```python
In [1]: import pandas as pd
```

```python
In [2]: import numpy as np
```

```python
In [3]: import matplotlib.pyplot as plt
```

```python
In [4]: import seaborn as sns
```

```python
In [5]: import os
```

```python
In [6]: df=pd.read_csv(r'D:\Python\Electric Vehicle Project\EV_charging_stations_dataset.csv',encoding='latin-1')
```

```python
In [6]: df.head()
```

Out[6]:

	Station ID	Latitude	Longitude	City	Availability	Distance to City (km)	Usage Stats (avg users/day)	Station Operator	Charging Capacity (kW)	Connector Types	Installation Year	Renewable Energy Source	Reviews (Rating)	Parking Spots	Maintenance Frequency
0	EVS00001	-33.400998	77.974972	Köln	9:00-18:00	4.95	35	EVgo	350	CCS, CHAdeMO	2013	Yes	4.0	7	Annually
1	EVS00002	37.861857	-122.490299	Stuttgart	24-Jul	4.96	83	EVgo	350	Tesla, Type 2	2010	Yes	3.9	2	Monthly
2	EVS00003	13.776092	100.412776	Berlin	6:00-22:00	8.54	24	ChargePoint	50	Type 2, CCS	2019	No	3.6	9	Annually
3	EVS00004	43.628250	-79.468935	Stuttgart	9:00-18:00	13.28	70	Greenlots	350	Type 2	2010	Yes	4.2	7	Monthly
4	EVS00005	19.119865	72.913368	Leipzig	9:00-18:00	9.76	19	EVgo	350	CCS	2015	Yes	3.7	6	Annually

```python
In [20]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 15 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Station ID                   5000 non-null   object 
 1   Latitude                     5000 non-null   float64
 2   Longitude                    5000 non-null   float64
 3   City                         5000 non-null   object 
 4   Availability                 5000 non-null   object 
 5   Distance to City (km)        5000 non-null   float64
 6   Usage Stats (avg users/day)  5000 non-null   int64  
