assignment1_eda

September 19, 2024

1 Assignment 1 : EDA

1.1 Task 1: Reproduce Exploratory Data Analysis as provided in the specimen (at least 20 variables). Discover an interesting aspect of the data that is not shown in the specimen. The specimen is using R but you will reproduce the analysis in Python using libraries of your on choice. (50 points)

1.1.1 Importing Libraries:

[202]:

```
import pandas as pd
import numpy as np

from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

import seaborn as sns
import matplotlib.pyplot as plt

import missingno as msno
```

1.1.2 Importing Data:

[22]:

```
df = pd.read_csv("vehicles.csv")
```

```
/var/folders/r4/9ncp61z543v94_5sltg11_bm0000gn/T/ipykernel_7041/1559163291.py:1:
DtypeWarning: Columns (73,74,76) have mixed types. Specify dtype option on
import or set low_memory=False.
  df = pd.read_csv("vehicles.csv")
```

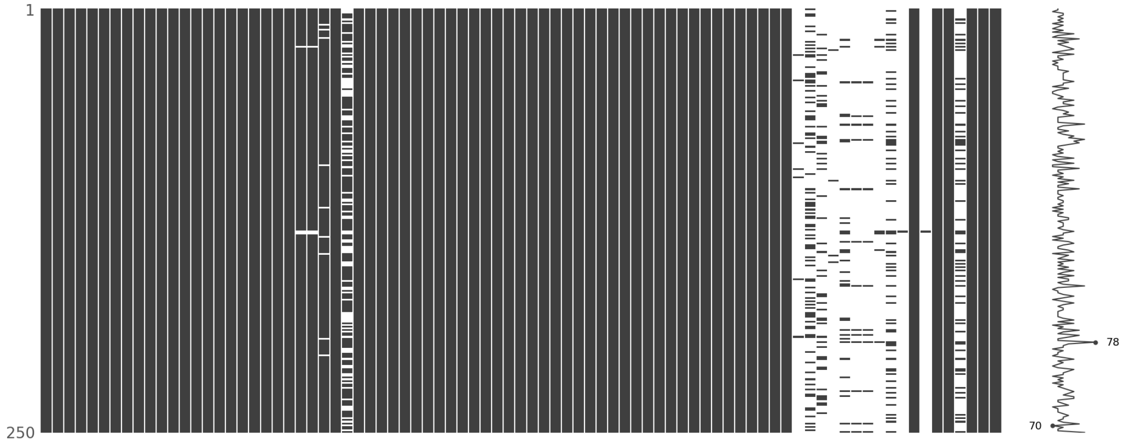
1.1.3 Data Exploration:

[203]:

```
msno.matrix(df.sample(250))
```

[203]:

```
<Axes: >
```



[23]:
```
# Getting the number of rows and columns
df.shape
```

[23]:
```
(40081, 83)
```

[24]:
```
# Exploring the top 5 rows
df.head()
```

[24]:
```
    barrels08  barrelsA08  charge120  charge240  city08  city08U  cityA08  \
0   15.695714         0.0        0.0        0.0      19      0.0        0
1   29.964545         0.0        0.0        0.0       9      0.0        0
2   12.207778         0.0        0.0        0.0      23      0.0        0
3   29.964545         0.0        0.0        0.0      10      0.0        0
4   17.347895         0.0        0.0        0.0      17      0.0        0

    cityA08U  cityCD  cityE  …  mfrCode  c240Dscr  charge240b  c240bDscr  \
0        0.0     0.0    0.0  …      NaN       NaN         0.0        NaN
1        0.0     0.0    0.0  …      NaN       NaN         0.0        NaN
2        0.0     0.0    0.0  …      NaN       NaN         0.0        NaN
3        0.0     0.0    0.0  …      NaN       NaN         0.0        NaN
4        0.0     0.0    0.0  …      NaN       NaN         0.0        NaN

                       createdOn                    modifiedOn  startStop  \
0   Tue Jan 01 00:00:00 EST 2013  Tue Jan 01 00:00:00 EST 2013        NaN
1   Tue Jan 01 00:00:00 EST 2013  Tue Jan 01 00:00:00 EST 2013        NaN
2   Tue Jan 01 00:00:00 EST 2013  Tue Jan 01 00:00:00 EST 2013        NaN
3   Tue Jan 01 00:00:00 EST 2013  Tue Jan 01 00:00:00 EST 2013        NaN
4   Tue Jan 01 00:00:00 EST 2013  Tue Jan 01 00:00:00 EST 2013        NaN

    phevCity  phevHwy  phevComb
0          0        0         0
1          0        0         0
```

```
2          0        0         0
3          0        0         0
4          0        0         0

[5 rows x 83 columns]
```

[25]:

```
# Understand the types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40081 entries, 0 to 40080
Data columns (total 83 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   barrels08        40081 non-null  float64
 1   barrelsA08       40081 non-null  float64
 2   charge120        40081 non-null  float64
 3   charge240        40081 non-null  float64
 4   city08           40081 non-null  int64
 5   city08U          40081 non-null  float64
 6   cityA08          40081 non-null  int64
 7   cityA08U         40081 non-null  float64
 8   cityCD           40081 non-null  float64
 9   cityE            40081 non-null  float64
 10  cityUF           40081 non-null  float64
 11  co2              40081 non-null  int64
 12  co2A             40081 non-null  int64
 13  co2TailpipeAGpm  40081 non-null  float64
 14  co2TailpipeGpm   40081 non-null  float64
 15  comb08           40081 non-null  int64
 16  comb08U          40081 non-null  float64
 17  combA08          40081 non-null  int64
 18  combA08U         40081 non-null  float64
 19  combE            40081 non-null  float64
 20  combinedCD       40081 non-null  float64
 21  combinedUF       40081 non-null  float64
 22  cylinders        39910 non-null  float64
 23  displ            39912 non-null  float64
 24  drive            38892 non-null  object
 25  engId            40081 non-null  int64
 26  eng_dscr         24182 non-null  object
 27  feScore          40081 non-null  int64
 28  fuelCost08       40081 non-null  int64
 29  fuelCostA08      40081 non-null  int64
 30  fuelType         40081 non-null  object
 31  fuelType1        40081 non-null  object
 32  ghgScore         40081 non-null  int64
 33  ghgScoreA        40081 non-null  int64
```

```
34  highway08        40081 non-null  int64
35  highway08U       40081 non-null  float64
36  highwayA08       40081 non-null  int64
37  highwayA08U      40081 non-null  float64
38  highwayCD        40081 non-null  float64
39  highwayE         40081 non-null  float64
40  highwayUF        40081 non-null  float64
41  hlv              40081 non-null  int64
42  hpv              40081 non-null  int64
43  id               40081 non-null  int64
44  lv2              40081 non-null  int64
45  lv4              40081 non-null  int64
46  make             40081 non-null  object
47  model            40081 non-null  object
48  mpgData          40081 non-null  object
49  phevBlended      40081 non-null  bool
50  pv2              40081 non-null  int64
51  pv4              40081 non-null  int64
52  range            40081 non-null  int64
53  rangeCity        40081 non-null  float64
54  rangeCityA       40081 non-null  float64
55  rangeHwy         40081 non-null  float64
56  rangeHwyA        40081 non-null  float64
57  trany            40070 non-null  object
58  UCity            40081 non-null  float64
59  UCityA           40081 non-null  float64
60  UHighway         40081 non-null  float64
61  UHighwayA        40081 non-null  float64
62  VClass           40081 non-null  object
63  year             40081 non-null  int64
64  youSaveSpend     40081 non-null  int64
65  guzzler          2377 non-null   object
66  trans_dscr       15047 non-null  object
67  tCharger         6302 non-null   object
68  sCharger         796 non-null    object
69  atvType          3374 non-null   object
70  fuelType2        1547 non-null   object
71  rangeA           1542 non-null   object
72  evMotor          736 non-null    object
73  mfrCode          9263 non-null   object
74  c240Dscr         65 non-null     object
75  charge240b       40081 non-null  float64
76  c240bDscr        63 non-null     object
77  createdOn        40081 non-null  object
78  modifiedOn       40081 non-null  object
79  startStop        8377 non-null   object
80  phevCity         40081 non-null  int64
81  phevHwy          40081 non-null  int64
```

```
 82  phevComb          40081 non-null  int64
dtypes: bool(1), float64(32), int64(27), object(23)
memory usage: 25.1+ MB
```

