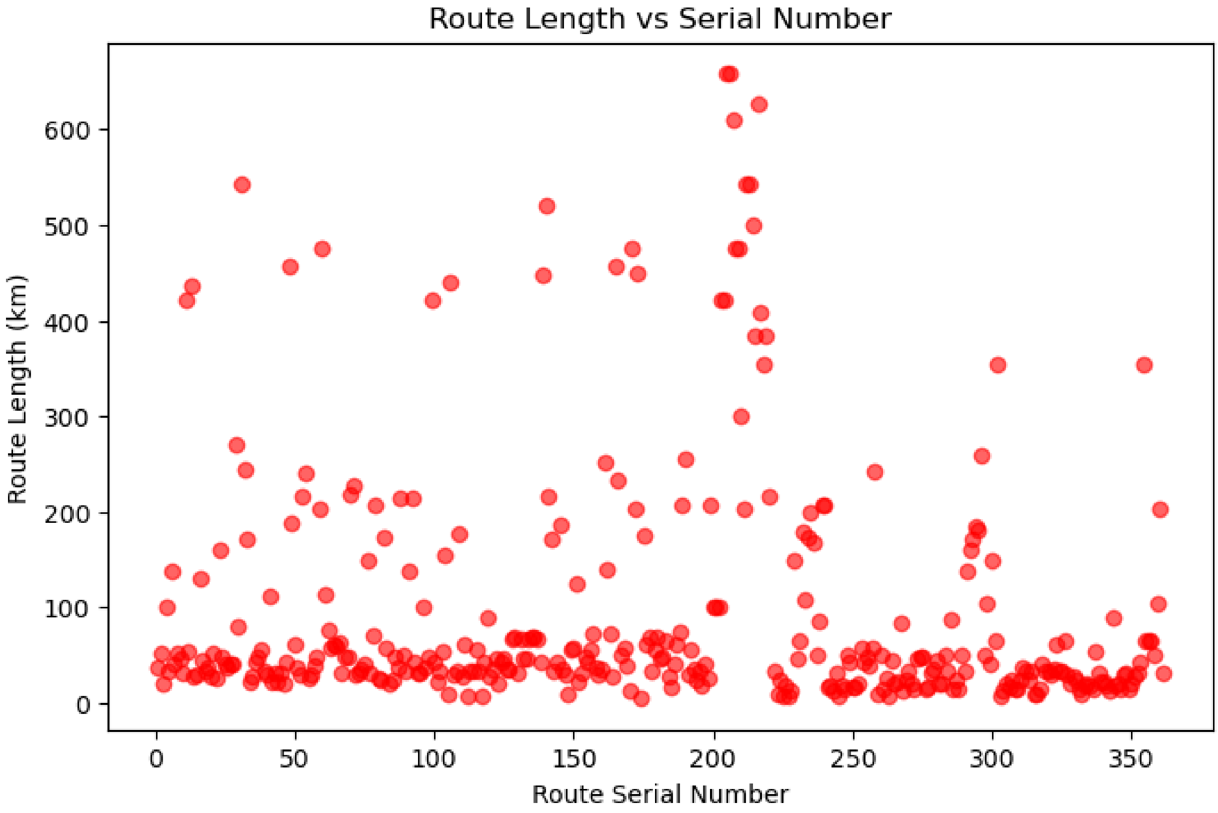
The percentage distribution of short, medium, and long routes.

In [46]:
```
route_counts = df['Route Type'].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(route_counts, labels=route_counts.index, autopct='%1.1f%%', colors=['lightblue
plt.title("Proportion of Route Types")
plt.show()
```



Plots route lengths against serial numbers to identify variations and clustering patterns.

In [48]:
```
plt.figure(figsize=(8, 5))
plt.scatter(df['Sl no'], df['ROUTE Length'], color='red', alpha=0.6)
plt.title("Route Length vs Serial Number")
plt.xlabel("Route Serial Number")
plt.ylabel("Route Length (km)")
plt.show()
```



In [49]:
```
df
```

Out[49]:

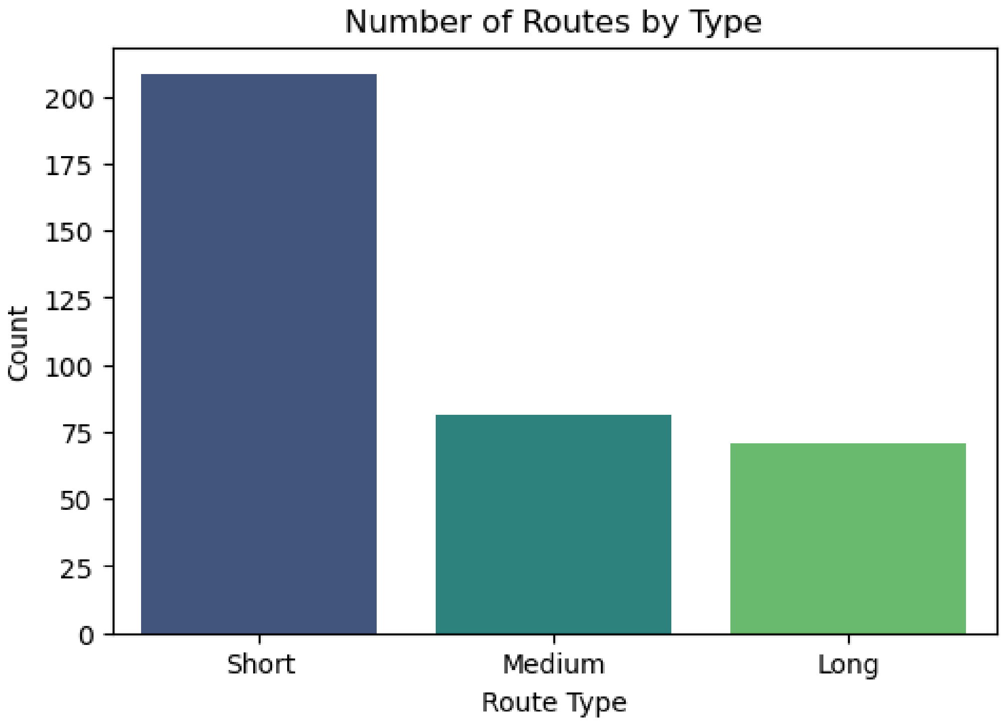
	Sl no	From	To	ROUTE Length	ROUTE Length (miles)	Route Type
0	1	Hubballi	Annigeri	38	23.598	Short
1	2	Hubballi	Alagawadi	52	32.292	Medium
2	3	Hubballi	Aralikatti	20	12.420	Short
3	4	Hubballi	Belgam	100	62.100	Medium
4	5	Hubballi	Bagadageri	34	21.114	Short
...	...	...	...	...	...	...
355	357	Mundagod	HLY	66	40.986	Medium
356	358	Nandigatti	HRBS	50	31.050	Medium
357	359	SIRSI	HUBLI	105	65.205	Medium
358	360	Hubballi	Panaji	204	126.684	Long
359	361	Kalaghatagi	Mundagod	32	19.872	Short

