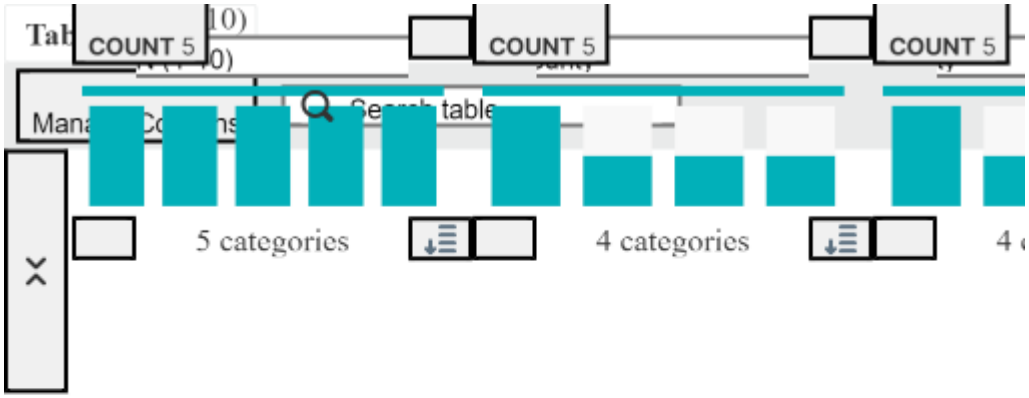
In [ ]:

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

df = pd.read_csv('Electric_Vehicle_Population_Data.csv')

df.head()
```



0

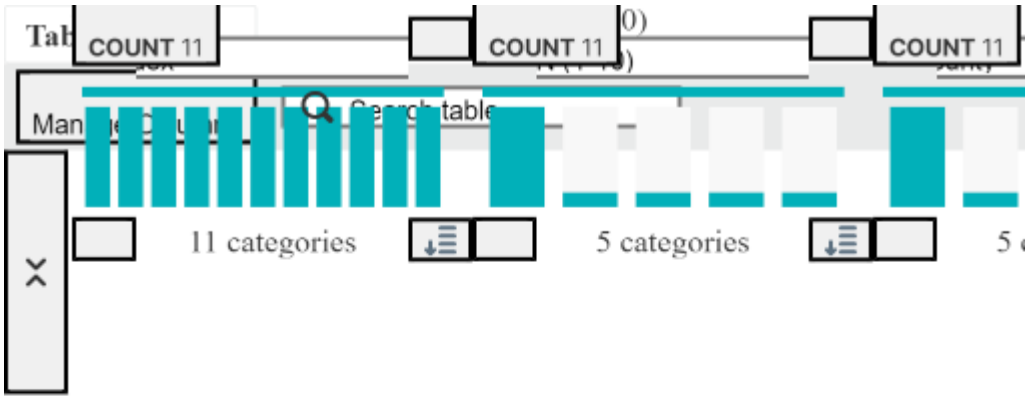
1

2

5YJXCAE26J Yakima Yakima

In [ ]:
```python
df.info()
df.describe(include='all')
```

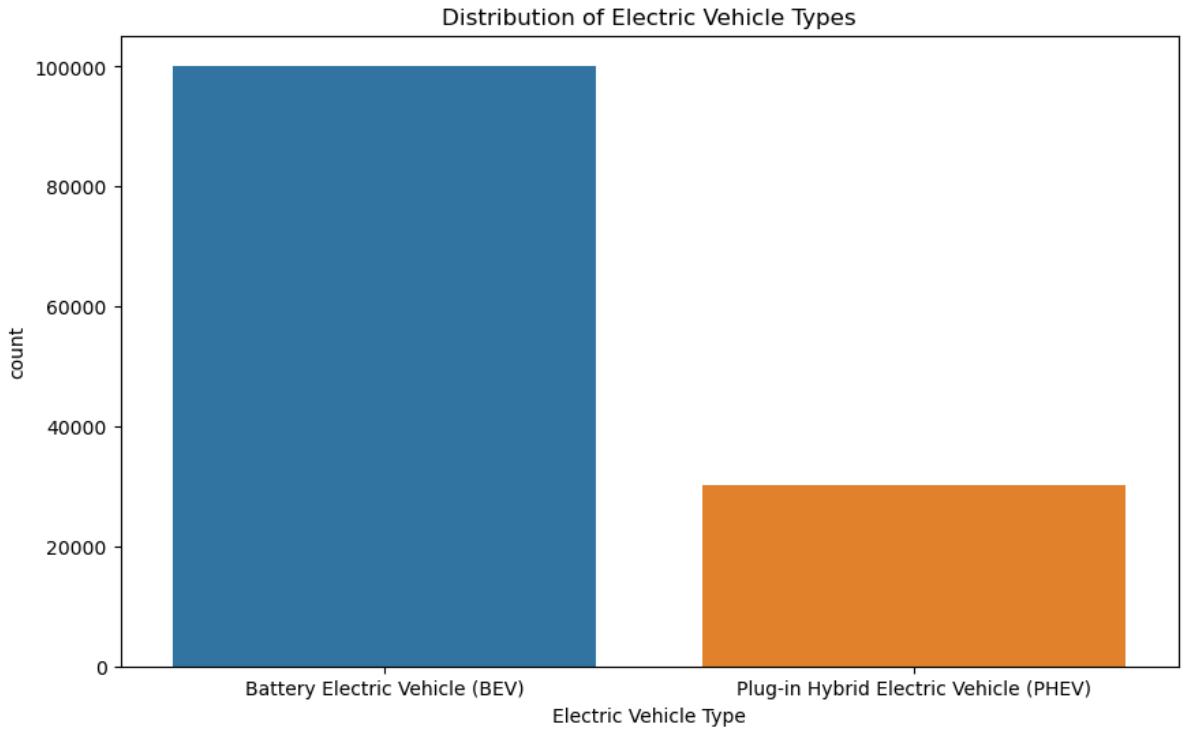
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 130443 entries, 0 to 130442
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dty
pe
---  ------                                             --------------   ---
--
 0   VIN (1-10)                                         130443 non-null  obj
ect
 1   County                                             130440 non-null  obj
ect
 2   City                                               130440 non-null  obj
ect
 3   State                                              130443 non-null  obj
ect
 4   Postal Code                                        130440 non-null  flo
at64
 5   Model Year                                         130443 non-null  int
64
 6   Make                                               130443 non-null  obj
ect
 7   Model                                              130221 non-null  obj
ect
 8   Electric Vehicle Type                              130443 non-null  obj
ect
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  130443 non-null  obj
ect
 10  Electric Range                                     130443 non-null  int
64
 11  Base MSRP                                          130443 non-null  int
64
 12  Legislative District                               130138 non-null  flo
at64
 13  DOL Vehicle ID                                     130443 non-null  int
64
 14  Vehicle Location                                   130410 non-null  obj
ect
 15  Electric Utility                                   130440 non-null  obj
ect
 16  2020 Census Tract                                  130440 non-null  flo
at64
dtypes: float64(3), int64(4), object(10)
memory usage: 16.9+ MB
```