[172]:

```python
class DataFrameStats:
    def __init__(self, df: pd.DataFrame):
        """
        initialize new pandas DF
        """
        self.df = df

    def show_date_range(self):
        """
        Get the start and end dates of the dataset
        """

        # filter for bad cols
        if 'year' not in self.df.columns:
            raise ValueError("The DataFrame must contain a 'year' column.")

        # check if the 'year' column has non-null values
        if self.df['year'].dropna().empty:
            raise ValueError("The 'year' column does not contain any valid entries.")

        # get the earliest and latest years
        earliest_year = int(self.df['year'].min())
        latest_year = int(self.df['year'].max())

        # print results
        print(f"The earliest year in the dataset is: {earliest_year}")
        print(f"The latest year in the dataset is: {latest_year}")

        return earliest_year, latest_year

    def get_min(self):
        """
        Return min value
        """
        return self.df.min()

    def get_max(self):
        """
        return max vaule
        """
        return self.df.max()
    def get_q1(self):
        """
        return Q1
        """
        return self.df.quantile(0.25)

    def get_median(self):
        """
        return median
        """
        return self.df.median()

    def get_mean(self):
        """
        Return mean
        """
        return self.df.mean()

    def get_q3(self):
        """
        returns Q3
        """
        return self.df.quantile(0.75)

    def get_numeric_summary(self):
        """
        Get the full summary of numeric data
        """
        numeric_df = self.df.select_dtypes(include=['number'])
        summary = pd.DataFrame({
            'Min': numeric_df.min(),
            'Q1': numeric_df.quantile(0.25),
            'Median': numeric_df.median(),
            'Mean': numeric_df.mean(),
            'Q3': numeric_df.quantile(0.75),
            'Max': numeric_df.max()
        })
        return summary

    def get_non_numeric_summary(self):
        """
        Get the full summary of non-numeric data
        """
        non_numeric_df = self.df.select_dtypes(exclude=['number'])
        non_numeric_summary = {}
        for col in non_numeric_df.columns:
            non_numeric_summary[col] = {
                'Unique Values': non_numeric_df[col].nunique(),
                'Most Frequent': non_numeric_df[col].mode().values[0] if not␣
↪non_numeric_df[col].mode().empty else None,
                'Frequency': non_numeric_df[col].value_counts().to_dict()
            }
        return non_numeric_summary

    def get_summary(self):
        """
        Get the full summary for both
        """
        numeric_summary = self.get_numeric_summary()
        non_numeric_summary = self.get_non_numeric_summary()
        return numeric_summary, non_numeric_summary

    def plot_summary(self):
        """
        Plot heatmap of summary statistics, but only for numerical columns.
        """
        # get the summary
        summary_df = self.get_numeric_summary()

        # create the heatmap
        plt.figure(figsize=(12, 8))
        sns.heatmap(summary_df, annot=True, fmt=".2f", cmap="Blues",␣
↪cbar_kws={'label': 'Value'})

        # add title and labels
        plt.title('Summary Statistics (Min, Q1, Median, Mean, Q3, Max)',␣
↪fontsize=14, fontweight='bold')
        plt.xlabel('Features', fontsize=12)
        plt.ylabel('Statistics', fontsize=12)
        plt.xticks(rotation=90)

        # show plot
        plt.show()


    def plot_countplot(self, col: str):
        """
        Plot a count plot for a non-numerical column
        """

        # filter bad cols
        if col not in self.df.columns:
            raise ValueError(f"Column '{col}' does not exist in the DataFrame.")
        # Filter for non-numeric
        non_numeric_df = self.df.select_dtypes(exclude=['number'])
        if col not in non_numeric_df.columns:
            raise ValueError(f"Column '{col}' is not a non-numerical feature.")

        # create plot
        plt.figure(figsize=(10, 6))
        sns.countplot(data=self.df, x=col, palette='viridis')

        # Add labels and title
        plt.title(f'Count Plot of {col}', fontsize=14, fontweight='bold')
        plt.xlabel(col, fontsize=12)
        plt.ylabel('Count', fontsize=12)
        plt.xticks(rotation=45, ha='right')

        # show plot
        plt.show()

    def plot_boxplot_with_swarm(self, x_col: str, y_col: str):
        """
        Plot a boxplot with swarm plot
        """

        # Logic to handle col name issues
        if x_col not in self.df.columns or y_col not in self.df.columns:
            raise ValueError(f"Either {x_col} or {y_col} does not exist in the␣
↪DataFrame.")

        # Set style
        plt.style.use('ggplot')

        # Create the fig
        plt.figure(figsize=(10, 6))

        # Add boxplot
        sns.boxplot(x=x_col, y=y_col, data=self.df, color='white', fliersize=2,␣
↪boxprops=dict(facecolor='None'))

        # Add swarmplot
        sns.swarmplot(x=x_col, y=y_col, data=self.df, color='black', size=2,␣
↪alpha=0.7)

        # Add labels
        plt.title(f'{y_col} vs {x_col}')
        plt.xlabel(x_col)
        plt.ylabel(y_col)
        plt.xticks(rotation=90)
        # Show plot
        plt.show()

    def plot_scatter_with_boxplot_aggregated(self, x_col: str, y_col: str,␣
↪agg_func='median'):
        """
        Plot a boxplot with swarm plot, but aggregated to reproduce other fig
        """

        # Filter bad cols
        if x_col not in self.df.columns or y_col not in self.df.columns:
            raise ValueError(f"Either {x_col} or {y_col} does not exist in the␣
↪DataFrame.")

        # Group the data by col1 (year) and aggregate the values
        aggregated_df = self.df.groupby(x_col)[y_col].agg(agg_func).
↪reset_index()

        # Create fig
        plt.style.use('ggplot')
        plt.figure(figsize=(10, 6))

        # Add scatter plot
        sns.scatterplot(x=x_col, y=y_col, data=aggregated_df, color='black',␣
↪s=50)

        # Add boxplot
        sns.boxplot(x=x_col, y=y_col, data=self.df, color='white', fliersize=2,␣
↪boxprops=dict(facecolor='None'))

        # Add labels
        plt.title(f'{y_col} vs {x_col} (Aggregated by {agg_func})')
        plt.xlabel(x_col)
        plt.ylabel(y_col)
        plt.xticks(rotation=90)

        # show plot
        plt.show()

    def plot_histogram(self, column: str, bins: int = 30):
        """
        Plot a histogram for a specific numerical column.
        """
        if column not in self.df.columns:
            raise ValueError(f"Column '{column}' does not exist in the␣
↪DataFrame.")
```

```
        numeric_df = self.df.select_dtypes(include=['number'])
        if column not in numeric_df.columns:
            raise ValueError(f"Column '{column}' must be numeric.")

        # Create the histogram
        plt.style.use('ggplot')
        plt.figure(figsize=(10, 6))
        plt.hist(self.df[column], bins=bins, edgecolor='black')

        # Add titles and labels
        plt.title(f'Histogram of {column}', fontsize=14, fontweight='bold')
        plt.xlabel(column, fontsize=12)
        plt.ylabel('Frequency', fontsize=12)

        # Set limits for the x-axis if necessary
        plt.xlim(0, self.df[column].max())

        # Show the plot
        plt.show()


    def plot_fuelType_distribution(self, y_col: str, filter_value: float, fuel_col: str = 'fuelType'):
        """
        Plot distribution of vehicles by fuel type where y_col exceeds a filter_value (e.g., UCity > 75).
        """
        if y_col not in self.df.columns or fuel_col not in self.df.columns:
            raise ValueError(f"Column '{y_col}' or '{fuel_col}' does not exist in the DataFrame.")

        # ensure y_col is numeric
        numeric_df = self.df.select_dtypes(include=['number'])
        if y_col not in numeric_df.columns:
            raise ValueError(f"Column '{y_col}' must be numeric.")

        # filter the DataFrame where y_col exceeds the filter_value
        filtered_df = self.df[self.df[y_col] > filter_value]

        # group by fuel type and count the occurrences