360 rows × 6 columns

In [50]:

```python
df['Route Type'] = df['ROUTE Length'].apply(lambda x: "Short" if x < 50 else "Medium"

plt.figure(figsize=(6, 4))
sns.countplot(x=df['Route Type'], palette="viridis")
plt.title("Number of Routes by Type")
plt.xlabel("Route Type")
plt.ylabel("Count")
plt.show()
```

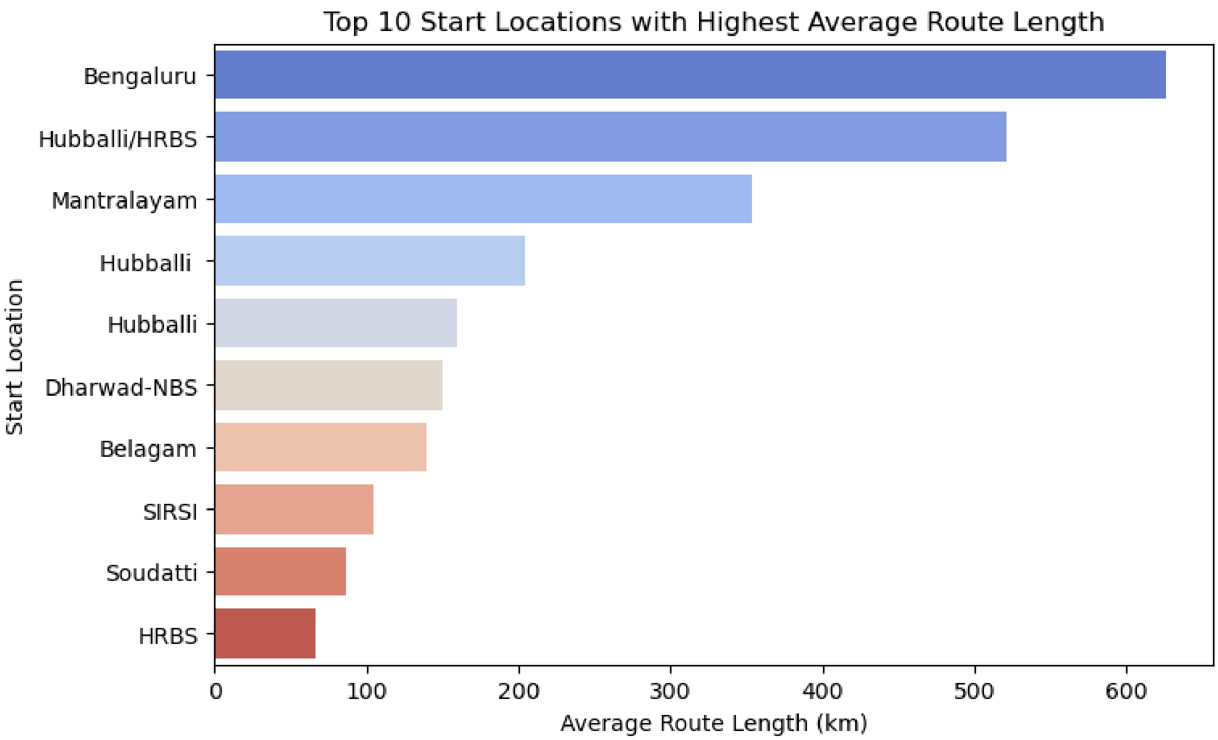
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\3764098485.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=df['Route Type'], palette="viridis")
```



In [52]:

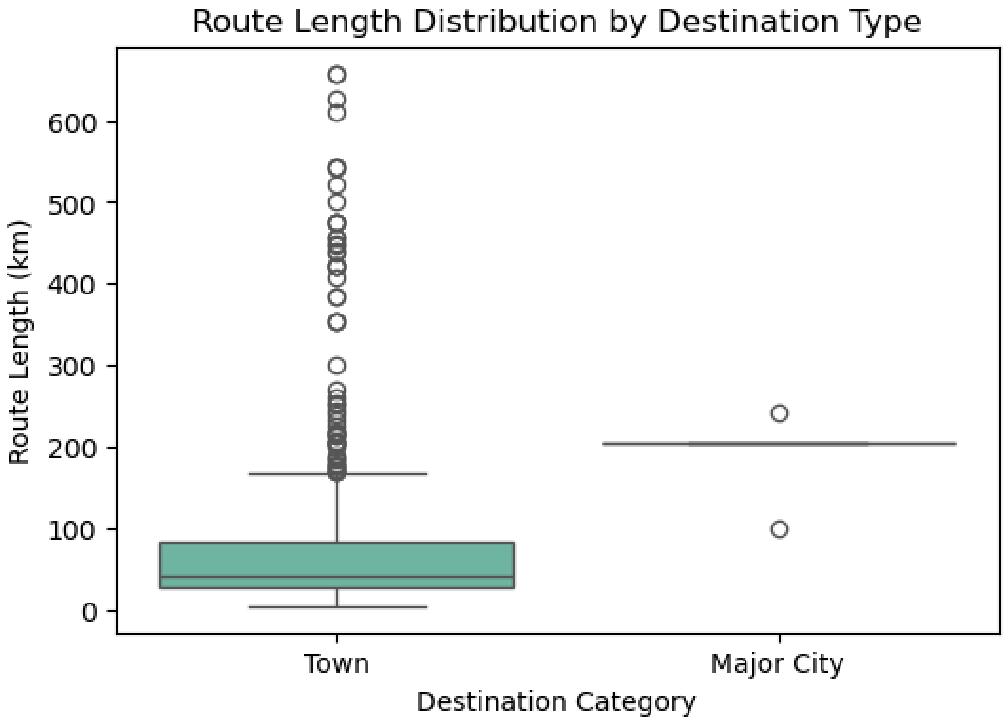
```
avg_route_length = df.groupby("From")["ROUTE Length"].mean().sort_values(ascending=Fal