 7   Station Operator             5000 non-null   object 
 8   Charging Capacity (kW)       5000 non-null   int64  
 9   Connector Types              5000 non-null   object 
 10  Installation Year            5000 non-null   int64  
 11  Renewable Energy Source      5000 non-null   object 
 12  Reviews (Rating)             5000 non-null   float64
 13  Parking Spots                5000 non-null   int64  
 14  Maintenance Frequency        5000 non-null   object 
dtypes: float64(4), int64(4), object(7)
memory usage: 586.1+ KB
```

```python
In [8]: df.shape
```

Out[8]: (5000, 15)

```python
In [9]: df.isnull().sum()
```

```
Out[9]: Station ID                     0
Latitude                       0
Longitude                      0
City                           0
Availability                   0
Distance to City (km)          0
Usage Stats (avg users/day)    0
Station Operator               0
Charging Capacity (kW)         0
Connector Types                0
Installation Year              0
Renewable Energy Source        0
Reviews (Rating)               0
Parking Spots                  0
Maintenance Frequency          0
dtype: int64
```

```python
In [7]: # Data transformation steps:
# Relace availability data
df['Availability'] = df['Availability'].replace('24-Jul', '24/7')
```

```python
In [8]: # removing extra spaces in Frankfurt city column
df['City'] = df['City'].str.strip()
# This formula cleans your City column by removing leading and trailing whitespace (spaces, tabs, newlines) from each city name.
```

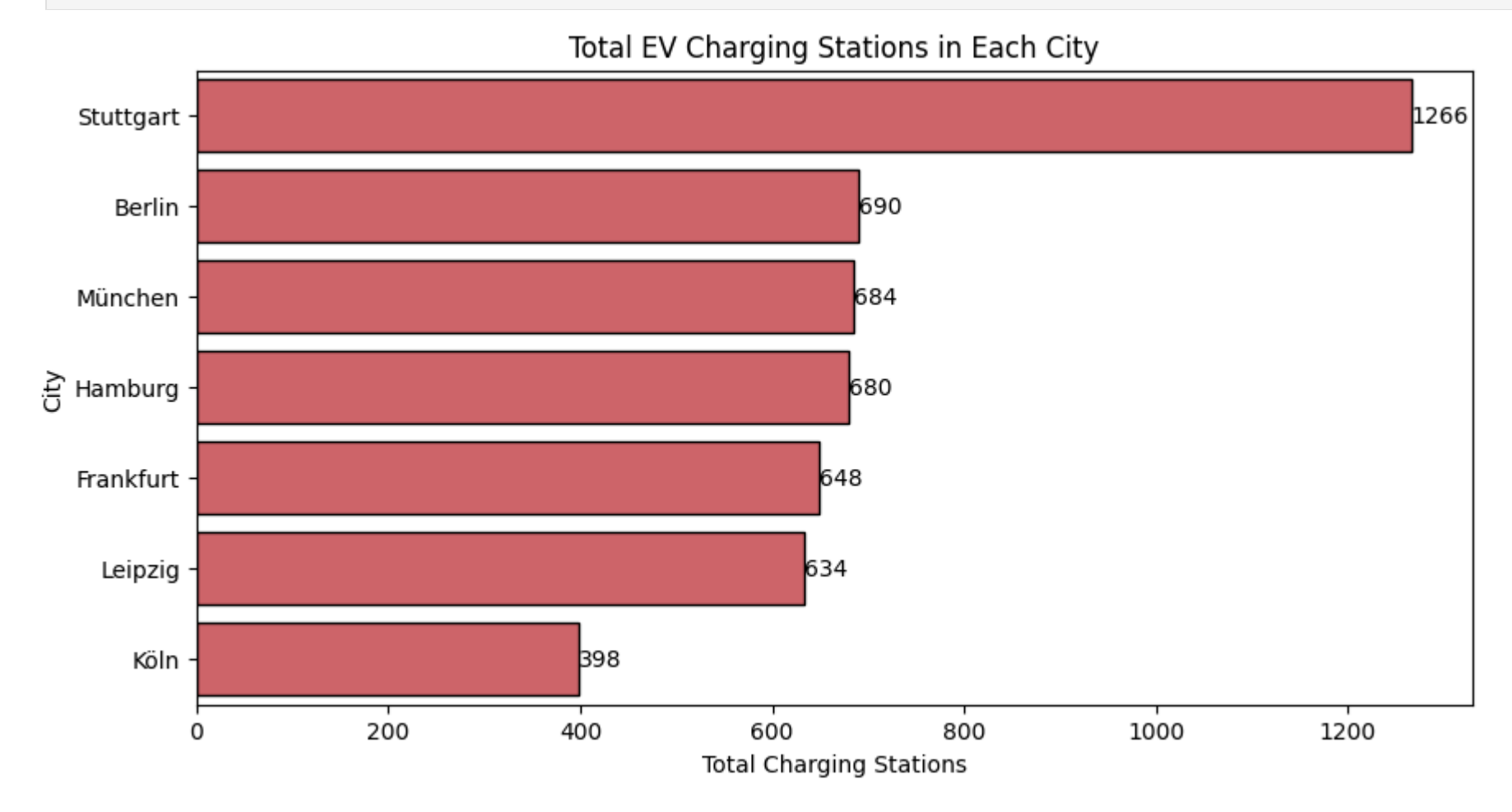
```python
In [9]: df.head(7)
```

Out[9]:

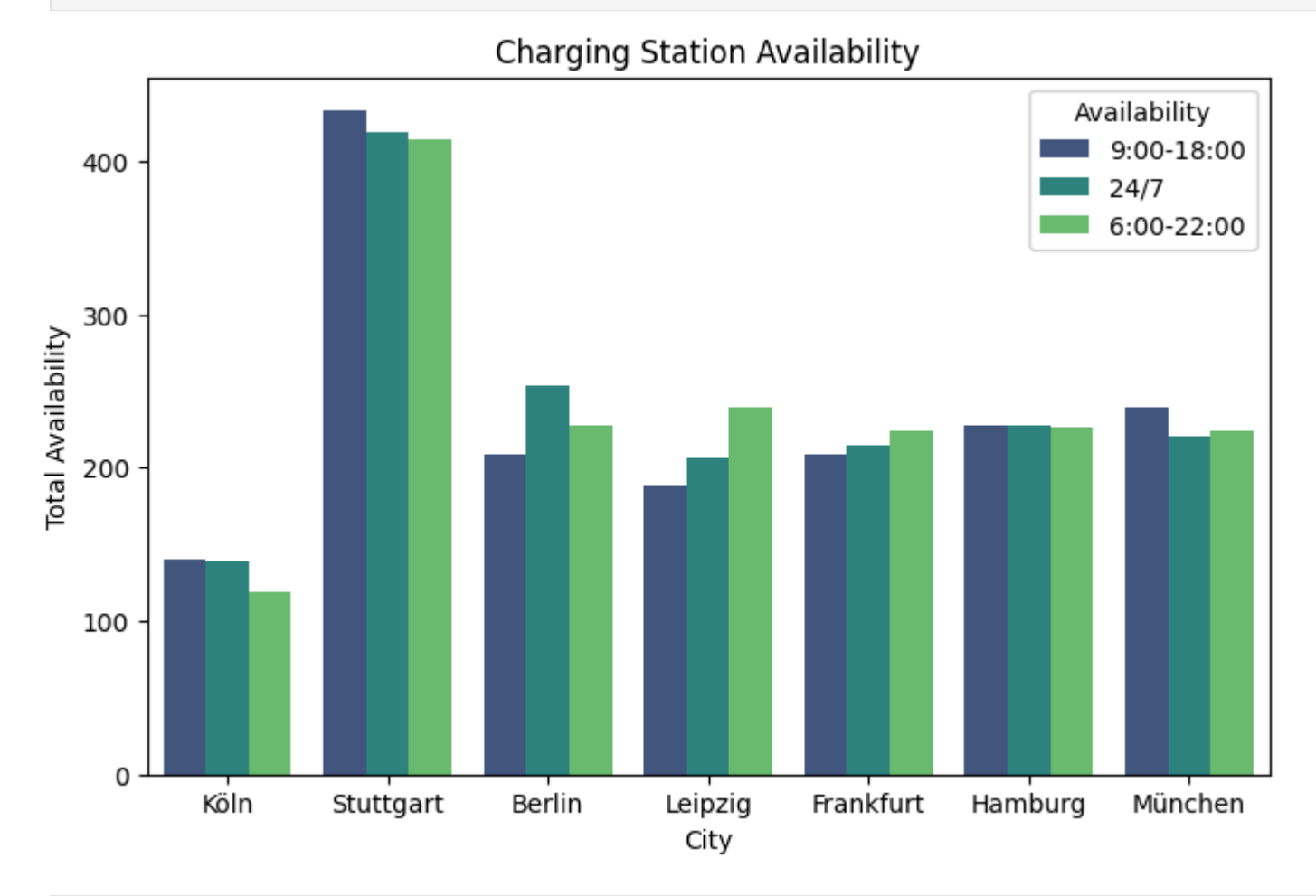