        fuel_counts = filtered_df[fuel_col].value_counts()

        # set the plot style to 'ggplot' for a similar visual style
        plt.style.use('ggplot')

        # create the figure
        plt.figure(figsize=(10, 6))

        # create the bar plot
        fuel_counts.plot(kind='bar', color='gray', edgecolor='black')

        # add titles and labels to match the desired style
        total_vehicles = len(filtered_df)
        plt.title(f'{fuel_col} of {y_col} > {filter_value}, Total␣
↪{total_vehicles}', fontsize=14, fontweight='bold')
        plt.xlabel(fuel_col, fontsize=12)
        plt.ylabel('Vehicles', fontsize=12)

        # Show plot
        plt.show()

    def get_zero_UCity_vehicles(self):
        """
        Get vehicles with ucity equal to 0
        """

        # filter the DataFrame
        zero_UCity_df = self.df[self.df['UCity'] == 0]

        # select the relevant columns
        selected_columns = ['make', 'model', 'fuelType', 'atvType']
        zero_UCity_selected = zero_UCity_df[selected_columns]

        # display the number of vehicles and the filtered DF
        total_zero_UCity = len(zero_UCity_selected)
        print(f"\n{total_zero_UCity} zero UCity vehicles:\n")

        # Show DF
        print(zero_UCity_selected.to_string(index=True))

        return zero_UCity_selected

    def plot_atvType_over_years(self):
        """
        Plot by atvType by the year
        """

        # filter for missing cols
        if 'atvType' not in self.df.columns or 'year' not in self.df.columns:
            raise ValueError("The DataFrame must contain 'atvType' and 'year'␣
↪columns.")
```

```
        # filter where 'atvType' is missing
        df_filtered = self.df.dropna(subset=['atvType', 'year'])

        # group by year and atvType
        df_grouped = df_filtered.groupby(['year', 'atvType']).size().
↪reset_index(name='Total')

        # create plot
        plt.figure(figsize=(12, 8))
        sns.lineplot(data=df_grouped, x='year', y='Total', hue='atvType',␣
↪marker='o', palette='Set2')

        # Add title
        plt.title('Vehicles atvType_Available', fontsize=16, fontweight='bold')
        plt.xlabel('-- Years -->', fontsize=12)
        plt.ylabel('Total', fontsize=12)
        plt.legend(title='atvType', bbox_to_anchor=(1.05, 1), loc='upper left')
        plt.grid(True)

        # Show plot
        plt.tight_layout()
        plt.show()


    def plot_ucity_over_years_by_atvType(self, atv_type: str):
        """
        Plot a boxplot of UCity values for a given atvType
        """
        # Ensure cols exist
        if 'atvType' not in self.df.columns or 'UCity' not in self.df.columns␣
↪or 'year' not in self.df.columns:
            raise ValueError("The DataFrame must contain 'atvType', 'UCity',␣
↪and 'year' columns.")

        # filter the df for the specified atvType
        filtered_df = self.df[self.df['atvType'] == atv_type]

        # check if empty
        if filtered_df.empty:
            print(f"No data found for atvType '{atv_type}'.")
            return

        # create plot
        plt.figure(figsize=(12, 8))
        sns.boxplot(data=filtered_df, x='year', y='UCity', color="white",␣
↪fliersize=2, linewidth=1.5)
```

```
        #add titles and labels
        plt.title(f'{atv_type}', fontsize=16, fontweight='bold')
        plt.xlabel('year', fontsize=12)
        plt.ylabel('UCity', fontsize=12)
        plt.xticks(rotation=45)

        # Show plot
        plt.tight_layout()
        plt.show()

    def plot_correlation_matrix(self):
        """
        Plot the correlation matrix for selected city MPG-related columns to reproduce the figure
        """

        # select relevant columns
        selected_columns = ['city08', 'city08U', 'cityA08', 'cityA08U', 'cityE', 'cityUF', 'phevCity', 'UCity']
        df_selected = self.df[selected_columns]

        # calculate the correlation
        corr_matrix = df_selected.corr()

        # create the mask for the upper triangle
        mask = np.triu(np.ones_like(corr_matrix, dtype=bool))

        # set up figure
        plt.figure(figsize=(10, 8))

        # create a heatmap with bubbles
        sns.heatmap(corr_matrix, annot=False, mask=mask, cmap="coolwarm", center=0,
                    linewidths=1, cbar_kws={"shrink": 0.8}, square=True)

        # overlay scatterplot for bubble sizes
        for i in range(corr_matrix.shape[0]):
            for j in range(i):
                plt.scatter(i + 0.5, j + 0.5, s=np.abs(corr_matrix.iloc[i, j]) * 1000,
                            color='b' if corr_matrix.iloc[i, j] > 0 else 'r')

        # rotate the x-axis
        plt.xticks(rotation=90, fontsize=12, color='red')
        plt.yticks(fontsize=12, color='red')

        # add title
        plt.title('Correlation Matrix with Bubble Sizes', fontsize=16,␣
↪fontweight='bold')

        # show plot
        plt.tight_layout()
        plt.show()

    def plot_feature_distribution(self, feature: str, top_n: int = None):
        """
        Plot a bar chart of the count of unique values in any categorical␣
↪feature column
        """

        # check the feature exists
        if feature not in self.df.columns:
            raise ValueError(f"The DataFrame does not contain a column named␣
↪'{feature}'.")

        # check the feature is categorical
        if self.df[feature].dtype not in ['object', 'category'] and self.
↪df[feature].nunique() > 100:
            raise ValueError(f"The column '{feature}' is not categorical or has␣
↪too many unique values to display.")

        # count occurences
        feature_counts = self.df[feature].value_counts()

        # limit to top N
        if top_n:
            feature_counts = feature_counts.head(top_n)

        # create the bar plot
        plt.figure(figsize=(12, 8))
        sns.barplot(x=feature_counts.index, y=feature_counts.values,␣
↪color='gray', edgecolor='black')

        # add titles and labels
        plt.title(f'Distribution of {feature}', fontsize=16, fontweight='bold')
        plt.xlabel(feature, fontsize=12)
        plt.ylabel('Count', fontsize=12)
        plt.xticks(rotation=90)

        # Show plot
        plt.tight_layout()
        plt.show()
```

```python
    def plot_full_correlation_matrix(self):
        """
        Plot full corr matrix
        """
        # get numeric only
        numeric_df = self.df.select_dtypes(include=['number'])

        # calculate the correlation
        corr_matrix = numeric_df.corr()

        # create a heatmap
        plt.figure(figsize=(12, 8))
        sns.heatmap(corr_matrix, annot=False, fmt='.2f', cmap='coolwarm', center=0, linewidths=0.5,
                    annot_kws={"size": 8})

        # add titles
        plt.title('Correlation Matrix of Numerical Features', fontsize=10, fontweight='bold')
        plt.xticks(rotation=90)
        plt.yticks(rotation=0)

        # show plot
        plt.tight_layout()
        plt.show()


    def plot_and_calculate_r2(self, x_col: str, y_col: str):
        """
        plot two variables against each other and calculate the correlation
        """
        # confirm columns
        if x_col not in self.df.columns or y_col not in self.df.columns:
            raise ValueError(f"Either {x_col} or {y_col} does not exist in the DataFrame.")

        # remove the nans
        df_filtered = self.df[[x_col, y_col]].dropna()

        # extract vars
        X = df_filtered[x_col].values.reshape(-1, 1)  # Reshape for sklearn
        Y = df_filtered[y_col].values

        # fit linear model
        model = LinearRegression()
        model.fit(X, Y)
        # predict vals and calculate
        Y_pred = model.predict(X)
        r2 = r2_score(Y, Y_pred)

        # create fig
        plt.figure(figsize=(8, 6))
        sns.scatterplot(x=X.flatten(), y=Y, color='blue', s=50)
        plt.plot(X.flatten(), Y_pred, color='red', label=f'R² = {r2:.2f}',␣
 ↪linewidth=2)

        # add titles
        plt.title(f'{y_col} vs {x_col}', fontsize=16, fontweight='bold')
        plt.xlabel(x_col, fontsize=12)
        plt.ylabel(y_col, fontsize=12)
        plt.legend(loc='best')

        # show plot
        plt.tight_layout()
        plt.show()
```