plt.figure(figsize=(8, 5))
sns.barplot(x=avg_route_length.values, y=avg_route_length.index, palette="coolwarm")
plt.title("Top 10 Start Locations with Highest Average Route Length")
plt.xlabel("Average Route Length (km)")
plt.ylabel("Start Location")
plt.show()
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\2594836211.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=avg_route_length.values, y=avg_route_length.index, palette="coolwar
m")
```



In [53]:
```
df['Destination Category'] = df['To'].apply(lambda x: "Major City" if x in ['Hubballi'

plt.figure(figsize=(6, 4))
sns.boxplot(x=df['Destination Category'], y=df['ROUTE Length'], palette="Set2")
plt.title("Route Length Distribution by Destination Type")
plt.xlabel("Destination Category")
plt.ylabel("Route Length (km)")
plt.show()
```

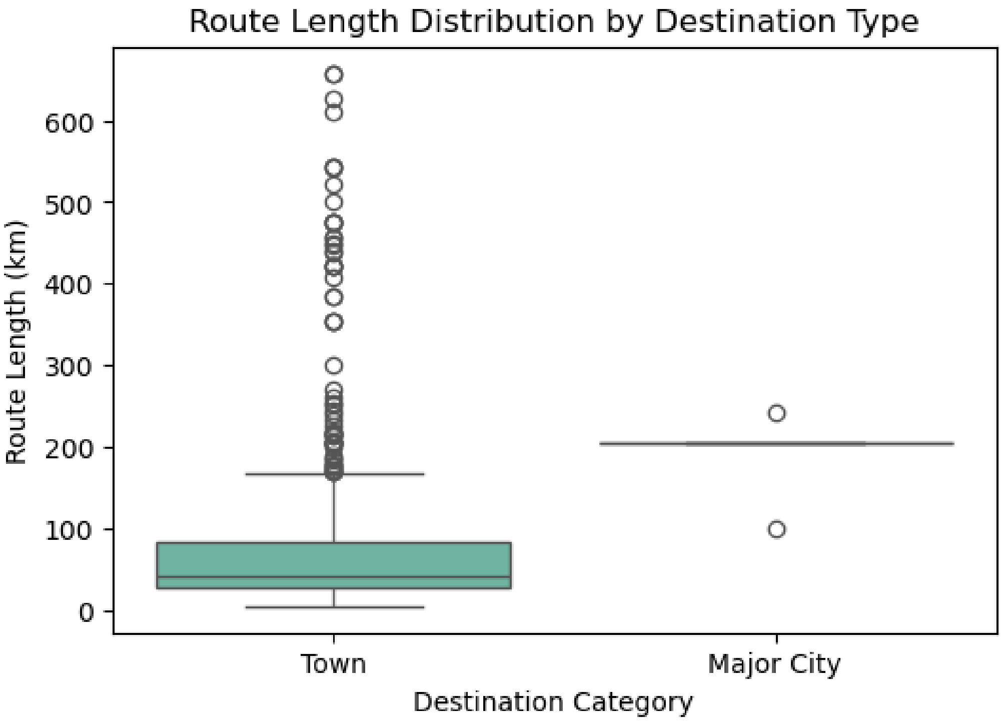
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\3359752712.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x=df['Destination Category'], y=df['ROUTE Length'], palette="Set2")
```



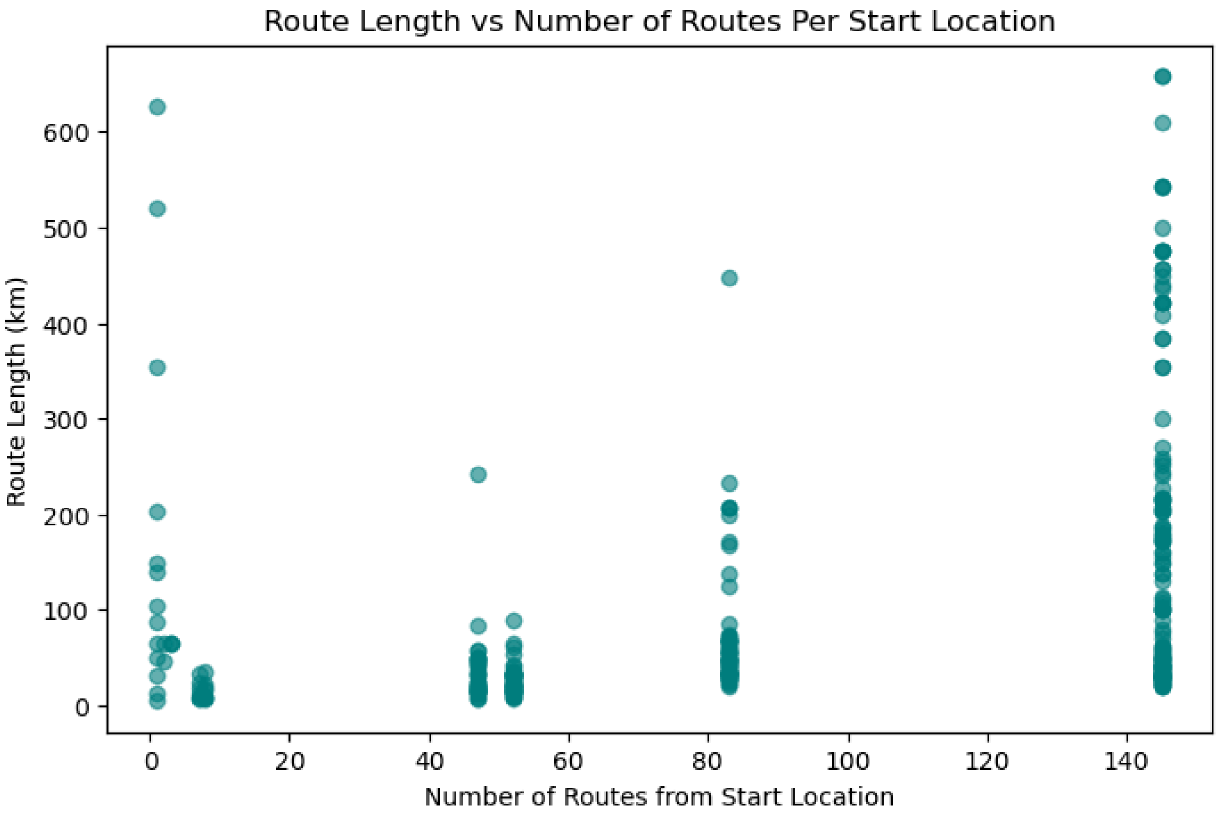
In [54]:

```
df['Destination Category'] = df['To'].apply(lambda x: "Major City" if x in ['Hubballi'

plt.figure(figsize=(6, 4))
sns.boxplot(x=df['Destination Category'], y=df['ROUTE Length'], palette="Set2")
plt.title("Route Length Distribution by Destination Type")
plt.xlabel("Destination Category")
plt.ylabel("Route Length (km)")
plt.show()
```