	Station ID	Latitude	Longitude	City	Availability	Distance to City (km)	Usage Stats (avg users/day)	Station Operator	Charging Capacity (kW)	Connector Types	Installation Year	Renewable Energy Source	Reviews (Rating)	Parking Spots	Maintenance Frequency
0	EVS00001	-33.400998	77.974972	Köln	9:00-18:00	4.95	35	EVgo	350	CCS, CHAdeMO	2013	Yes	4.0	7	Annually
1	EVS00002	37.861857	-122.490299	Stuttgart	24/7	4.96	83	EVgo	350	Tesla, Type 2	2010	Yes	3.9	2	Monthly
2	EVS00003	13.776092	100.412776	Berlin	6:00-22:00	8.54	24	ChargePoint	50	Type 2, CCS	2019	No	3.6	9	Annually
3	EVS00004	43.628250	-79.468935	Stuttgart	9:00-18:00	13.28	70	Greenlots	350	Type 2	2010	Yes	4.2	7	Monthly
4	EVS00005	19.119865	72.913368	Leipzig	9:00-18:00	9.76	19	EVgo	350	CCS	2015	Yes	3.7	6	Annually
5	EVS00006	-23.695008	-46.548187	Frankfurt	6:00-22:00	2.02	39	Ionity	350	Tesla	2016	Yes	3.2	9	Quarterly
6	EVS00007	55.762409	37.655830	Hamburg	24/7	2.68	89	ChargePoint	350	Tesla, CCS	2011	Yes	4.0	4	Annually

```python
In [10]: # City counts in ascending order
city_order=df['City'].value_counts().index
```

```python
In [11]: # Number of stations in each city- station ID and group by station ID
plt.figure(figsize=(10,5))
ax=sns.countplot(df['City'],order=city_order,color='#E15759',edgecolor='black')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Total EV Charging Stations in Each City')
plt.xlabel('Total Charging Stations')
plt.savefig(r'D:\Python\Electric Vehicle Project\Total_infrastructure_by_city.png', dpi=300)
plt.show()

# conclusion- Stuttgart, Berlin, Hamburg, and München cities have the highest number of electric vehicle infrastructures than Köln, Leipzig, Frankfurt,
```




```python
In [25]: # Country-wise availability of charging stations