[173]:
```
sdf = DataFrameStats(df)

sdf.show_date_range()
```

```
The earliest year in the dataset is: 1984
The latest year in the dataset is: 2019
```

[173]:
```
(1984, 2019)
```

[121]:
```
print(sdf.get_numeric_summary())
```

```
                     Min          Q1     Median        Mean  \
barrels08           0.06   14.330870    16.4805   17.363564
barrelsA08          0.00    0.000000     0.0000    0.220069
charge120           0.00    0.000000     0.0000    0.000000
charge240           0.00    0.000000     0.0000    0.036086
city08              6.00   15.000000    17.0000   18.213318
city08U             0.00    0.000000     0.0000    5.494777
cityA08             0.00    0.000000     0.0000    0.616077
cityA08U            0.00    0.000000     0.0000    0.466164
cityCD              0.00    0.000000     0.0000    0.000471
cityE               0.00    0.000000     0.0000    0.274113
cityUF              0.00    0.000000     0.0000    0.001279
co2                -1.00   -1.000000    -1.0000   80.114069
co2A               -1.00   -1.000000    -1.0000    5.713131
co2TailpipeAGpm     0.00    0.000000     0.0000   17.719449
co2TailpipeGpm      0.00  386.391304   447.0000  468.544572
comb08              7.00   17.000000    20.0000   20.461890
```

```
comb08U              0.00      0.000000      0.0000      6.149154
combA08              0.00      0.000000      0.0000      0.677104
combA08U             0.00      0.000000      0.0000      0.504176
combE                0.00      0.000000      0.0000      0.280361
combinedCD           0.00      0.000000      0.0000      0.000363
combinedUF           0.00      0.000000      0.0000      0.001261
cylinders            2.00      4.000000      6.0000      5.721949
displ                0.00      2.200000      3.0000      3.301581
engId                0.00      0.000000    186.0000   8377.335695
feScore             -1.00     -1.000000     -1.0000      0.238891
fuelCost08         500.00   1950.000000   2350.0000   2377.809935
fuelCostA08          0.00      0.000000      0.0000     90.062623
ghgScore            -1.00     -1.000000     -1.0000      0.237020
ghgScoreA           -1.00     -1.000000     -1.0000     -0.922357
highway08            9.00     20.000000     24.0000     24.350989
highway08U           0.00      0.000000      0.0000      7.282794
highwayA08           0.00      0.000000      0.0000      0.782590
highwayA08U          0.00      0.000000      0.0000      0.575074
highwayCD            0.00      0.000000      0.0000      0.000242
highwayE             0.00      0.000000      0.0000      0.288492
highwayUF            0.00      0.000000      0.0000      0.001237
hlv                  0.00      0.000000      0.0000      2.019585
hpv                  0.00      0.000000      0.0000     10.355630
id                   1.00  10021.000000  20042.0000  20153.739777
lv2                  0.00      0.000000      0.0000      1.814675
lv4                  0.00      0.000000      0.0000      6.139243
pv2                  0.00      0.000000      0.0000     13.556224
pv4                  0.00      0.000000      0.0000     33.823383
range                0.00      0.000000      0.0000      0.616377
rangeCity            0.00      0.000000      0.0000      0.578557
rangeCityA           0.00      0.000000      0.0000      0.067532
rangeHwy             0.00      0.000000      0.0000      0.563910
rangeHwyA            0.00      0.000000      0.0000      0.062571
UCity                0.00     18.110500     21.2965     22.981798
UCityA               0.00      0.000000      0.0000      0.789437
UHighway             0.00     27.661300     33.0246     34.105932
UHighwayA            0.00      0.000000      0.0000      1.076877
year              1984.00   1991.000000   2002.0000   2001.068586
youSaveSpend    -29000.00  -5750.000000  -4000.0000  -4134.565006
charge240b           0.00      0.000000      0.0000      0.007497
phevCity             0.00      0.000000      0.0000      0.122851
phevHwy              0.00      0.000000      0.0000      0.123375
phevComb             0.00      0.000000      0.0000      0.122527

                           Q3           Max
barrels08           19.388824     47.087143
barrelsA08           0.000000     18.311667
charge120            0.000000      0.000000
```

```
charge240            0.000000     12.000000
city08              20.000000    150.000000
city08U             12.273600    150.000000
cityA08              0.000000    145.000000
cityA08U             0.000000    145.083500
cityCD               0.000000      5.350000
cityE                0.000000    122.000000
cityUF               0.000000      0.896000
co2                 -1.000000    847.000000
co2A                -1.000000    713.000000
co2TailpipeAGpm      0.000000    713.000000
co2TailpipeGpm     523.000000   1269.571429
comb08              23.000000    136.000000
comb08U             14.273000    136.000000
combA08              0.000000    133.000000
combA08U             0.000000    133.266200
combE                0.000000    121.000000
combinedCD           0.000000      4.800000
combinedUF           0.000000      0.888000
cylinders            6.000000     16.000000
displ                4.300000      8.400000
engId             4301.000000  69102.000000
feScore             -1.000000     10.000000
fuelCost08        2700.000000   7350.000000
fuelCostA08          0.000000   3800.000000
ghgScore            -1.000000     10.000000
ghgScoreA           -1.000000      8.000000
highway08           28.000000    123.000000
highway08U          17.545700    123.340000
highwayA08           0.000000    121.000000
highwayA08U          0.000000    121.200500
highwayCD            0.000000      4.060000
highwayE             0.000000    120.000000
highwayUF            0.000000      0.877000
hlv                  0.000000     49.000000
hpv                  0.000000    195.000000
id               30331.000000  40434.000000
lv2                  0.000000     41.000000
lv4                 13.000000     55.000000
pv2                  0.000000    194.000000
pv4                 91.000000    192.000000
range                0.000000    335.000000
rangeCity            0.000000    333.111500
rangeCityA           0.000000    103.030000
rangeHwy             0.000000    346.900000
rangeHwyA            0.000000     90.550000
UCity               25.700000    224.800000
UCityA               0.000000    207.262200
```