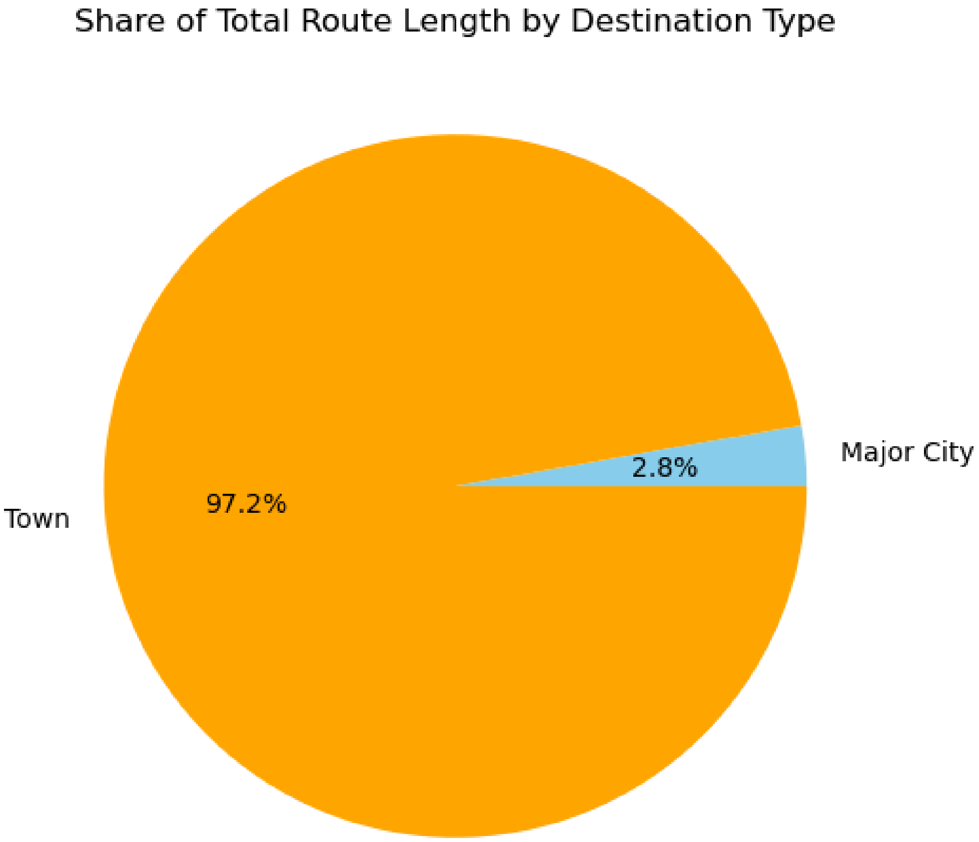
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\3359752712.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x=df['Destination Category'], y=df['ROUTE Length'], palette="Set2")
```



```
In [55]: start_counts = df['From'].value_counts()
         df['Start Count'] = df['From'].map(start_counts)

         plt.figure(figsize=(8, 5))
         plt.scatter(df['Start Count'], df['ROUTE Length'], alpha=0.6, color='teal')
         plt.title("Route Length vs Number of Routes Per Start Location")
         plt.xlabel("Number of Routes from Start Location")
         plt.ylabel("Route Length (km)")
         plt.show()
```



In [56]:
```
route_type_sums = df.groupby("Destination Category")["ROUTE Length"].sum()