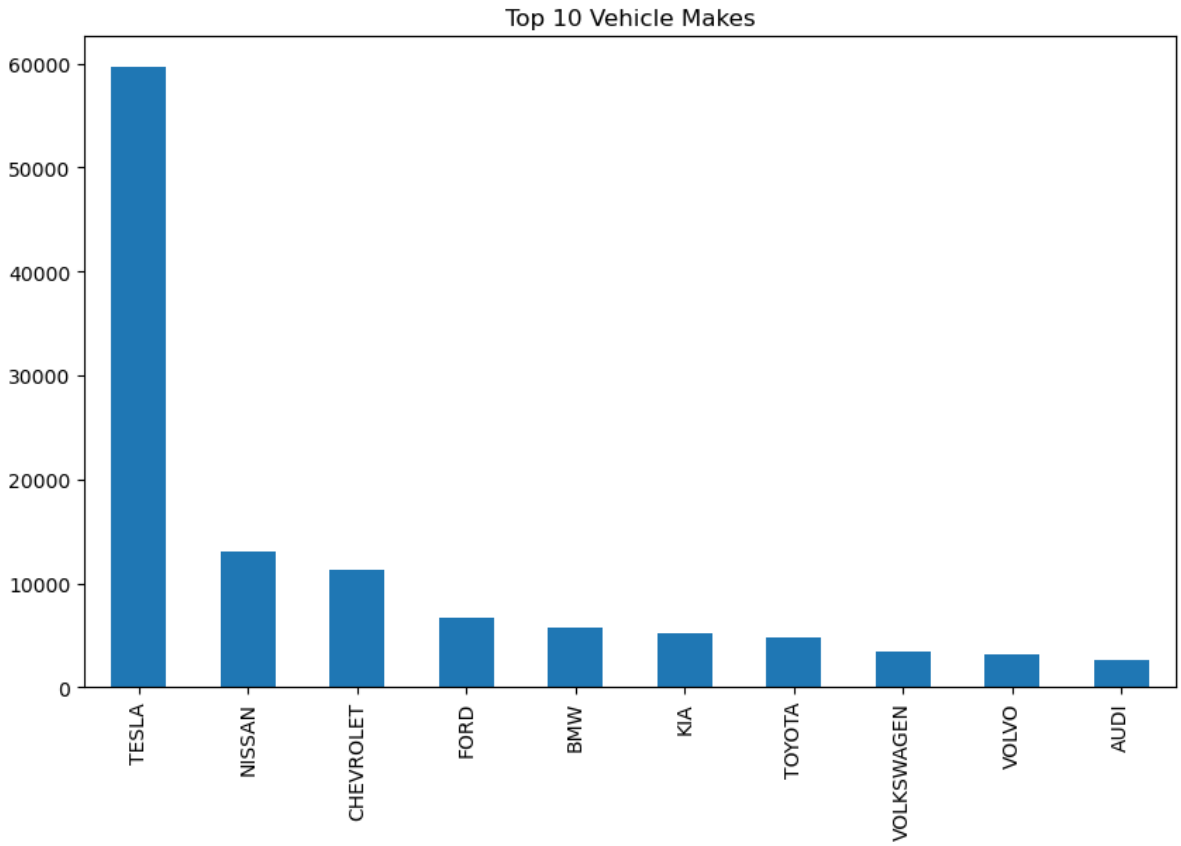


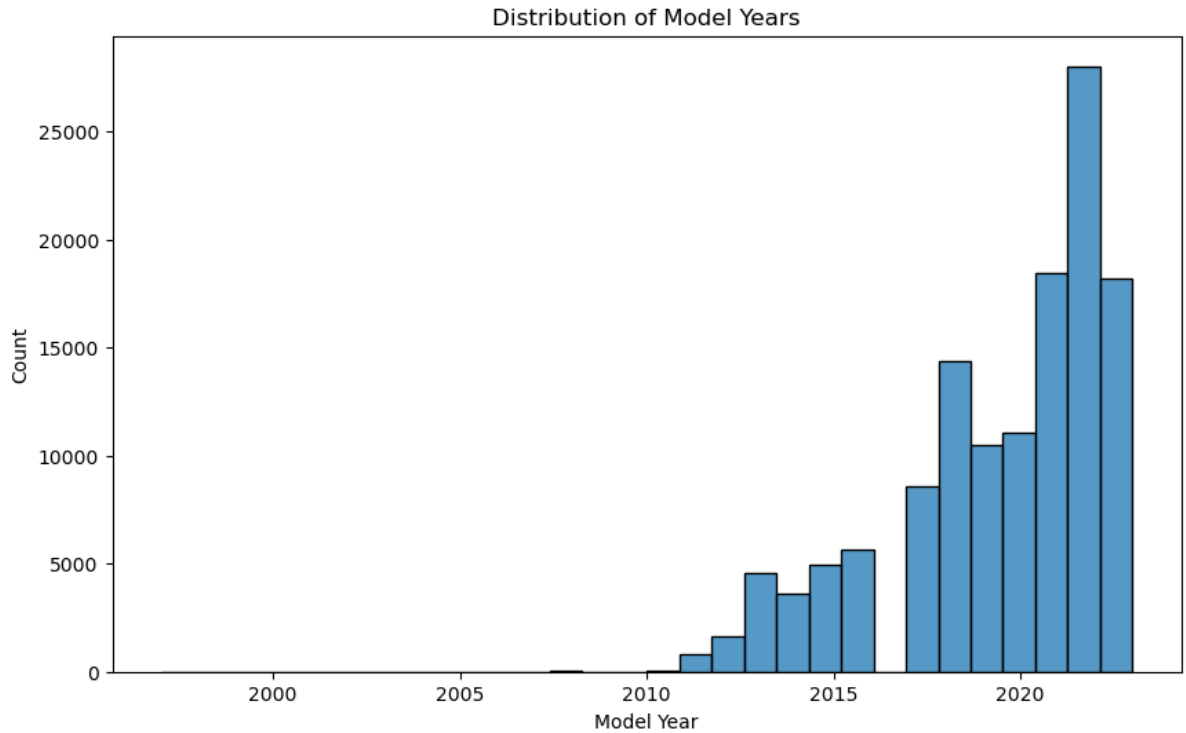
0		
1		
2		
3		
4		
5		
count	130443	130440

In [ ]:

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

# Distribution of Electric Vehicle Types
plt.figure(figsize=(10, 6))
sns.countplot(x='Electric Vehicle Type', data=df)
plt.title('Distribution of Electric Vehicle Types')
plt.show()

# Top 10 Makes
plt.figure(figsize=(10, 6))
df['Make'].value_counts().head(10).plot(kind='bar')
plt.title('Top 10 Vehicle Makes')
plt.show()

# Distribution of Model Years
plt.figure(figsize=(10, 6))
sns.histplot(df['Model Year'], kde=False, bins=30)
plt.title('Distribution of Model Years')
plt.show()
```


Distribution of Electric Vehicle Types
100000
80000
60000
40000
20000
0
count
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)
Electric Vehicle Type


Top 10 Vehicle Makes
60000
50000
40000
30000
20000
10000
0
TESLA
NISSAN
CHEVROLET
FORD
BMW
KIA
TOYOTA
VOLKSWAGEN
VOLVO
AUDI



In [ ]:
```
import geopandas as gpd
from shapely import wkt

# Convert the 'Vehicle Location' column to a GeoSeries
df['Vehicle Location'] = df['Vehicle Location'].apply(wkt.loads)
gdf = gpd.GeoDataFrame(df, geometry='Vehicle Location')

# Plot the data
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
fig, ax = plt.subplots(1, 1)
world[world.name == 'United States'].plot(ax=ax, color='white', edgecolor='b
gdf.plot(ax=ax, color='red')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[4], line 5
      2 from shapely import wkt
      4 # Convert the 'Vehicle Location' column to a GeoSeries
----> 5 df['Vehicle Location'] = df['Vehicle Location'].apply(wkt.loads)
      6 gdf = gpd.GeoDataFrame(df, geometry='Vehicle Location')
      8 # Plot the data

File /opt/conda/lib/python3.9/site-packages/pandas/core/series.py:4771, in S
eries.apply(self, func, convert_dtype, args, **kwargs)
   4661 def apply(
   4662     self,
   4663     func: AggFuncType,
   (...)
   4666     **kwargs,
   4667 ) -> DataFrame | Series:
   4668     """
   4669     Invoke function on values of Series.
   4670
   (...)
   4769     dtype: float64
   4770     """
-> 4771     return SeriesApply(self, func, convert_dtype, args, kwargs).app
ly()

File /opt/conda/lib/python3.9/site-packages/pandas/core/apply.py:1123, in Se
riesApply.apply(self)
   1120     return self.apply_str()
   1122 # self.f is Callable
-> 1123 return self.apply_standard()

File /opt/conda/lib/python3.9/site-packages/pandas/core/apply.py:1174, in Se
riesApply.apply_standard(self)
   1172     else:
   1173         values = obj.astype(object)._values
-> 1174         mapped = lib.map_infer(
   1175             values,
   1176             f,
   1177             convert=self.convert_dtype,
   1178         )
   1180 if len(mapped) and isinstance(mapped[0], ABCSeries):
   1181     # GH#43986 Need to do list(mapped) in order to get treated as ne
sted
   1182     #  See also GH#25959 regarding EA support
   1183     return obj._constructor_expanddim(list(mapped), index=obj.index)

File /opt/conda/lib/python3.9/site-packages/pandas/_libs/lib.pyx:2924, in pa
ndas._libs.lib.map_infer()

File /opt/conda/lib/python3.9/site-packages/shapely/wkt.py:22, in loads(dat
a)
      9 def loads(data):
     10     """
     11     Load a geometry from a WKT string.
     12
```

```
   (...)
     20     Shapely geometry object
     21     """
---> 22     return geos.WKTReader(geos.lgeos).read(data)

File /opt/conda/lib/python3.9/site-packages/shapely/geos.py:328, in WKTReade
r.read(self, text)
    326 """Returns geometry from WKT"""
    327 if not isinstance(text, str):
--> 328     raise TypeError("Only str is accepted.")
    329 text = text.encode()
    330 c_string = c_char_p(text)

TypeError: Only str is accepted.
```

In [ ]:
```
df['Vehicle Location'].head()
```

0

1

2

POINT (-120.56916 46.58514)

In [ ]:
```
# Remove rows with missing 'Vehicle Location' data
df = df.dropna(subset=['Vehicle Location'])