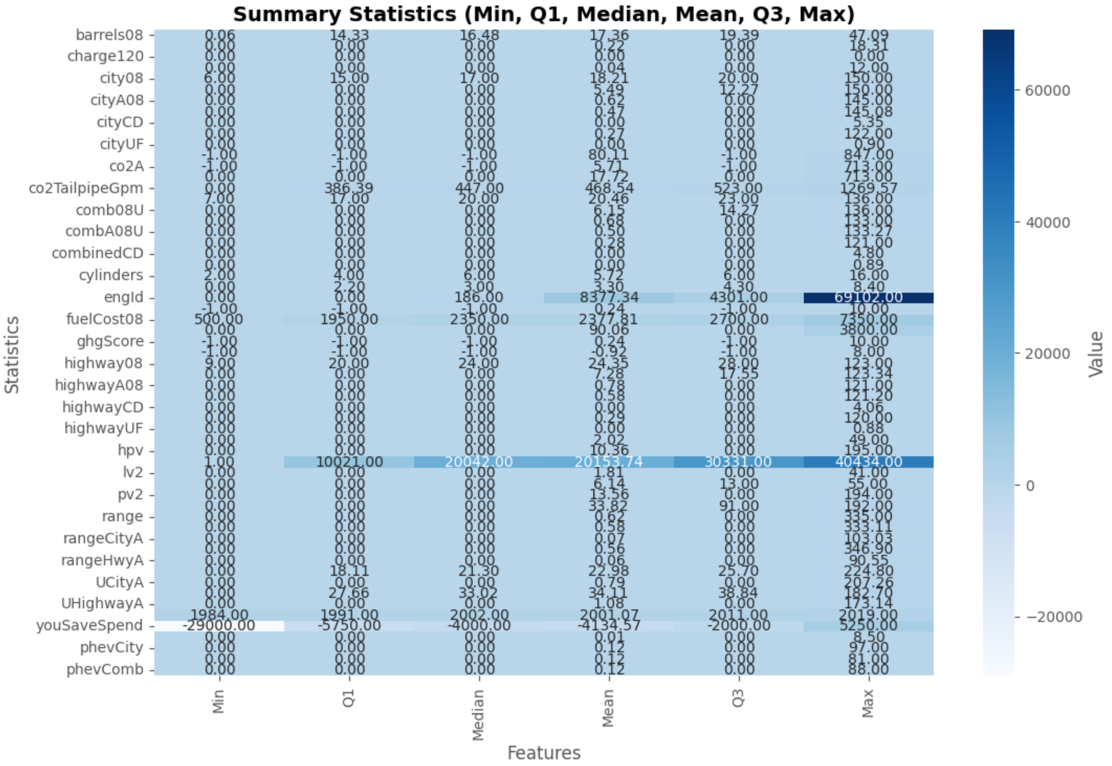
```
UHighway            38.839200    182.700000
UHighwayA            0.000000    173.143600
year              2011.000000   2019.000000
youSaveSpend     -2000.000000   5250.000000
charge240b           0.000000      8.500000
phevCity             0.000000     97.000000
phevHwy              0.000000     81.000000
phevComb             0.000000     88.000000
```