plt.figure(figsize=(6, 6))
plt.pie(route_type_sums, labels=route_type_sums.index, autopct='%1.1f%%', colors=['sky
plt.title("Share of Total Route Length by Destination Type")
plt.show()
```

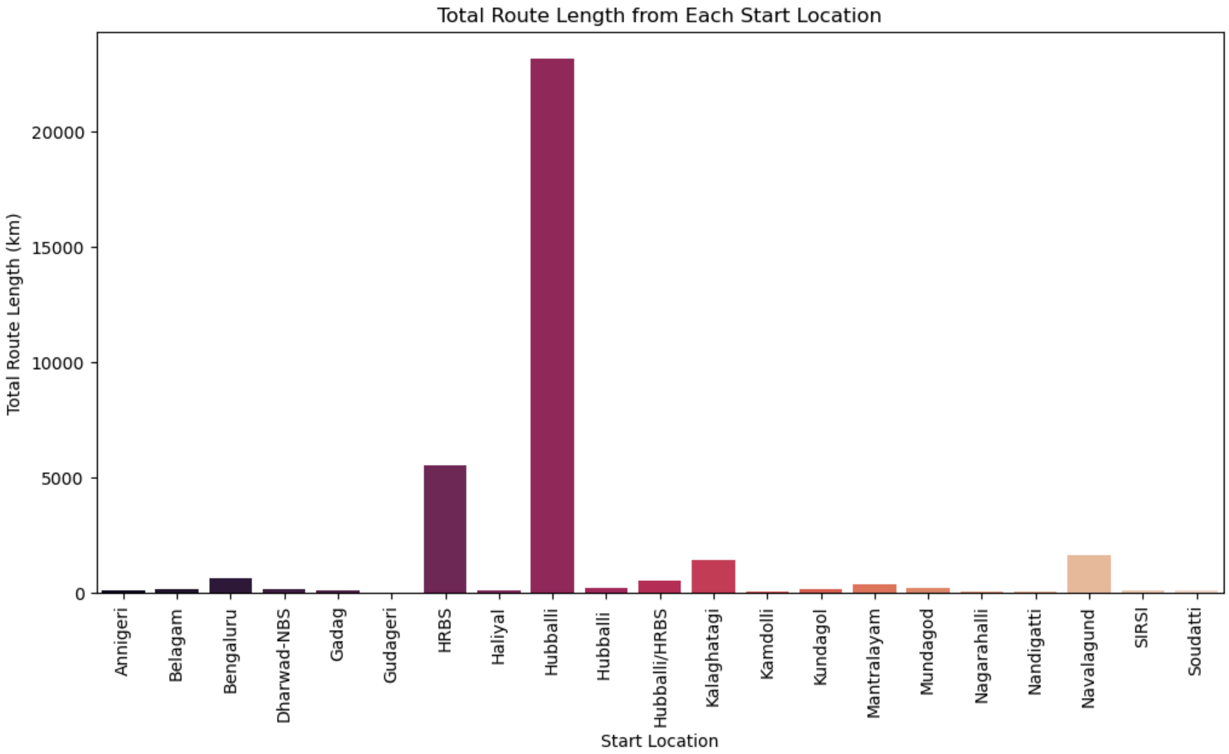


In [59]:
```
plt.figure(figsize=(1, 1))
sns.barplot(x=df.groupby("From")["ROUTE Length"].sum().index,
            y=df.groupby("From")["ROUTE Length"].sum().values,
            hue=df['Route Type'],
            palette="rocket")
plt.xticks(rotation=90)
plt.title("Total Route Length from Each Start Location (Grouped by Route Type)")
plt.xlabel("Start Location")
plt.ylabel("Total Route Length (km)")
plt.legend(title="Route Type")
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[59], line 2
      1 plt.figure(figsize=(1, 1))
----> 2 sns.barplot(x=df.groupby("From")["ROUTE Length"].sum().index,
      3             y=df.groupby("From")["ROUTE Length"].sum().values,
      4             hue=df['Route Type'],
      5             palette="rocket")
      6 plt.xticks(rotation=90)
      7 plt.title("Total Route Length from Each Start Location (Grouped by Route T
ype)")

File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:2341, in barplot(data,
x, y, hue, order, hue_order, estimator, errorbar, n_boot, seed, units, weights, or
ient, color, palette, saturation, fill, hue_norm, width, dodge, gap, log_scale, na
tive_scale, formatter, legend, capsize, err_kws, ci, errcolor, errwidth, ax, **kwa
rgs)
   2338 if estimator is len:
   2339     estimator = "size"
```

```
In [60]: df_grouped = df.groupby("From", as_index=False)["ROUTE Length"].sum()

         plt.figure(figsize=(12, 6))
         sns.barplot(x=df_grouped["From"], y=df_grouped["ROUTE Length"], palette="rocket")
         plt.xticks(rotation=90)
         plt.title("Total Route Length from Each Start Location")
         plt.xlabel("Start Location")
         plt.ylabel("Total Route Length (km)")
         plt.show()
```

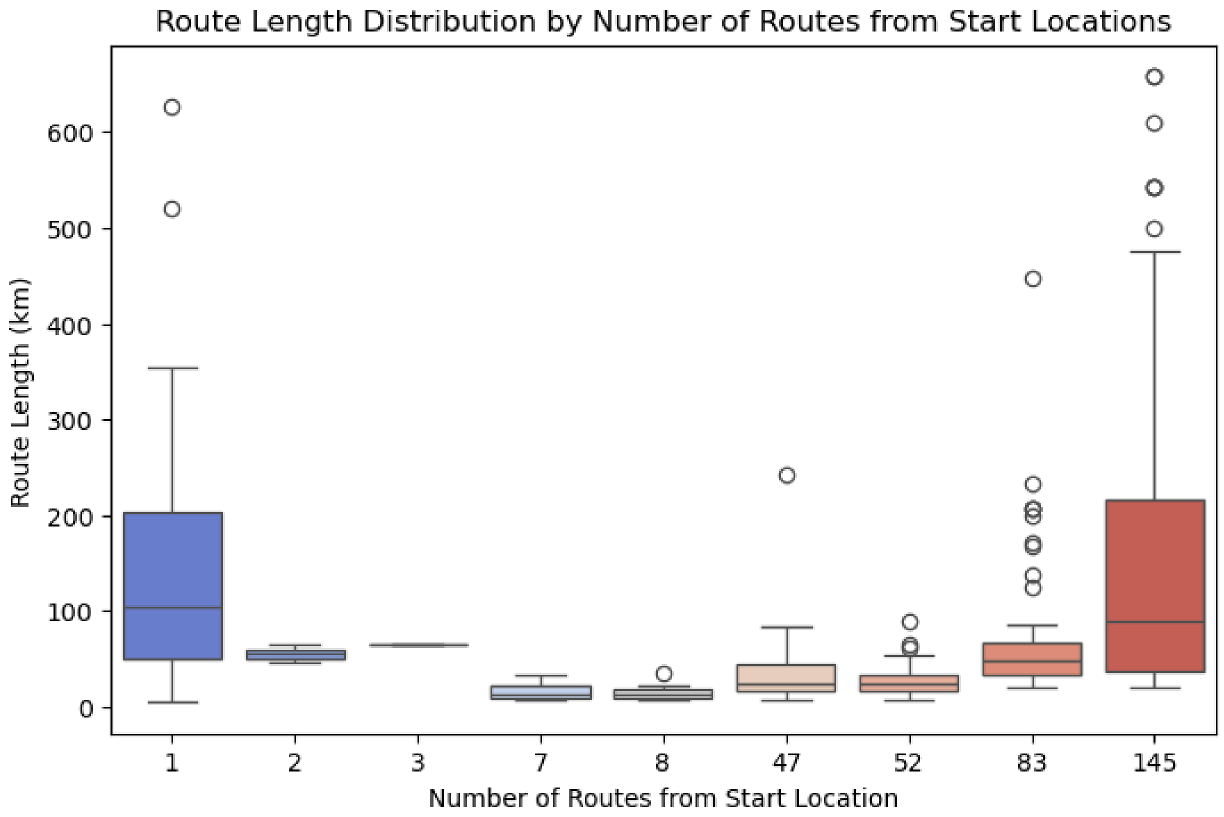
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\2769524211.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=df_grouped["From"], y=df_grouped["ROUTE Length"], palette="rocket")
```