# (Group by x-axis- countries, y-axis- station id, hue- availability)
plt.figure(figsize=(8,5))
sns.countplot(x='City',hue='Availability',data=df,palette='viridis')
plt.title('Charging Station Availability')
plt.xlabel('City')
plt.ylabel('Total Availability')
plt.savefig(r'D:\Python\Electric Vehicle Project\Charging Station Availability in each city.png', dpi=300)
plt.show()
```



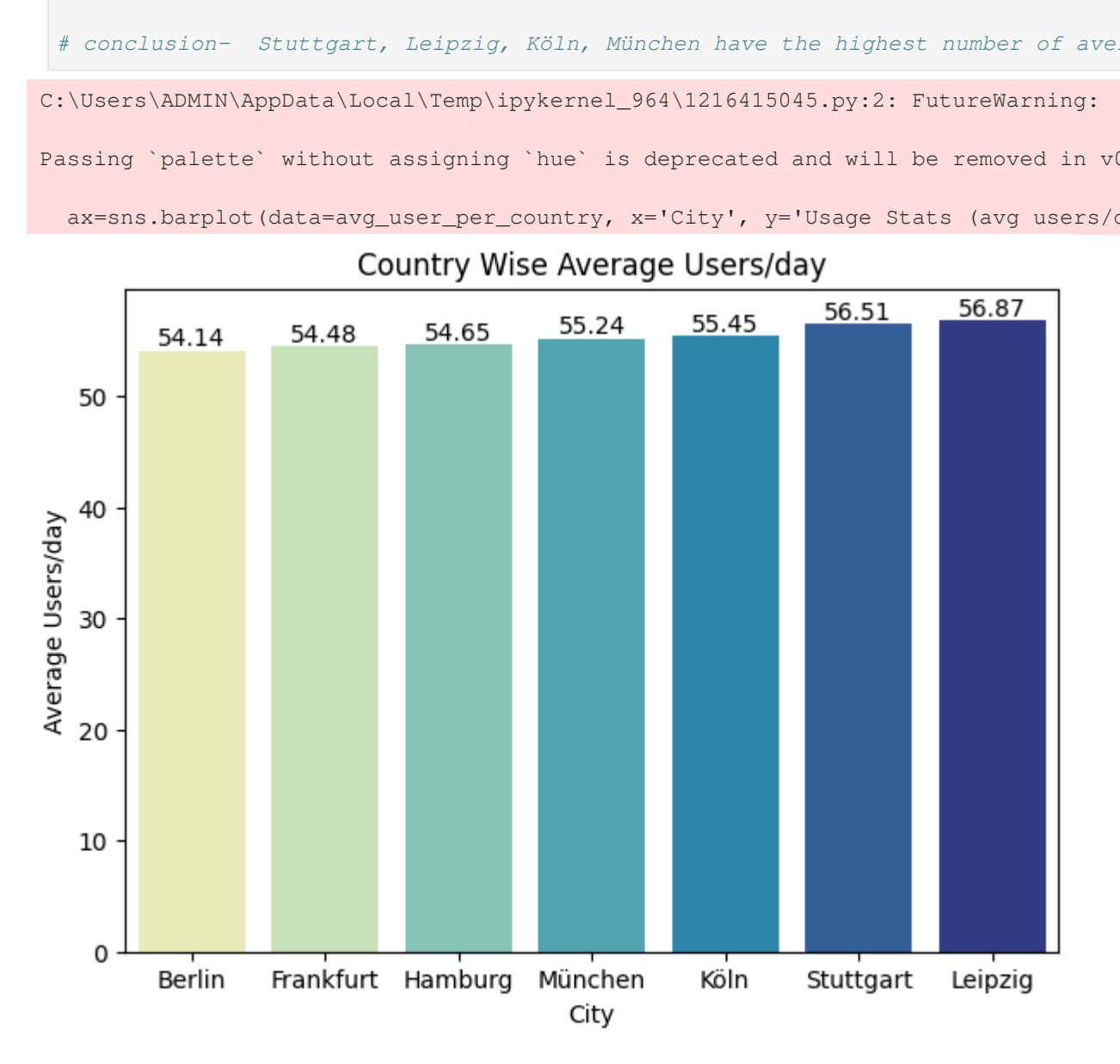

```python
In [27]: # Country-wise avg users/day (group by-country, avg-users/day)
avg_user_per_country=df.groupby('City')['Usage Stats (avg users/day)'].mean().reset_index()
avg_user_per_country=avg_user_per_country.sort_values(by='Usage Stats (avg users/day)',ascending=True)
print(avg_user_per_country)
```

```
        City  Usage Stats (avg users/day)
0     Berlin                    54.143478
1  Frankfurt                    54.476852
2    Hamburg                    54.645588
5    München                    55.235380
3       Köln                    55.452261
6  Stuttgart                    56.511058
4    Leipzig                    56.870662
```

```python
In [26]: # Create the directory if it doesn't exist
output_dir = r'D:\Python\Electric Vehicle Project'
os.makedirs(output_dir, exist_ok=True)
```

```python
In [29]: plt.figure(figsize=(7,5))
ax=sns.barplot(data=avg_user_per_country, x='City', y='Usage Stats (avg users/day)', palette='YlGnBu')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', fontsize=10, color='black',
                xytext=(0, 5), textcoords='offset points')
plt.title('Country Wise Average Users/day')
plt.xlabel('City')
plt.ylabel('Average Users/day')
# Save the figure
output_path = os.path.join(output_dir, 'avg_users_per_day.png')
plt.savefig(output_path, dpi=300, bbox_inches='tight')