[95]:
```
sdf.get_non_numeric_summary()
```

[95]:
```
{'drive': {'Unique Values': 8,
  'Most Frequent': 'Front-Wheel Drive',
  'Frequency': {'Front-Wheel Drive': 13939,
   'Rear-Wheel Drive': 13539,
   '4-Wheel or All-Wheel Drive': 6648,
   'All-Wheel Drive': 2713,
   '4-Wheel Drive': 1328,
   '2-Wheel Drive': 507,
   'Part-time 4-Wheel Drive': 217,
   'Automatic (A1)': 1}},
 'eng_dscr': {'Unique Values': 550,
  'Most Frequent': '(FFS)',
  'Frequency': {'(FFS)': 8827,
   'SIDI': 4902,
   '(FFS) CA model': 926,
   '(FFS)      (MPFI)': 734,
   'FFV': 683,
   '(FFS,TRBO)': 666,
   '(350 V8) (FFS)': 411,
   '(GUZZLER)  (FFS)': 366,
   'SOHC': 354,
   'SIDI; FFV': 287,
   '(NO-CAT)': 238,
   'SIDI & PFI': 225,
   'FLEX-FUEL': 198,
   'GUZZLER': 195,
   '(FFS)      (SPFI)': 194,
   '(GUZZLER)  (FFS)      (MPFI)': 122,
   '(350 V8)': 120,
   'CA model': 113,
   '(350 V8) (FFS)  (MPFI)': 106,
   '(GM-CHEV)': 102,
   'DOHC       (FFS)': 96,
   '(DIESEL)': 95,
   'PR': 91,
   '(GUZZLER) (FFS)': 84,
   '(FFS,TRBO) CA model': 81,
   'DOHC': 79,
   'SOHC       (FFS)': 78,
   'DOHC TURBO (FFS,TRBO)': 76,
   'V-6': 75,
   '(305)      (FFS)': 71,
   '(DIESEL) CA model': 71,
   '(CAL)(FFS)': 67,
   'SIDI; PHEV': 64,
   '(DSL,TRBO)': 60,
   'SOHC-4     (FFS)': 50,
   'HEV': 50,
   '(GM-CHEV)  (FFS)': 46,
   '(CALIF)': 45,
   '(DOHC)     (FFS)': 44,
   'PHEV': 42,
   '(GUZZLER)': 42,
   'DOHC-IL4': 42,
   '(GUZZLER)  (FFS,TRBO)': 40,
   '(305)': 40,
   '(SOHC)     (FFS)': 39,
   'LM7': 38,
   'SOHC-4 2WD (FFS)': 37,
   '(DSL,TRBO) (NO-CAT)': 37,
   '(FFS) fuel injection': 37,
   'SIDI; Stop-Start': 35,
   'FFS': 34,
   'VTEC': 33,
   '(4A-FE)    (FFS)': 32,
   'SOHC-4 4WD (FFS)': 32,
   '(FFS)      (S-CHARGE)': 32,
   '(350 V8) (DIESEL)': 31,
   '(350 V8) (FFS) CA model': 30,
   '(FFS,TRBO) (MPFI)': 29,
   '(FFS) 2 barrel carb': 29,
   'V6': 28,
   '(POLICE)   (FFS)': 28,
   '(307)      (FFS)': 27,
   'SOHC-VTEC': 27,
   '(MPFI)     (NO-CAT)': 26,
   'B235R': 26,
   'DOHC-VTEC': 25,
   '(DSL,TRBO) CA model': 24,
   '(DSL,TRBO) (MPFI)': 24,
   '4V': 24,
   'Coupe or Conv.': 23,
   '(GM-CHEV) CA model': 23,
   '(GM-OLDS)  (FFS)': 21,
   '(3S-FE)    (FFS)': 21,
   '(DIESEL) (NO-CAT)': 20,
   'GAS 330': 20,
   '(FFS) (MPFI)': 20,
   'I4': 20,
   '(FFS)      (GUZZLER)': 20,
   '4-VALVE': 19,
   'SIDI; with Stop-Start option': 18,
   'SIDI; with Stop-Start Option': 18,
   'SPORTS': 18,
   'MOTORSPORT': 18,
   '(16-VALVE) (FFS)': 18,
   '(VTEC)     (FFS)': 17,
   'L410MT2': 17,
   '(CALIF)    (FFS,TRBO)': 17,
   'DOHC-T/C': 16,
   'L-4': 16,
   '(121)      (FFS)': 16,
   '(MPFI)': 16,
   '4 VALVE': 16,
   '(GUZZLER) CA model': 16,
   'B235E': 16,
   '(GUZZLER)  (FFS,TRBO) (MPFI)': 16,
   '(GM-CHEV)  (FFS) CA model': 15,
   '(DOHC)     (FFS,TRBO)': 15,
   '(16VALVES) (FFS)': 14,
   'B205R': 14,
   'VTEC       (FFS)': 14,
   '(122)      (FFS)': 14,
   '2-VALVE': 14,
   '390-540': 14,
   '(GM-OLDS) CA model': 13,
   '(GM-OLDS)': 13,
   'SMG TRANS': 13,
   '(4-VALVE)  (FFS)': 13,
   'DOHC TURBO': 12,
   'SIDI; with i-ELOOP Technology Package': 12,
   'Hellcat engine': 12,
   '(GM-CHEV)  (FFS)      (MPFI)': 12,
   '2V': 12,
   'SIDI; PZEV (SULEV) emissions': 12,
   '(A-ENGINE) (FFS,TRBO)': 12,
   '2.0Z': 11,
   'VCM': 11,
   'VTEC-E': 11,
   '(TURBO)    (FFS,TRBO)': 11,
   '(GM-BUICK) CA model': 11,
   '275HP': 11,
   '(2-VALVE)  (FFS)': 10,
   '(GUZZLER)  (TURBO)': 10,
   '(MPFI)     (TURBO)': 10,
   'SIDI; Hybrid': 10,
   'SIDI; Z06': 10,
   'SOHC-IL4': 10,
   'SOHC-3     (FFS)': 10,
   '(350 V8) (POLICE)   (FFS)': 9,
   '(4A-GE)    (FFS)': 9,
   '4 valve': 9,
   'B205L': 8,
   'DOD': 8,
   'GAS 340': 8,
   'B207R': 8,
   'B308E': 8,
   '(GM-OLDS)  (FFS) CA model': 8,
   '(OHC)': 8,
   '4-Valve': 8,
   'CNG': 8,
   '(EGR) (FFS)': 8,
   'BPE        (FFS)': 8,
   '4V DOHC': 8,
   '(GM-CHEV)  (FFS)      (SPFI)': 8,
   '(EGR)      (FFS)': 8,
   '(FFS)      (MPFI)     (S-CHARGE)': 8,
   'SOHC-V6    (FFS)': 8,
   '4.6N': 8,
   '(B202)     (FFS,TRBO)': 7,
   'V-6        (FFS)': 7,
   '(GM-BUICK)': 7,
   '(TURBO)': 7,
   '(GUZZLER)  (L410MT2)  (FFS,TRBO)': 7,
   'B202       (FFS,TRBO)': 7,
   '2 VALVE': 6,
   'SC': 6,
   'GAS 360': 6,
   '(FFS/TRBO)': 6,
   '0.0004': 6,
   '(OHC)      (FFS)': 6,
   '4.0E-R': 6,
   'VTC/VTEC': 6,
   '300HP': 6,
   'FLEX FUEL': 6,
   '440-580': 6,
   'DOHC       (FFS)      (MPFI)': 6,
   'DOHC/VTEC  (FFS)': 6,
   'SIDI; ZL1': 6,
   '2-mode (Normal/Eco) Transmission': 6,
   '164/164L   (FFS)      (MPFI)': 6,
   'SOHC  4WD  (FFS)': 6,
   'SIDI;': 6,
   '(3S-GE)    (FFS)': 6,
   '(FFS/TRBO) CA model': 6,
   '500': 6,
   'Off Road Package': 6,
   '(MFI)      (FFS)': 6,
   '(TBI)      (FFS)': 6,
   'SOHC       (FFS)      (MPFI)': 6,
   '(8-VALVE)  (FFS)': 6,
   'SOHC  2WD  (FFS)': 6,
   'SOHC FFS': 6,
   'POLICE  FFS': 6,
   'B204L3 FFS,TURBO': 6,
   'DOHC-TC': 6,
   '(FFS) DOHC': 6,
   'DCT TRANS': 6,
   'Sold at Ford dealerships': 6,
   '440-470': 6,
   '2.5M': 6,
   '430-470': 6,
   'SIDI & PFI; FFV': 6,
   '2-Valve': 5,
   '(FFS) Lock-up': 5,
   '(4A-F)     (FFS)': 5,
   'COUPE': 5,
   'B202       (FFS)': 5,
   '(DSL,TRBO) (MPFI)     (NO-CAT)': 5,
   '(FFS)   FI': 5,
   '(350 V8) (FFS)   FI': 5,
   '(FFS)(SIL) CA model': 5,
   'DI': 5,
   'GUZZLER  FFS': 5,
   'AFM': 5,
   '2.0L CVH': 5,
   '4.2E-R': 5,
   '4.6E-R': 5,
   '5.4E-R': 5,
   'DOHC-4     (FFS)': 5,
   '(4-VALVE)  (FFS)      (MPFI)': 5,
   '(FFS,TRBO) (ROTARY)': 5,
   '(GUZZLER)             (FFS)': 5,
   'DOHC-VTEC  (FFS)': 5,
   'NGV': 5,
   'Part-time 4WD': 4,
   'BPD        (FFS)': 4,
   'DOHC V-6   (FFS)': 4,
   '(FFS)      (ROTARY)': 4,
   '(FFS) 4 barrel carb': 4,
   '2.5E-R': 4,
   'B234I3     (FFS)': 4,
   'SOHC 4': 4,
   'Full Time 4WD': 4,
   'Part Time 4WD': 4,
   'GAS 380': 4,
   'RNG=170': 4,
   'new body style': 4,
   'B204L3 (FFS,TRBO)': 4,
   'GUZZLER POLICE 4.6N': 4,
   '(SOHC) (FFS)  FI': 4,
   'B204L3     (FFS,TRBO)': 4,
   'FFS,TURBO': 4,
   'RNG420': 4,
   '(OHV)': 4,
   'SIDI; Ecoboost': 4,