# Try to convert the 'Vehicle Location' column to a GeoSeries again
try:
    df['Vehicle Location'] = df['Vehicle Location'].apply(wkt.loads)
except Exception as e:
    print(f'Error: {e}')

df['Vehicle Location'].head()
```

0

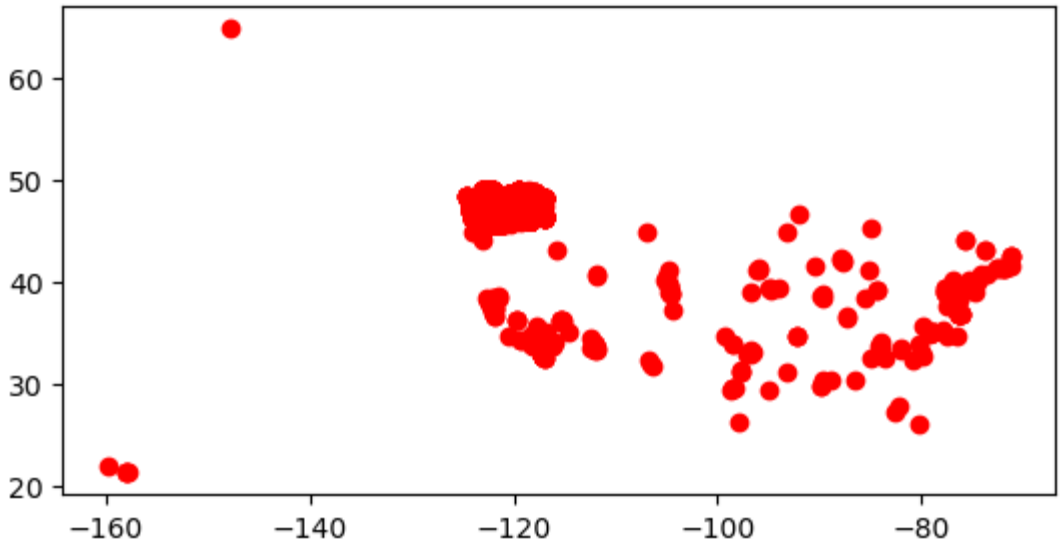
1

2

{"type": "Point",

In [ ]:
```
gdf = gpd.GeoDataFrame(df, geometry='Vehicle Location')

# Plot the data
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
fig, ax = plt.subplots(1, 1)
world[world.name == 'United States'].plot(ax=ax, color='white', edgecolor='b
gdf.plot(ax=ax, color='red')
plt.show()
```

```
ERROR 1: PROJ: proj_create_from_database: Open of /opt/conda/share/proj fail
ed
```



In [ ]:
```
top_models_per_city = df.groupby('City')['Model'].apply(lambda x: x.value_co
top_models_per_city
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
5, in MultiIndex._get_level_number(self, level)
   1554 try:
-> 1555     level = self.names.index(level)
   1556 except ValueError as err:

ValueError: 'None' is not in list

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:100, in ha
ndle_format(obj, with_ipython_display, ipython_shell, extra_metadata)
     99 try:
--> 100     payload, metadata = datalink_processing(
    101         df,
    102         default_index_used,
    103         ipython_shell=ipython,
    104         with_ipython_display=with_ipython_display,
    105         extra_metadata=extra_metadata,
    106     )
    107 except Exception as e:

File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:49, in dat
alink_processing(df, default_index_used, ipython_shell, with_ipython_displa
y, extra_metadata)
     42 def datalink_processing(
     43     df: pd.DataFrame,
     44     default_index_used: bool,
   (...)
     47     extra_metadata: Optional[dict] = None,
     48 ):
---> 49     dxdf = DXDataFrame(df)
     50     parent_display_id = determine_parent_display_id(dxdf)

File /opt/conda/lib/python3.9/site-packages/dx/utils/tracking.py:68, in DXDa
taFrame.__init__(self, df, ipython_shell)
     66 self.index_name = get_df_index(df.index)
---> 68 self.df = normalize_index_and_columns(df)
     69 self.hash = generate_df_hash(self.df)

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:187, in n
ormalize_index_and_columns(df)
    183 """
    184 Any additional formatting that needs to happen to the index,
    185 the columns, or the data itself should be done here.
    186 """
--> 187 df = normalize_index(df)
    188 df = normalize_columns(df)

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:235, in n
ormalize_index(df)
    234         clean_levels.append(clean_level)
--> 235     df.index.set_levels(clean_levels, level=index_name, inplace=Tru
```

```
e)
    237 return df

File /opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:331,
in deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args,
**kwargs)
    326     warnings.warn(
    327         msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
    330     )
--> 331 return func(*args, **kwargs)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:94
3, in MultiIndex.set_levels(self, levels, level, inplace, verify_integrity)
    942 idx._reset_identity()
--> 943 idx._set_levels(
    944     levels, level=level, validate=True, verify_integrity=verify_int
egrity
    945 )
    946 if not inplace:

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in MultiIndex._set_levels(self, levels, level, copy, validate, verify_int
egrity)
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in <listcomp>(.0)
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
8, in MultiIndex._get_level_number(self, level)
   1557 if not is_integer(level):
-> 1558     raise KeyError(f"Level {level} not found") from err
   1559 elif level < 0:

KeyError: 'Level None not found'

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
5, in MultiIndex._get_level_number(self, level)
   1554 try:
-> 1555     level = self.names.index(level)
   1556 except ValueError as err:

ValueError: 'None' is not in list

The above exception was the direct cause of the following exception:
```

```
KeyError                                  Traceback (most recent call last)
Cell In[8], line 2
      1 top_models_per_city = df.groupby('City')['Model'].apply(lambda x:
x.value_counts().head(5))
----> 2 top_models_per_city

File /opt/conda/lib/python3.9/site-packages/IPython/core/displayhook.py:268,
in DisplayHook.__call__(self, result)
    266 self.start_displayhook()
    267 self.write_output_prompt()
--> 268 format_dict, md_dict = self.compute_format_data(result)
    269 self.update_user_ns(result)
    270 self.fill_exec_result(result)

File /opt/conda/lib/python3.9/site-packages/IPython/core/displayhook.py:157,
in DisplayHook.compute_format_data(self, result)
    127 def compute_format_data(self, result):
    128     """Compute format data of the object to be displayed.
    129
    130     The format data is a generalization of the :func:`repr` of an ob
ject.
   (...)
    155
    156     """
--> 157     return self.shell.display_formatter.format(result)

File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:126, in DX
DisplayFormatter.format(self, obj, **kwargs)
    124 def format(self, obj, **kwargs):
    125     if IN_NOTEBOOK_ENV and isinstance(obj, tuple(settings.get_render
able_types())):
--> 126         handle_format(obj)
    127         return ({}, {})
    129     return DEFAULT_IPYTHON_DISPLAY_FORMATTER.format(obj, **kwargs)

File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:110, in ha
ndle_format(obj, with_ipython_display, ipython_shell, extra_metadata)
    108     logger.debug(f"Error in datalink_processing: {e}")
    109     # fall back to default processing
--> 110     df = normalize_index_and_columns(df)
    111     payload, metadata = format_output(
    112         df,
    113         default_index_used=default_index_used,
    114         with_ipython_display=with_ipython_display,
    115         extra_metadata=extra_metadata,
    116     )
    118 return payload, metadata

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:187, in n
ormalize_index_and_columns(df)
    182 def normalize_index_and_columns(df: pd.DataFrame) -> pd.DataFrame:
    183     """
    184     Any additional formatting that needs to happen to the index,
    185     the columns, or the data itself should be done here.
    186     """
--> 187     df = normalize_index(df)
```

```
    188     df = normalize_columns(df)
    189     df = deconflict_index_and_column_names(df)

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:235, in n
ormalize_index(df)
    233         clean_level = clean_series_values(pd.Series(level))
    234         clean_levels.append(clean_level)
--> 235     df.index.set_levels(clean_levels, level=index_name, inplace=Tru
e)
    237 return df

File /opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:331,
in deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args,
**kwargs)
    325 if len(args) > num_allow_args:
    326     warnings.warn(
    327         msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
    330     )
--> 331 return func(*args, **kwargs)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:94
3, in MultiIndex.set_levels(self, levels, level, inplace, verify_integrity)
    941     idx = self._view()
    942 idx._reset_identity()
--> 943 idx._set_levels(
    944     levels, level=level, validate=True, verify_integrity=verify_int
egrity
    945 )
    946 if not inplace:
    947     return idx

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in MultiIndex._set_levels(self, levels, level, copy, validate, verify_int
egrity)
    815     new_levels = FrozenList(
    816         ensure_index(lev, copy=copy)._view() for lev in levels
    817     )
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)
    821     for lev_num, lev in zip(level_numbers, levels):

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in <listcomp>(.0)
    815     new_levels = FrozenList(
    816         ensure_index(lev, copy=copy)._view() for lev in levels
    817     )
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)
    821     for lev_num, lev in zip(level_numbers, levels):

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
8, in MultiIndex._get_level_number(self, level)
```

```
   1556 except ValueError as err:
   1557     if not is_integer(level):
-> 1558         raise KeyError(f"Level {level} not found") from err
   1559     elif level < 0:
   1560         level += self.nlevels