In [61]:

```
df['Start Count'] = df['From'].map(df['From'].value_counts())

plt.figure(figsize=(8, 5))
sns.boxplot(x=df['Start Count'], y=df['ROUTE Length'], palette="coolwarm")
plt.title("Route Length Distribution by Number of Routes from Start Locations")
plt.xlabel("Number of Routes from Start Location")
plt.ylabel("Route Length (km)")
plt.show()
```

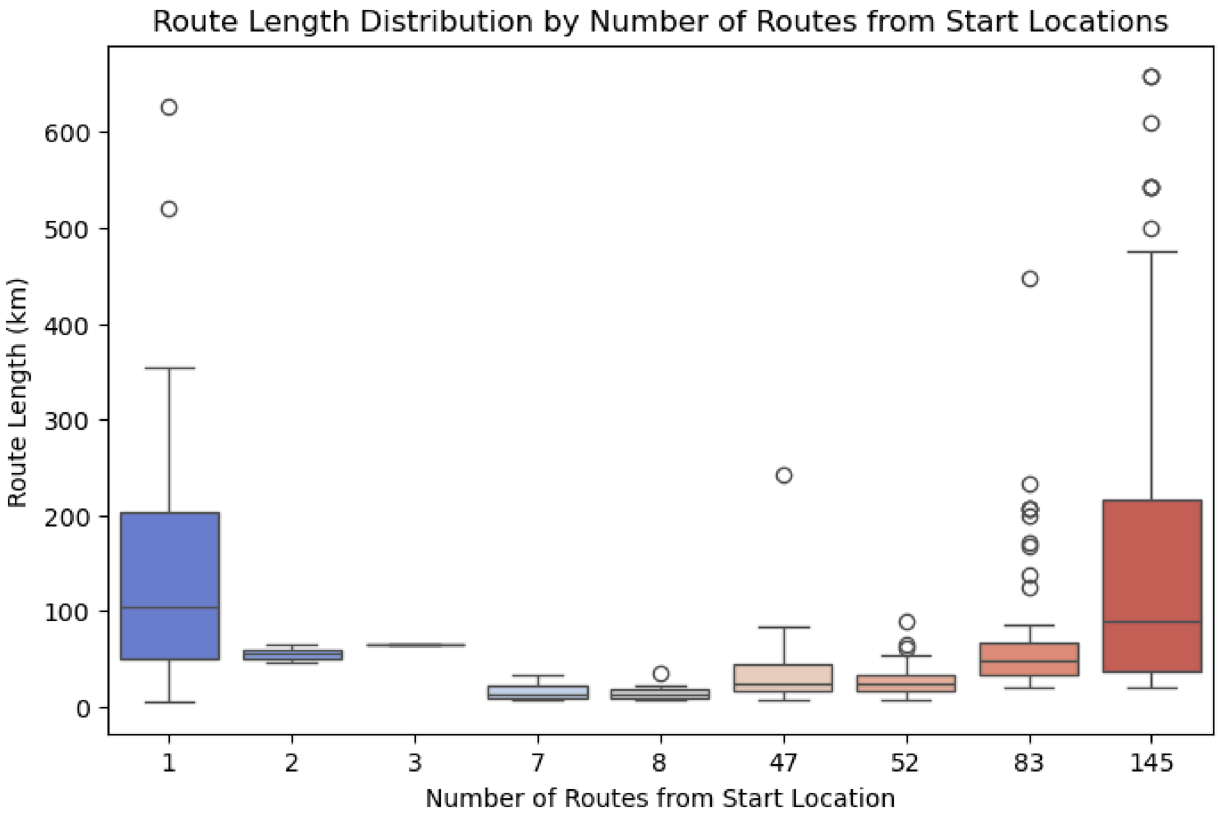
```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\1309940642.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x=df['Start Count'], y=df['ROUTE Length'], palette="coolwarm")
```