   '460': 4,
   'GAS 370': 4,
   'GMP4 ANDIC (FFS,TRBO) (MPFI)': 4,
   '(20-VALVE) (FFS)': 4,
   'Lock-up': 4,
   'FFS MPFI': 4,
   'SIDI; eAssist': 4,
   '(GUZZLER)  (L410MN2)  (FFS)': 4,
   '420-540': 4,
   '3.0L 2V': 4,
   'RNG390/520': 4,
   '(GUZZLER)  (FFS) CA model': 4,
   '(C-ENGINE) (FFS)      (MPFI)': 4,
   'SOHC-VIS': 4,
   'RNG=390': 4,
   '3.8E-F': 4,
   'SIDI; PZEV(SULEV) emissions': 4,
   'V6-SOHC-2  (FFS)': 4,
   'LEAN-BURN': 4,
   'B207L': 4,
   'V6-SOHC-4  (FFS)': 4,
   'V6-SOHC    (FFS)': 4,
   'DIESEL CA model': 4,
   'B234I3': 4,
   '(SPFI)': 4,
   'B234I3 (FFS)': 4,
   'I-4        (FFS)': 4,
   '(OHV)      (FFS)': 4,
   '(CAL,FFS)': 4,
   '(FFS)(SIL)': 4,
   '(B-ENGINE) (FFS,TRBO)': 4,
   '4VALVE     (FFS)': 4,
   '(POLICE)   (FFS) 2 barrel carb': 4,
   '(POLICE)   (FFS) 4 barrel carb': 4,
   'SOHC-2     (FFS)': 4,
   'ASTON-DB7': 4,
   '(20-VALVE) (FFS,TRBO)': 4,
   'SUPERCHR': 4,
   'B235L': 4,
   '(B202)     (FFS)': 4,
   'Lead Acid': 4,
   'NiMH': 4,
   'LEAN BURN': 4,
   '(GUZZLER)  ASTON-DB7  (FFS)': 4,
   '(GUZZLER)  (L410MNKT) (FFS)': 4,
   'POLICE': 4,
   'DOHC I-4   (FFS)': 4,
   'RNG=340': 3,
   'CVT2L': 3,
   '(BENDIX)   (FFS)': 3,
   '(CALIF) CA model': 3,
   'SOHC V-6': 3,
   'RNG390': 3,
   '(FFS,SPFI) CA model': 3,
   'Direct Injection': 3,
   '(FFS,MPFI) CA model': 3,
   '3.8N': 3,
   'Hybrid; PR': 3,
   'RNG=360': 3,
   '255HP': 3,
   'SIDI; with Stop/Start Technology': 3,
   '530': 3,
   '(GUZZLER) (GUZZLER)': 3,
   'B235E5 FFS,TURBO': 3,
   '0.00046': 3,
   '(ROTARY)': 3,
   '(GUZZLER)  (L410MTKT) (FFS,TRBO)': 3,
   'SIDI; FFV; Sport Transmission': 3,
   '(CALIF)    (DSL,TRBO)': 3,
   '448S': 3,
   '(16-VALVE) (FFS)      (MPFI)': 3,
   '3.0M': 3,
   'GUZZLER POLICE': 3,
   '(FFS) Fuel Injection': 3,
   '(FFS) SOHC': 3,
   'RNG=380': 3,
   'ELAN TURBO (FFS,TRBO)': 3,
   'SOHC I-4   (FFS)': 2,
   'Mild Hybrid': 2,
   '(FFS)      (MPFI) DOHC': 2,
   '(4-VLV)    (FFS)': 2,
   'SOHC-4     (FFS)': 2,
   '(GUZZLER) (FFS) (S-CHARGE)': 2,
   '(FFS) (DIESEL) (NO-CAT)': 2,
   'V6-DOHC    (FFS)': 2,
   'DOHC VTEC': 2,
   'SIDI; Mild Hybrid': 2,
   'EGR/2-VLV  (FFS)': 2,
   '(16 VALVE) (FFS)      (MPFI)': 2,
   'FFV; Part-time 4-Wheel Drive': 2,
   'RNG=193': 2,
   '3-mode (Power/Normal/Eco) Transmission': 2,
   'B201       (FFS)': 2,
   '(PRE-CAT) (FFS)': 2,
   'SIDI; Ecoboost; GVWR>7599 LBS': 2,
   'SIDI & PFI; VCR': 2,
   'B258I3     (FFS)': 2,
   'FFV; GVWR>7599 LBS': 2,
   '(350 V8) POLICE FFS MPFI': 2,
   '(FFS) (SPFI)': 2,
   '(ROTARY) CA model': 2,
   'B234E4     (FFS,TRBO)': 2,
   'B234L/R4   (FFS,TRBO)': 2,
   'FFV; PR': 2,
   '4V         (FFS)': 2,
   'NONE FFS': 2,
   'PRE-CAT    (FFS)': 2,
   'Part-time 4-Wheel Drive': 2,
   'CNG RNG180': 2,
   'B258I3 (FFS)': 2,
   'SIDI; Short Wheelbase': 2,
   'Stop-Start': 2,
   '(FFS) 3 barrel carb': 2,
   'SIDI; Long Wheelbase': 2,
   'SOHC-4 (FFS)': 2,
   'EGR/4-VLV  (FFS)': 2,
   '(POLICE) (FFS)': 2,
   'VTEC-E     (FFS)': 2,
   '2VALVE     (FFS)': 2,
   '(GM-CHEV) (FFS)': 2,
   '(GUZZLER)  FFS MPFI   (FFS)': 2,
   '3.0W-F': 2,
   'RNG=120': 2,
   '164S       (FFS)      (MPFI)': 2,
   'ESPRIT     (FFS,TRBO) (MPFI)': 2,
   '0.00054': 2,
   'HEV LB': 2,
   '(350 V8) (GUZZLER)  (FFS)': 2,
   '0.00042': 2,
   'JED        (FFS)': 2,
   'JEE        (FFS)': 2,
   '(FFS) BOSCH': 2,
   '275 hp': 2,
   '275 HP': 2,
   '300 HP': 2,
   'Gasoline only': 2,
   '4.6W': 2,
   '(FFS,TRBO) (SPFI)': 2,
   'LPT': 2,
   'STS/DTS': 2,
   '(PRE-CAT)  (FFS)': 2,
   'B6D        (FFS)': 2,
   'AWD': 2,
   '(FFS TURBO)': 2,
   'B6E2       (FFS)': 2,
   'BPD/DOHC   (FFS)': 2,
   'BPE/SOHC   (FFS)': 2,
   'SST': 2,
   '0.0005': 2,
   'SUPERCHRGD': 2,
   '(GUZZLER)  5 SERIES   (FFS)': 2,
   'K8D        (FFS)': 2,
   'B6E4       (FFS)': 2,
   '(GUZZLER)  (GUZZLER)  (FFS)': 2,
   'BI-FUEL': 2,
   '5.0E-R': 2,
   'MIATA T/C': 2,
   'GT/GTS-VIS': 2,
   'GTS-VIS': 2,
   'V6DOHC     (FFS)': 2,
   'Flex Fuel (E85)': 2,
   '275': 2,
   'CALIF. ECS (FFS)      (MPFI)': 2,
   'B235E5': 2,
   'GAS 440': 2,
   'B204R3': 2,
   '450': 2,
   '430': 2,
   '4 VALVES   (FFS)': 2,
   '(B201)     (FFS)': 2,
   '(FFS)      (MPFI) SOHC': 2,
   'B204L3': 2,
   'Hybrid': 2,
   'B204L3': 2,
   '3.0N-F': 2,
   '3.0E-F': 2,
   'B234R4': 2,
   '(2 VALVE)  (FFS)      (MPFI)': 2,
   '(4A-LC)    (FFS)': 2,
   'SOHC VTEC': 2,
   '2 VALVES   (FFS)': 2,
   'RNG=450': 2,
   '(350 V8) (GUZZLER)  (FFS,TRBO)': 2,
   'SELECT SFT': 2,
   'GAS 350': 2,
   'DIRECTINJ': 2,
   '16-VALVE   (FFS)      (MPFI)': 2,
   '470-610': 2,
   '3.9L4V-N': 2,
   'RNG=420': 2,
   '400-425': 2,
   '(V6-DOHC)  (FFS)': 1,
   '(GUZZLER)  (VOLVO780) (FFS)': 1,
   'Sport Transmission': 1,
   'SIDI; ULEV Emissions': 1,
   'SIDI; Over 6000 lbs curb weight': 1,
   'SIDI; Ecoboost; GVWR>6649 LBS': 1,
   '(FFS) 1 barrel carb': 1,
   '(164S)     (FFS)      (MPFI)': 1,
   '(3S-GTE)   (FFS,TRBO)': 1,
   '(GUZZLER)  (TURBO) CA model': 1,
   'California Emission Control System': 1,
   'Z/28': 1,
   'SIDI; Ecoboost; GVWR>6799 LBS': 1,
   'PHEV; 94 Amp-hour battery': 1,
   'SIDI & PFI; Stop-start': 1,
   'SIDI                                   S': 1,
   '16-V, CAL. (FFS)      (MPFI)': 1,
   'GMP4       (FFS,TRBO) (MPFI)': 1,
   '(GMP4+IC)  (FFS,TRBO) (MPFI)': 1,
   '(VOLVO760) (FFS)': 1,
   'California emission control system': 1,
   'SIDI & PFI; with Off Road Package': 1,
   'RNG=192': 1,
   '(SPG)      (FFS,TRBO)': 1,
   'SIDI & PFI; Stop-Start': 1,
   'N        (FFS)': 1,
   'V-6        (FFS)      (S-CHARGE)': 1,
   '(GMP4)     (FFS,TRBO) (MPFI)': 1,
   'RNG200/300': 1,
   'SOHC V-6   (FFS)': 1,
   '(FFS,TRBO) Low Boost': 1,
   'E2.0LMA': 1,
   'E2.5CAB': 1,
   'E2.5CMB': 1,
   '(16-VALVE) (FFS,TRBO)': 1,
   '4.0N': 1,
   'MAZDA3 T/C': 1,
   'MAZDA6 T/C': 1,
   'GAS 410': 1,
   'GAS 400': 1,
   'GAS 420': 1,
   'RNG=200': 1,
   '3': 1,
   '320/510': 1,
   '320-500': 1,
   '4.6M': 1,
   'B308E5': 1,
   'RNG=390/400': 1,
   '54RA12': 1,
   '44RA8': 1,
   'GAS 430': 1,
   'RNG=370': 1,