plt.show()

# conclusion-  Stuttgart, Leipzig, Köln, München have the highest number of average users per day.
```

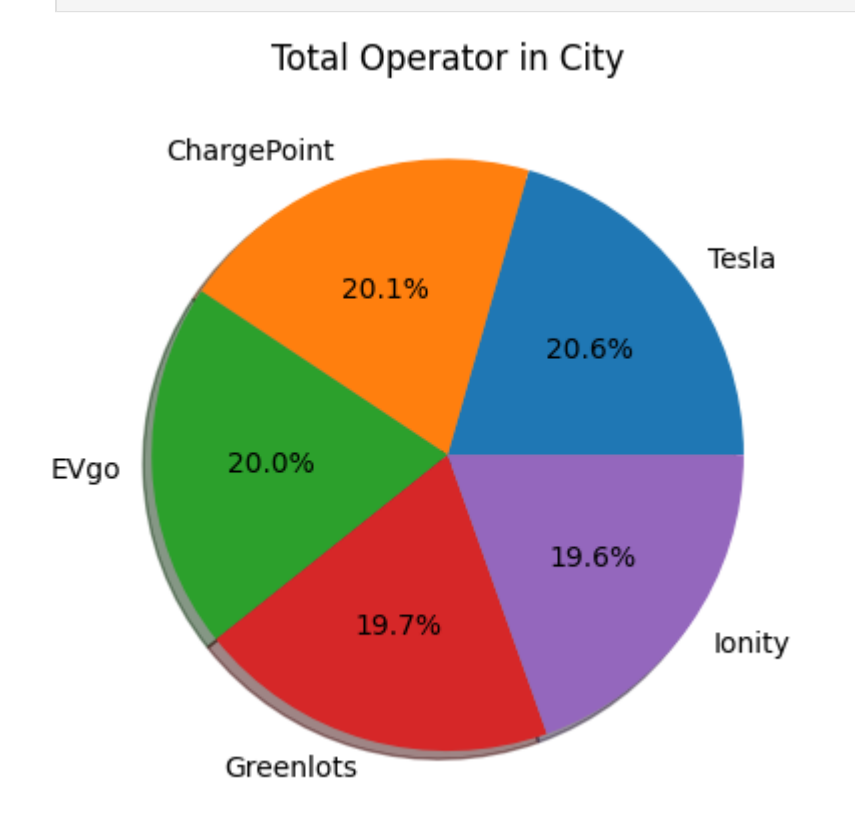
```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_964\1216415045.py:2: FutureWarning: 

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax=sns.barplot(data=avg_user_per_country, x='City', y='Usage Stats (avg users/day)', palette='YlGnBu')
```




```python
In [26]: # Count of station operators in each city
count_operator=df['Station Operator'].value_counts()
print(count_operator)
```

```
Station Operator
Tesla          1029
ChargePoint    1004
EVgo           1002
Greenlots       987
Ionity          978
Name: count, dtype: int64
```

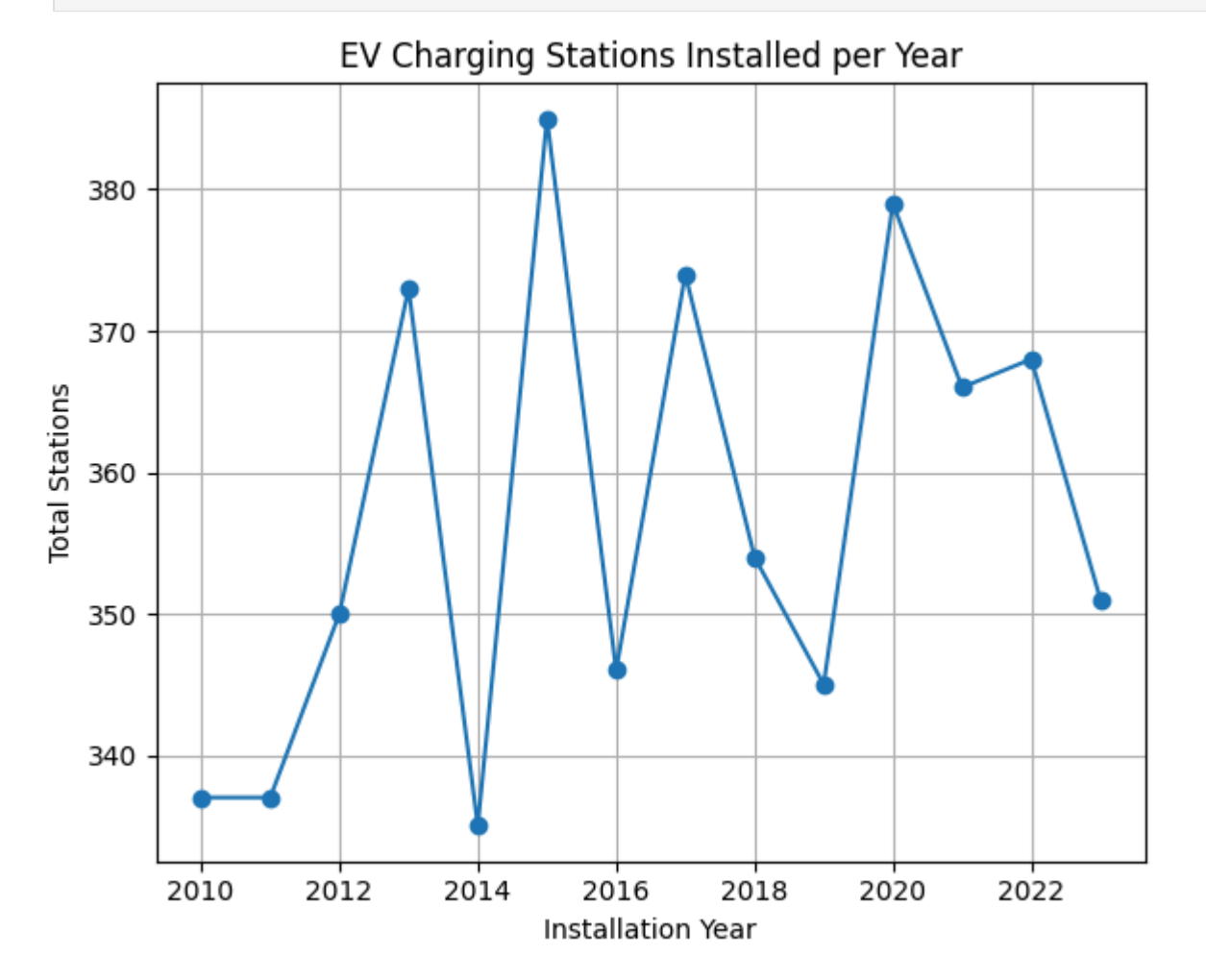
```python
In [27]: # Count of station operators in each city ( group by- city, count-operator)
plt.pie(x=count_operator,labels=count_operator.index,autopct='%1.1f%%',shadow=True)
plt.title('Total Operator in City')
plt.savefig(r'D:\Python\Electric Vehicle Project\Operator_by_city.png', dpi=300)
plt.show()
```




```python
In [12]: # Year-wise installed charging stations (group by - Installed year, count-station id)
total_count=df.groupby('Installation Year')['Station ID'].size()
print(total_count)
```

```
Installation Year
2010    337
2011    337
2012    350
2013    373
2014    335
2015    385
2016    346
2017    374
2018    354
2019    345
2020    379
2021    366
2022    368
2023    351
Name: Station ID, dtype: int64
```

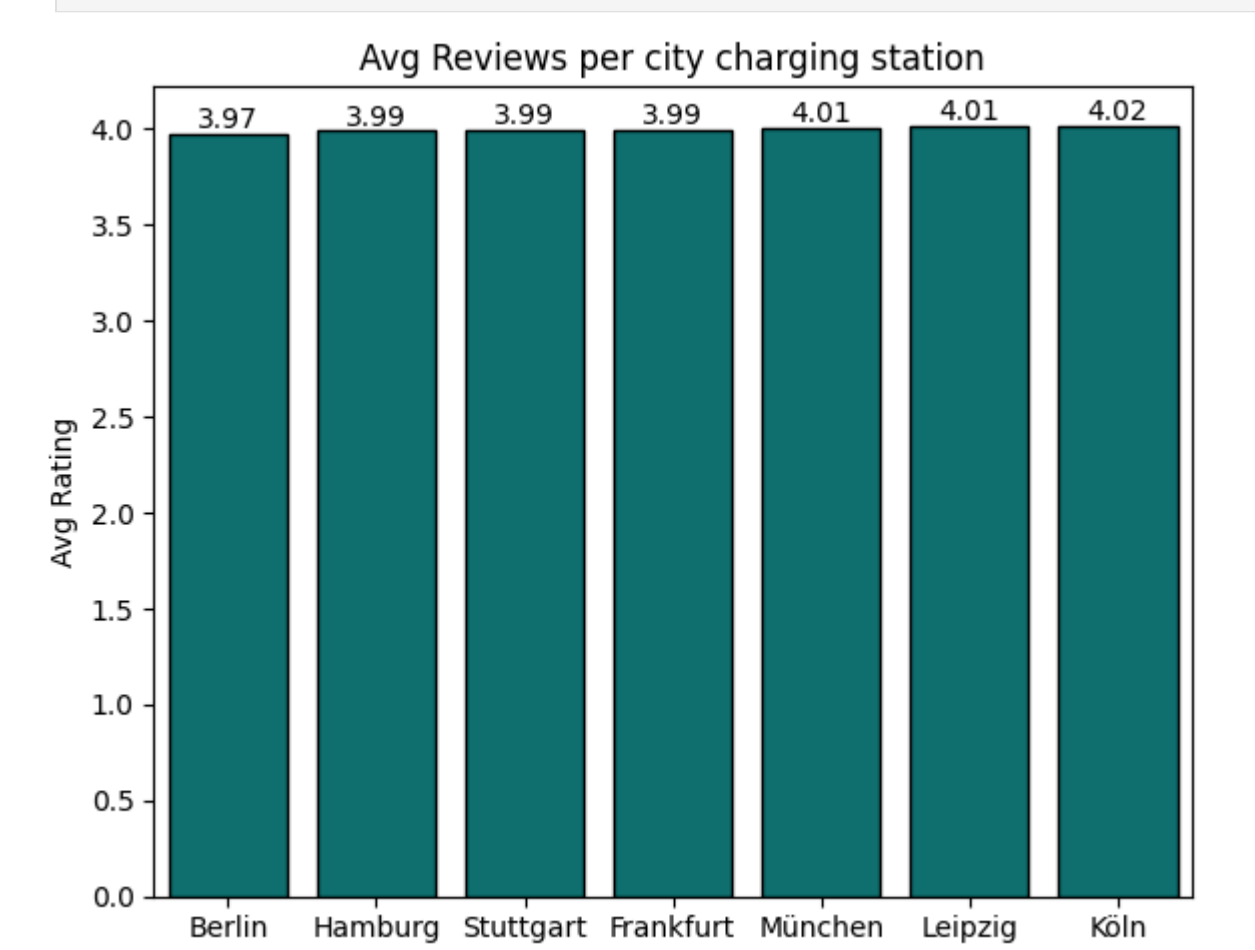
```python
In [13]: plt.figure(figsize=(6,5))
plt.plot(total_count.index,total_count.values,marker='o')
plt.title('EV Charging Stations Installed per Year')