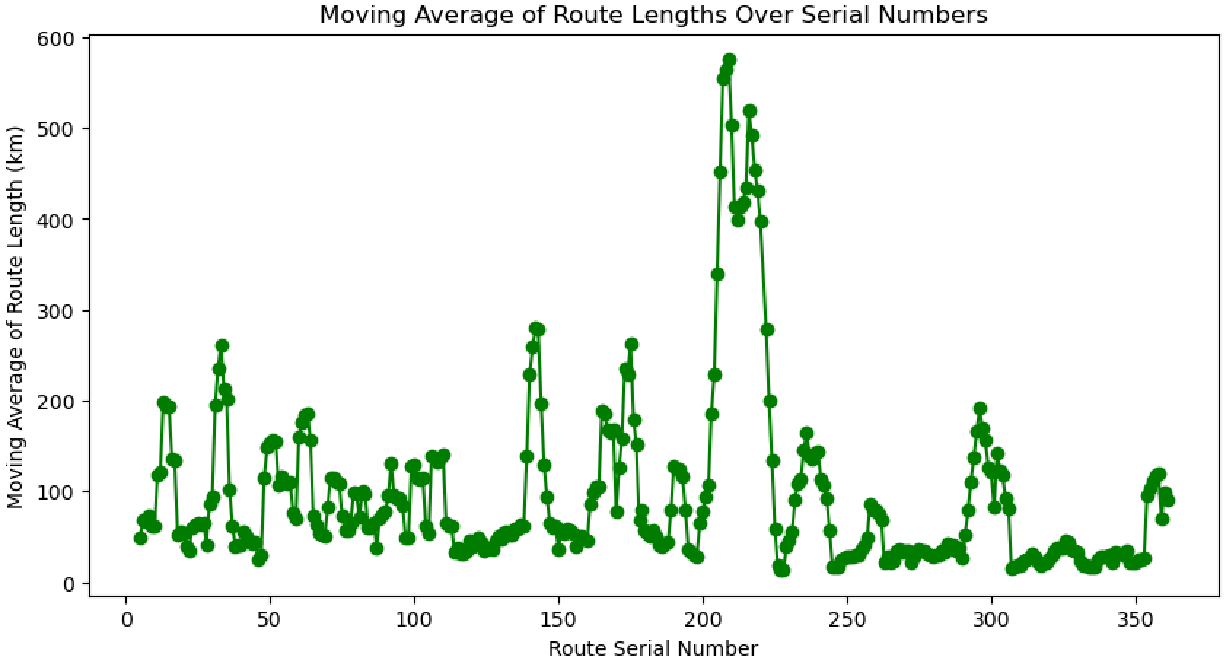
In [62]:

```
df['Start Count'] = df['From'].map(df['From'].value_counts())

plt.figure(figsize=(8, 5))
sns.boxplot(x=df['Start Count'], y=df['ROUTE Length'], palette="coolwarm")
plt.title("Route Length Distribution by Number of Routes from Start Locations")
plt.xlabel("Number of Routes from Start Location")
plt.ylabel("Route Length (km)")
plt.show()
```

```
C:\Users\Saiko\AppData\Local\Temp\ipykernel_15452\1309940642.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x=df['Start Count'], y=df['ROUTE Length'], palette="coolwarm")
```

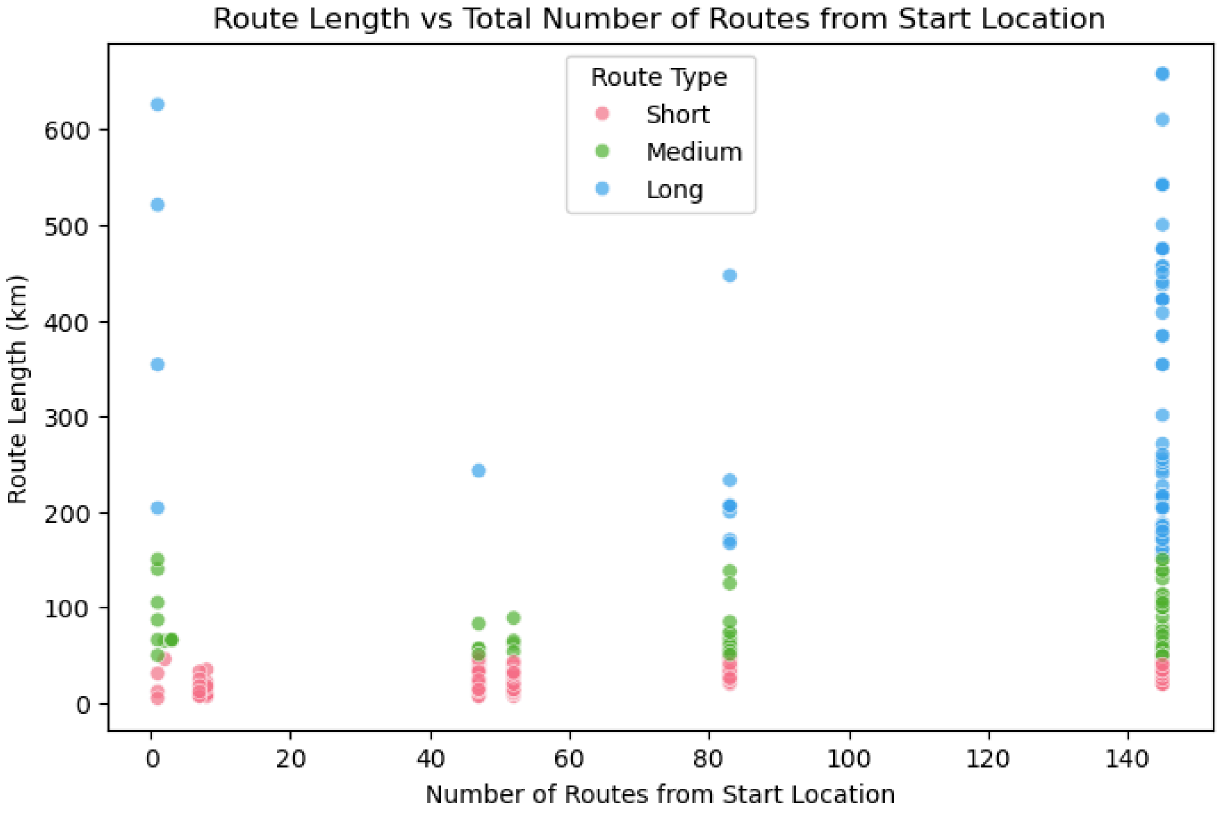


In [63]:
```
df['Moving Average'] = df['ROUTE Length'].rolling(window=5).mean()

plt.figure(figsize=(10, 5))
plt.plot(df['Sl no'], df['Moving Average'], color='green', linestyle='-', marker='o')
plt.title("Moving Average of Route Lengths Over Serial Numbers")
plt.xlabel("Route Serial Number")
plt.ylabel("Moving Average of Route Length (km)")
plt.show()
```