KeyError: 'Level None not found'
```

In [ ]:

```
# Remove rows with missing 'City' data
df = df.dropna(subset=['City'])

# Try to get the top 5 most popular electric vehicle models in each city aga
try:
    top_models_per_city = df.groupby('City')['Model'].apply(lambda x: x.valu
except Exception as e:
    print(f'Error: {e}')

top_models_per_city
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
5, in MultiIndex._get_level_number(self, level)
   1554 try:
-> 1555     level = self.names.index(level)
   1556 except ValueError as err:

ValueError: 'None' is not in list

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:100, in ha
ndle_format(obj, with_ipython_display, ipython_shell, extra_metadata)
     99 try:
--> 100     payload, metadata = datalink_processing(
    101         df,
    102         default_index_used,
    103         ipython_shell=ipython,
    104         with_ipython_display=with_ipython_display,
    105         extra_metadata=extra_metadata,
    106     )
    107 except Exception as e:

File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:49, in dat
alink_processing(df, default_index_used, ipython_shell, with_ipython_displa
y, extra_metadata)
     42 def datalink_processing(
     43     df: pd.DataFrame,
     44     default_index_used: bool,
   (...)
     47     extra_metadata: Optional[dict] = None,
     48 ):
---> 49     dxdf = DXDataFrame(df)
     50     parent_display_id = determine_parent_display_id(dxdf)

File /opt/conda/lib/python3.9/site-packages/dx/utils/tracking.py:68, in DXDa
taFrame.__init__(self, df, ipython_shell)
     66 self.index_name = get_df_index(df.index)
---> 68 self.df = normalize_index_and_columns(df)
     69 self.hash = generate_df_hash(self.df)

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:187, in n
ormalize_index_and_columns(df)
    183 """
    184 Any additional formatting that needs to happen to the index,
    185 the columns, or the data itself should be done here.
    186 """
--> 187 df = normalize_index(df)
    188 df = normalize_columns(df)

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:235, in n
ormalize_index(df)
    234         clean_levels.append(clean_level)
--> 235     df.index.set_levels(clean_levels, level=index_name, inplace=Tru
```

```
e)
    237 return df

File /opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:331,
in deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args,
**kwargs)
    326     warnings.warn(
    327         msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
    330     )
--> 331 return func(*args, **kwargs)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:94
3, in MultiIndex.set_levels(self, levels, level, inplace, verify_integrity)
    942 idx._reset_identity()
--> 943 idx._set_levels(
    944     levels, level=level, validate=True, verify_integrity=verify_int
egrity
    945 )
    946 if not inplace:

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in MultiIndex._set_levels(self, levels, level, copy, validate, verify_int
egrity)
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in <listcomp>(.0)
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
8, in MultiIndex._get_level_number(self, level)
   1557 if not is_integer(level):
-> 1558     raise KeyError(f"Level {level} not found") from err
   1559 elif level < 0:

KeyError: 'Level None not found'

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
5, in MultiIndex._get_level_number(self, level)
   1554 try:
-> 1555     level = self.names.index(level)
   1556 except ValueError as err:

ValueError: 'None' is not in list

The above exception was the direct cause of the following exception:
```

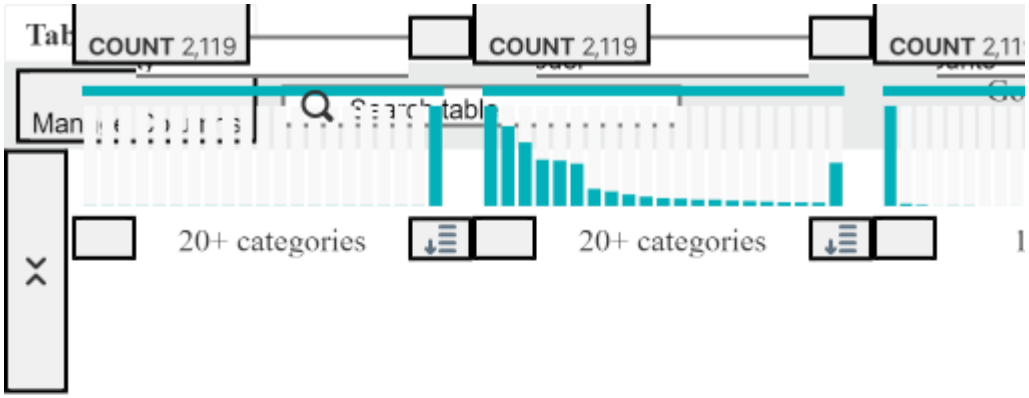
```
KeyError                                  Traceback (most recent call last)
Cell In[9], line 10
      7 except Exception as e:
      8     print(f'Error: {e}')
---> 10 top_models_per_city