plt.xlabel('Installation Year')
plt.ylabel('Total Stations')
plt.grid()
plt.tight_layout()
plt.savefig(r'D:\Python\Electric Vehicle Project\Total installation by year.png',dpi=300)
plt.show()

# conclusion- Years 2013, 2015, 2017, 2020,and 2022 have the highest number of charging infrastructure in Germany.
```




```python
In [15]: # Country-wise average reviews to charging stations
Avg_reviews=df.groupby('City')['Reviews (Rating)'].mean().reset_index()
Avg_reviews=Avg_reviews.sort_values(by='Reviews (Rating)',ascending=True)
print(Avg_reviews)
```

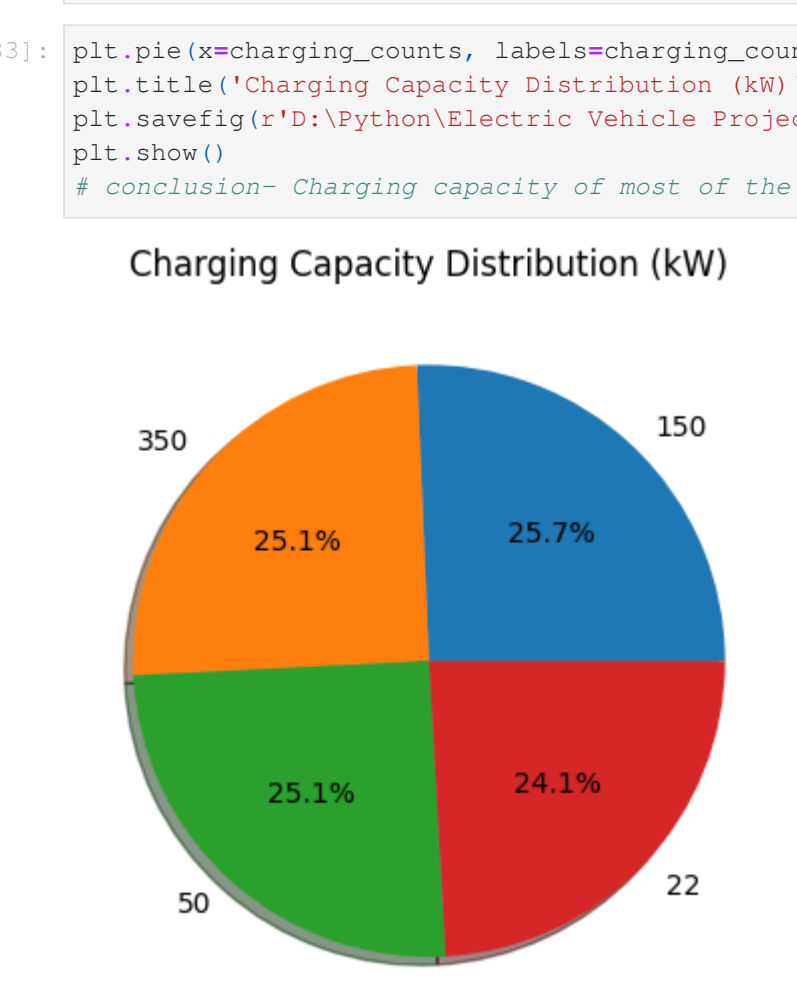
```
        City  Reviews (Rating)
0     Berlin          3.968986
2    Hamburg          3.989559
6  Stuttgart          3.991390
1  Frankfurt          3.991975
5    München          4.005702
4    Leipzig          4.010883
3       Köln          4.019598
```

```python
In [30]: plt.figure(figsize=(6,5))
sns.barplot(x='City',y='Reviews (Rating)',data=Avg_reviews,color='teal',edgecolor='black')
for p in plt.gca().patches:
    plt.gca().annotate(f'{p.get_height():.2f}',
                       (p.get_x() + p.get_width() / 2., p.get_height()),
                       ha='center', va='center',
                       fontsize=10, color='black',
                       xytext=(0, 5), textcoords='offset points')
plt.title('Avg Reviews per city charging station')
plt.xlabel('City')
plt.ylabel('Avg Rating')
plt.tight_layout()

# Save the figure
output_path = os.path.join(output_dir, 'avg_reviews_per_city.png')
plt.savefig(output_path, dpi=300, bbox_inches='tight')
plt.show()

# Conclusion-  Köln,  Leipzig, and München cities have good average reviews per city charging station, which are more than 4.