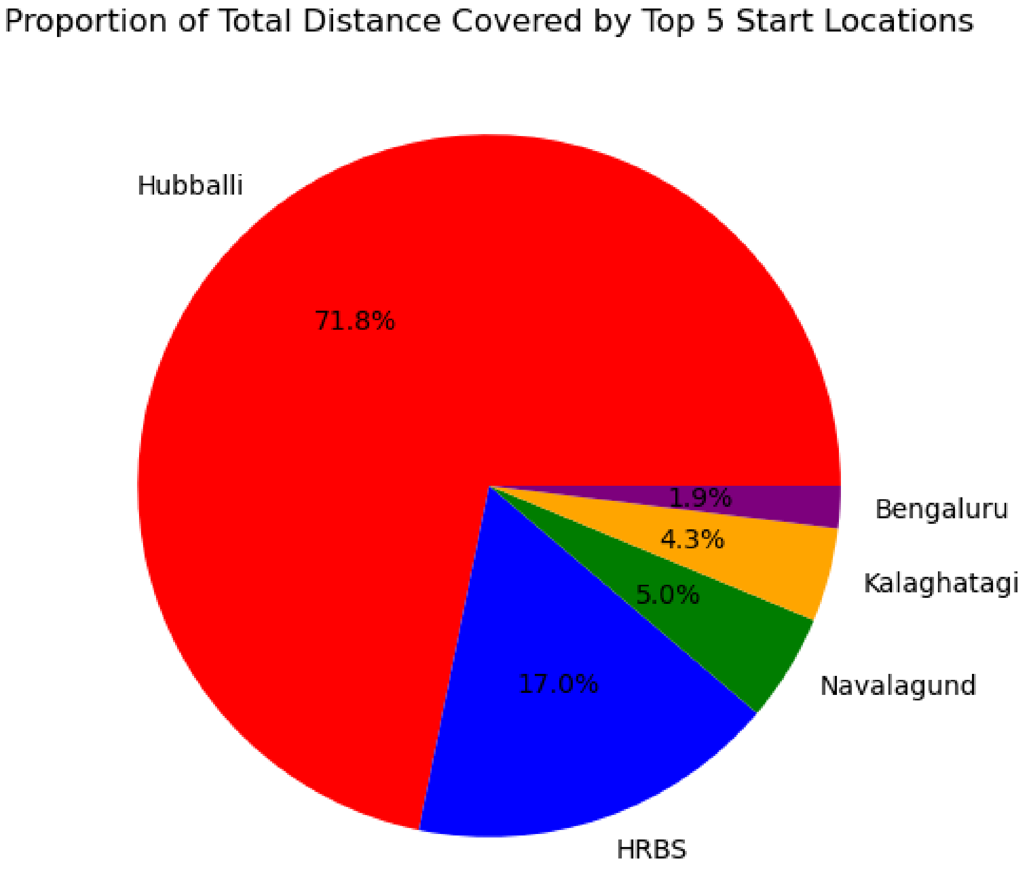


In [64]:

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['Start Count'], y=df['ROUTE Length'], hue=df['Route Type'], palet
plt.title("Route Length vs Total Number of Routes from Start Location")
plt.xlabel("Number of Routes from Start Location")
plt.ylabel("Route Length (km)")
plt.show()
```



In [65]:
```python
top_5_starts = df.groupby("From")["ROUTE Length"].sum().nlargest(5)

plt.figure(figsize=(6, 6))
plt.pie(top_5_starts, labels=top_5_starts.index, autopct='%1.1f%%', colors=['red', 'bl
plt.title("Proportion of Total Distance Covered by Top 5 Start Locations")
plt.show()
```



DATASET OBSERVATION:

When we first looked at the **Hubli Route Division dataset**, it wasn’t perfect—there were some missing values, inconsistencies, and things that needed a little fixing before we could make sense of the data. Here’s how we cleaned it up and what we found along the way.

1. Dealing with Missing Values

We noticed that some route distances (ROUTE Length) were missing. Instead of just deleting those entries, we filled them with the **median route length**. Why median? Because it prevents extreme values (very long or very short routes) from affecting the data too much.

2. Checking Data Types and Converting Units

We made sure each column had the correct data type—for example, making sure numbers were actually stored as numbers instead of text. We also added a **"Miles" column** by converting kilometers into miles (1 km ≈ 0.62 miles). This made the dataset more useful for different types of analysis.

3. Removing Duplicates and Identifying Outliers

We checked if there were any duplicate rows and removed them. Then, we looked for **outliers**—unusually short or long routes—by using a box plot. This helped us spot errors or rare cases that needed attention.

4. Categorizing Routes into Short, Medium, and Long

Instead of treating all routes the same, we grouped them into:

- **Short Routes** (less than 50 km)
- **Medium Routes** (50–150 km)
- **Long Routes** (more than 150 km)
  This helped us see patterns in travel distances.

5. Adding Useful Features

To make the data more insightful, we added:

- **Start Count**: How many times a location appeared as a starting point.
- **Moving Average**: A way to smooth out route distance variations to find trends.

6. What the Visuals Told Us

Bar Plot (Total Route Length per Start Location)

- Hubli had the most starting points and the highest total distance.
- Some places contributed more to travel than others.
  Box Plot (Route Lengths Across Different Locations)
- Some locations had a wide range of distances.
- Outliers showed that some places had extremely long routes.
  Line Plot (Route Length Trends Over Time)
- Route distances fluctuated a lot, but when smoothed out, we could see certain trends.
  Scatter Plot (Start Count vs Route Length)
- More routes from a location didn’t always mean longer distances. Some locations had fewer routes but much longer distances.
  Pie Chart (Top 5 Locations Covering the Most Distance)
- A few locations accounted for most of the total route length, proving their importance in the network.

```
In [ ]:
```