File /opt/conda/lib/python3.9/site-packages/IPython/core/displayhook.py:268,
in DisplayHook.__call__(self, result)
    266 self.start_displayhook()
    267 self.write_output_prompt()
--> 268 format_dict, md_dict = self.compute_format_data(result)
    269 self.update_user_ns(result)
    270 self.fill_exec_result(result)

File /opt/conda/lib/python3.9/site-packages/IPython/core/displayhook.py:157,
in DisplayHook.compute_format_data(self, result)
    127 def compute_format_data(self, result):
    128     """Compute format data of the object to be displayed.
    129
    130     The format data is a generalization of the :func:`repr` of an ob
ject.
   (...)
    155
    156     """
--> 157     return self.shell.display_formatter.format(result)

File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:126, in DX
DisplayFormatter.format(self, obj, **kwargs)
    124 def format(self, obj, **kwargs):
    125     if IN_NOTEBOOK_ENV and isinstance(obj, tuple(settings.get_render
able_types())):
--> 126         handle_format(obj)
    127         return ({}, {})
    129     return DEFAULT_IPYTHON_DISPLAY_FORMATTER.format(obj, **kwargs)

File /opt/conda/lib/python3.9/site-packages/dx/formatters/main.py:110, in ha
ndle_format(obj, with_ipython_display, ipython_shell, extra_metadata)
    108     logger.debug(f"Error in datalink_processing: {e}")
    109     # fall back to default processing
--> 110     df = normalize_index_and_columns(df)
    111     payload, metadata = format_output(
    112         df,
    113         default_index_used=default_index_used,
    114         with_ipython_display=with_ipython_display,
    115         extra_metadata=extra_metadata,
    116     )
    118 return payload, metadata

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:187, in n
ormalize_index_and_columns(df)
    182 def normalize_index_and_columns(df: pd.DataFrame) -> pd.DataFrame:
    183     """
    184     Any additional formatting that needs to happen to the index,
    185     the columns, or the data itself should be done here.
    186     """
--> 187     df = normalize_index(df)
```

```
    188     df = normalize_columns(df)
    189     df = deconflict_index_and_column_names(df)

File /opt/conda/lib/python3.9/site-packages/dx/utils/formatting.py:235, in n
ormalize_index(df)
    233         clean_level = clean_series_values(pd.Series(level))
    234         clean_levels.append(clean_level)
--> 235     df.index.set_levels(clean_levels, level=index_name, inplace=Tru
e)
    237 return df

File /opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:331,
in deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args,
**kwargs)
    325 if len(args) > num_allow_args:
    326     warnings.warn(
    327         msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
    330     )
--> 331 return func(*args, **kwargs)

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:94
3, in MultiIndex.set_levels(self, levels, level, inplace, verify_integrity)
    941     idx = self._view()
    942 idx._reset_identity()
--> 943 idx._set_levels(
    944     levels, level=level, validate=True, verify_integrity=verify_int
egrity
    945 )
    946 if not inplace:
    947     return idx

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in MultiIndex._set_levels(self, levels, level, copy, validate, verify_int
egrity)
    815     new_levels = FrozenList(
    816         ensure_index(lev, copy=copy)._view() for lev in levels
    817     )
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)
    821     for lev_num, lev in zip(level_numbers, levels):

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:81
9, in <listcomp>(.0)
    815     new_levels = FrozenList(
    816         ensure_index(lev, copy=copy)._view() for lev in levels
    817     )
    818 else:
--> 819     level_numbers = [self._get_level_number(lev) for lev in level]
    820     new_levels_list = list(self._levels)
    821     for lev_num, lev in zip(level_numbers, levels):

File /opt/conda/lib/python3.9/site-packages/pandas/core/indexes/multi.py:155
8, in MultiIndex._get_level_number(self, level)
```

```
   1556 except ValueError as err:
   1557     if not is_integer(level):
-> 1558         raise KeyError(f"Level {level} not found") from err
   1559     elif level < 0:
   1560         level += self.nlevels

KeyError: 'Level None not found'
```

In [ ]:

```
top_models_per_city = df.groupby(['City', 'Model']).size().reset_index(name=
top_models_per_city = top_models_per_city.groupby('City').apply(lambda x: x.
top_models_per_city
```



0

1

2

3

4

5

6