```



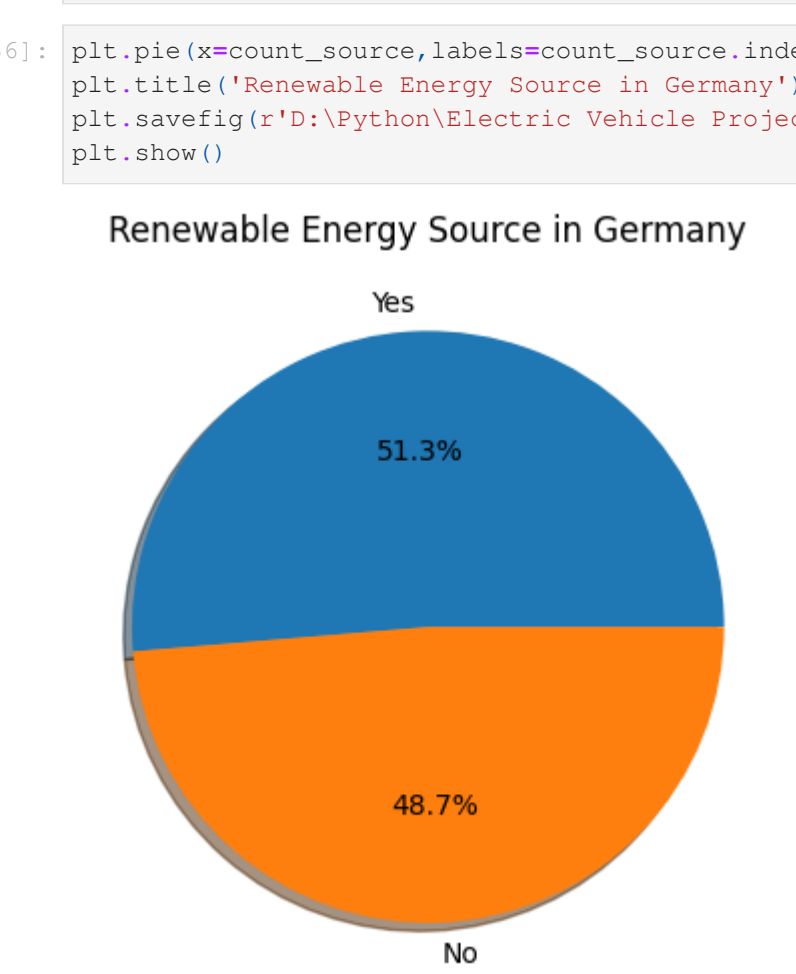

```python
In [32]: # Charging capacity of stations by charger type
# (X-axis- charging capacity, y-axis- station ID, hue- charger type)
charging_counts = df['Charging Capacity (kW)'].value_counts()
```

```python
In [33]: plt.pie(x=charging_counts, labels=charging_counts.index,autopct='%1.1f%%',shadow=True)
plt.title('Charging Capacity Distribution (kW)')
plt.savefig(r'D:\Python\Electric Vehicle Project\charging capacity distribution.png',dpi=300)
plt.show()
# conclusion- Charging capacity of most of the charging stations is 150 kW, 350 kW, and 50 kW.
```

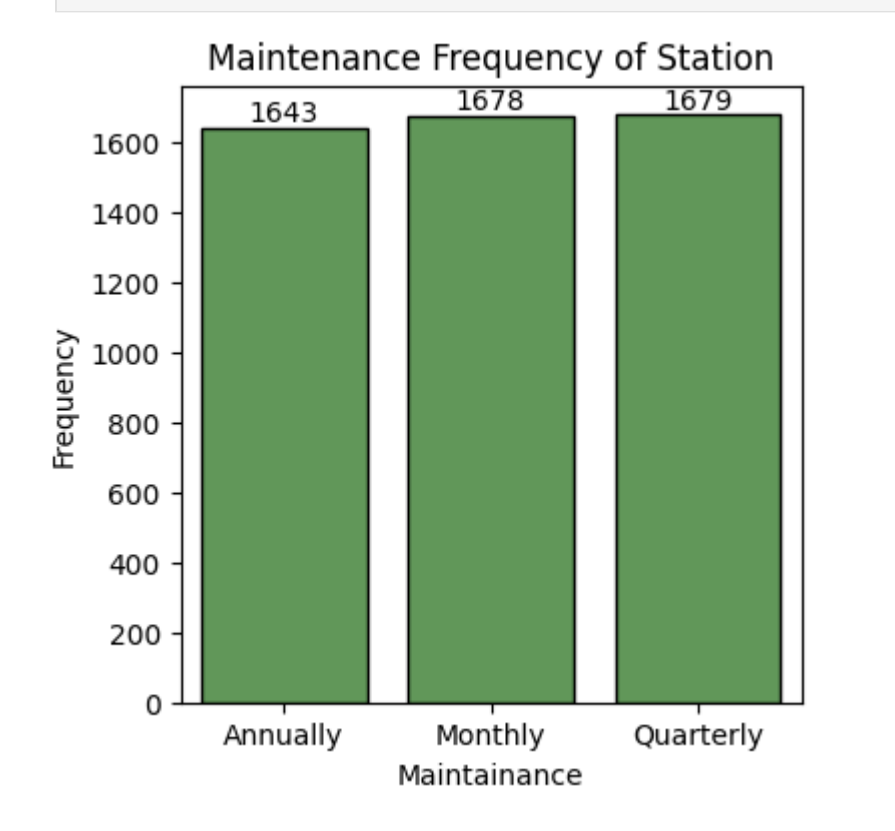



```python
In [35]: # renewable energy source in all of Germany
count_source=df['Renewable Energy Source'].value_counts()
```

```python
In [36]: plt.pie(x=count_source,labels=count_source.index,autopct='%1.1f%%',shadow=True)
plt.title('Renewable Energy Source in Germany')
plt.savefig(r'D:\Python\Electric Vehicle Project\renewable source.png',dpi=300)
plt.show()
```

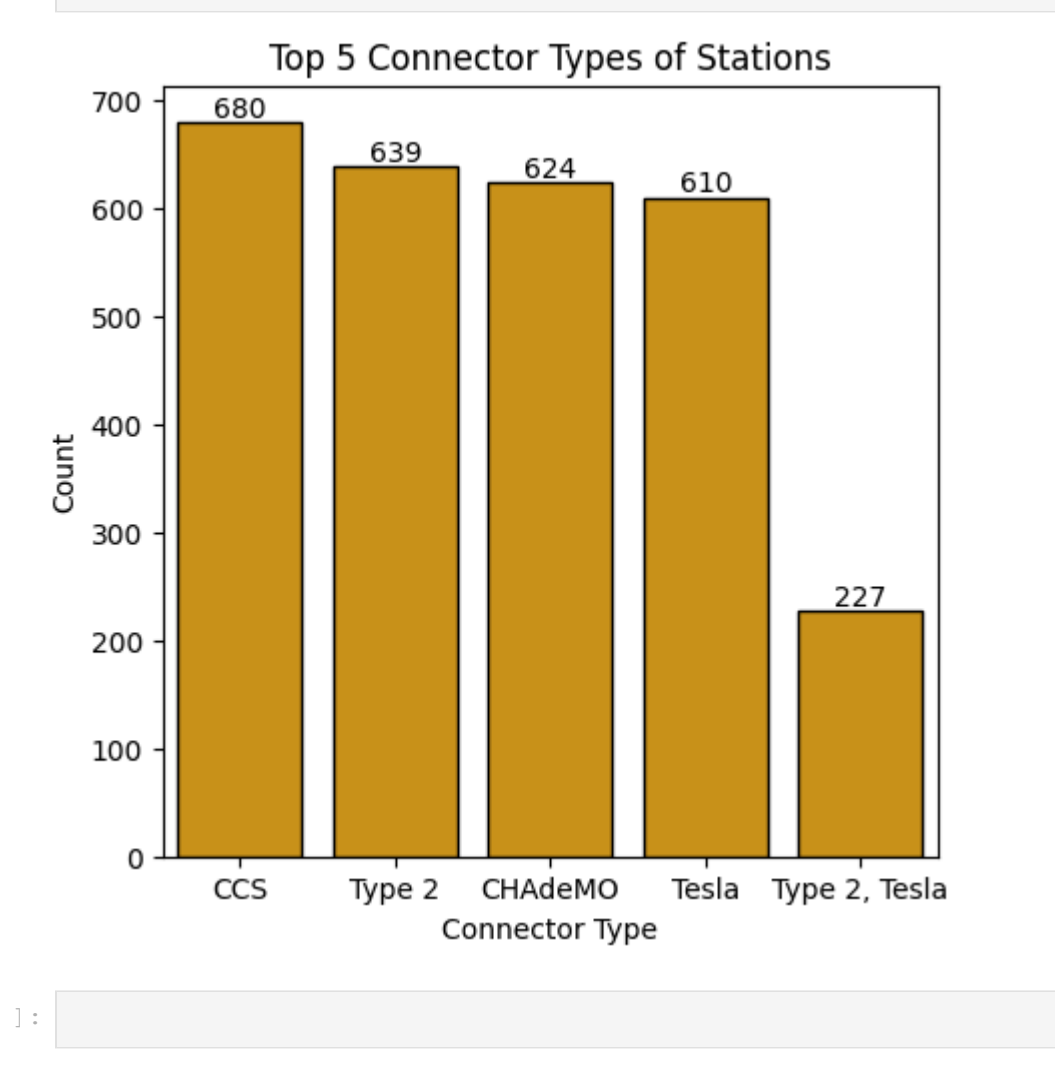



```python
In [37]: # maintenance frequecy each station
plt.figure(figsize=(4,4))
ax=sns.countplot(x='Maintenance Frequency',data=df,color='#59A14F',edgecolor='black')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Maintenance Frequency of Station')
plt.xlabel('Maintainance')
plt.ylabel('Frequency')
plt.savefig(r'D:\Python\Electric Vehicle Project\Maintenance Frequency.png',dpi=300)
plt.show()

# conclusion- Monthly and quarterly maintenance frequency of charging stations is mostly preferred in Germany.
```




```python
In [38]: # Connector types
# Get the top 5 connector types and their counts
top_connectors = df['Connector Types'].value_counts().head(5)

# Convert the Series to a DataFrame
top_connectors_df = top_connectors.reset_index()
top_connectors_df.columns = ['Connector Types', 'Count']
```

```python
In [39]: # Connector types
plt.figure(figsize=(5,5))
ax=sns.barplot(x='Connector Types', y='Count', data=top_connectors_df,color='#E69F00',edgecolor='black')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Top 5 Connector Types of Stations')
plt.xlabel('Connector Type')
plt.ylabel('Count')
plt.savefig(r'D:\Python\Electric Vehicle Project\Top 5 connector used.png',dpi=300)
plt.show()

# These 5 types of connectors are mostly used in Germany charging stations.
```




```python
In [ ]: 
```