   'E2.0LAA': 1,
   '5.4S/C': 1,
   'LK9': 1,
   '4.0S-4 SOHC': 1,
   'RNG250': 1,
   'SUPER-CHGD': 1,
   'RNG140/220': 1,
   'RNG=290': 1,
   'SPORT': 1,
   'NG': 1,
   'NONE': 1,
   '4.0S-4': 1,
   '5.4V-4': 1,
   '5.4V-R': 1,
   '16-VALVE   (FFS)': 1,
   'IN COMMENT': 1,
   'RNG130/170': 1,
   '3.3N-F': 1,
   '2-MODE': 1,
   '3-VALVE': 1,
   'RNG=250': 1,
   '200': 1,
   'EX2US40E5G': 1,
   'DVVT': 1,
   'RNG=400': 1,
   'Dual clutch transmission': 1,
   'EFFV-GAS': 1,
   'Full time 4WD': 1,
   'DSL, TRBO': 1,
   'GAS 420 FFS': 1,
   '(GUZZLER)  (GM-CHEVY) (FFS)': 1,
   '(FFS) (MPFI)  FI': 1,
   '(FFS) (SPFI)  FI': 1,
   'FFV; Active Fuel Management': 1,
   'AWD (All Wheel Drive)': 1,
   '(FFS,DOHC)': 1,
   '63 and 67 motor/generators': 1,
   'Part time 4WD': 1,
   'GUZZLER  FFS,TURBO': 1,
   'Off-road Package': 1,
   'Cabrio model': 1,
   '(GUZZLER)  (FFS)      (S-CHARGE)': 1,
   'SIDI; Select shift transmission': 1,
   'FFV; Cargo Van': 1,
   'RNG=190': 1,
   'Shelby GT500; Coupe or Conv.': 1,
   '(FFS)      (VARIABLE)': 1,
   'SIDI; PZEV (SULEV) Emissions': 1,
   'GUZZLER V8 FFS': 1,
   '(NGV)      (FFS)': 1,
   'MFFV-GAS': 1,
   '(GUZZLER) (FFS) (MPFI)': 1,
   'SOHC L-5': 1,
   'V8': 1,
   '(LH-3.1) (FFS)': 1,
   '(GUZZLER) FFS MPFI (FFS)': 1,
   '(BOSCH) (FFS)': 1,
   '(FFS)      (FFS)': 1,
   '(4- VALVE) (FFS)': 1,
   '(350 V8) (GUZZLER) (POLICE) (FFS)': 1,
   'R-ENG (FFS,TRBO)': 1,
   'B308I4     (FFS)': 1,
   'V-6 FFS': 1,
   'B234E4 (FFS,TRBO) High Power': 1,
   'B234E4 (FFS,TRBO)': 1,
   'B234L/R4 (FFS,TRBO)': 1,
   'GUZZLER V8 FFS,TURBO': 1,
   '4.6M FFS MPFI': 1,
   'CNG FFS': 1,
   'POLICE FFS MPFI': 1,
   'B308E5 FFS,TURBO': 1,
   '5.4E-R FFS MPFI': 1,
   'B308I4 (FFS) (VARIABLE)': 1}},
 'fuelType': {'Unique Values': 14,
  'Most Frequent': 'Regular',
  'Frequency': {'Regular': 25997,
   'Premium': 11067,
   'Gasoline or E85': 1287,
   'Diesel': 1142,
   'Electricity': 168,
   'Premium or E85': 125,
   'Midgrade': 100,
   'CNG': 60,
   'Premium and Electricity': 47,
   'Regular Gas and Electricity': 29,
   'Premium Gas or Electricity': 28,
   'Gasoline or natural gas': 20,
   'Gasoline or propane': 8,
   'Regular Gas or Electricity': 3}},
 'fuelType1': {'Unique Values': 6,
  'Most Frequent': 'Regular Gasoline',
  'Frequency': {'Regular Gasoline': 27344,
   'Premium Gasoline': 11267,
   'Diesel': 1142,
   'Electricity': 168,
   'Midgrade Gasoline': 100,
   'Natural Gas': 60}},
 'make': {'Unique Values': 135,
  'Most Frequent': 'Chevrolet',
  'Frequency': {'Chevrolet': 3944,
   'Ford': 3284,
   'Dodge': 2559,
   'GMC': 2471,
   'Toyota': 2010,
   'BMW': 1856,
   'Mercedes-Benz': 1444,
   'Nissan': 1424,
   'Volkswagen': 1155,
   'Mitsubishi': 1052,
   'Porsche': 1028,
   'Mazda': 984,
   'Honda': 970,
   'Audi': 969,
   'Jeep': 900,
   'Pontiac': 893,
   'Subaru': 880,
   'Volvo': 808,
   'Hyundai': 735,
   'Chrysler': 718,
   'Buick': 659,
   'Mercury': 609,
   'Cadillac': 575,
   'Kia': 572,
   'Plymouth': 526,
   'Suzuki': 515,
   'Lexus': 463,
   'Oldsmobile': 462,
   'Isuzu': 434,
   'Saab': 432,
   'Jaguar': 429,
   'MINI': 405,
   'Infiniti': 397,
   'Acura': 331,
   'Lincoln': 325,
   'Saturn': 278,
   'Ferrari': 229,
   'Rolls-Royce': 179,
   'Land Rover': 176,
   'Eagle': 161,
   'Aston Martin': 150,
   'Geo': 147,
   'Bentley': 125,
   'Maserati': 125,
   'Lamborghini': 117,
   'Peugeot': 98,
   'Scion': 84,
   'Ram': 67,
   'Daewoo': 67,
   'Fiat': 65,
   'Tesla': 58,
   'Roush Performance': 57,
   'Lotus': 57,
   'Renault': 56,
   'Alfa Romeo': 55,
   'J.K. Motors': 36,
   'smart': 36,
   'Wallace Environmental': 32,
   'Genesis': 32,
   'Maybach': 31,
   'American Motors Corporation': 27,
   'Hummer': 19,
   'CX Automotive': 17,
   'Daihatsu': 17,
   'Merkur': 14,
   'Federal Coach': 14,
   'Spyker': 13,
   'Import Trade Services': 13,
   'McLaren Automotive': 13,
   'Sterling': 12,
   'Dabryan Coach Builders Inc': 9,
   'Bugatti': 9,
   'Yugo': 8,
   'Bertone': 7,
   'Mcevoy Motors': 6,
   'BYD': 6,
   'Pininfarina': 6,
   'Tecstar, LP': 6,
   'AM General': 6,
   'Bitter Gmbh and Co. Kg': 5,
   'VPG': 5,
   'Saleen Performance': 5,
   'Saleen': 5,
   'Bill Dovell Motor Car Company': 4,
   'Grumman Olson': 4,
   'Autokraft Limited': 4,
   'Mobility Ventures LLC': 4,
   'Vector': 4,
   'TVR Engineering Ltd': 4,
   'Kenyon Corporation Of America': 4,
   'Panther Car Company Limited': 4,
   'Texas Coach Company': 4,
   'BMW Alpina': 3,
   'Morgan': 3,
   'Ruf Automobile Gmbh': 3,
   'Consulier Industries Inc': 3,
   'Evans Automobiles': 3,
   'Pagani': 3,
   'Dacia': 3,
   'SRT': 2,
   'PAS, Inc': 2,
   'CODA Automotive': 2,
   'Laforza Automobile Inc': 2,
   'CCC Engineering': 2,
   'Azure Dynamics': 2,
   'PAS Inc - GMC': 2,
   'Red Shift Ltd.': 2,
   'Quantum Technologies': 2,
   'Avanti Motor Corporation': 2,
   'Mahindra': 1,
   'Aurora Cars Ltd': 1,
   'Isis Imports Ltd': 1,
   'Goldacre': 1,
   'London Coach Co Inc': 1,
   'General Motors': 1,
   'Shelby': 1,
   'Environmental Rsch and Devp Corp': 1,
   'Grumman Allied Industries': 1,
   'JBA Motorcars, Inc.': 1,
   'Import Foreign Auto Sales Inc': 1,
   'S and S Coach Company  E.p. Dutton': 1,
   'Karma': 1,
   'Koenigsegg': 1,
   'Panoz Auto-Development': 1,
   'E. P. Dutton, Inc.': 1,
   'Superior Coaches Div E.p. Dutton': 1,
   'Vixen Motor Company': 1,
   'Volga Associated Automobile': 1,
   'Qvale': 1,
   'ASC Incorporated': 1,
   'Panos': 1,
   'London Taxi': 1,
   'Fisker': 1,
   'Excalibur Autos': 1,
   'Lambda Control Systems': 1}},
 'model': {'Unique Values': 3960,
  'Most Frequent': 'F150 Pickup 2WD',
  'Frequency': {'F150 Pickup 2WD': 215,
   'F150 Pickup 4WD': 193,
   'Mustang': 192,
   'Jetta': 190,
   'Truck 2WD': 187,
   'Camaro': 174,
   'Ranger Pickup 2WD': 169,
   'Accord': 152,
   'Civic': 152,
   'Sierra 1500 4WD': 149,
   'Sierra 1500 2WD': 149,
   'Eclipse': 133,
   'Corolla': 129,
   'Camry': 125,
   'Sentra': 124,
   'S10 Pickup 2WD': 118,
   'Truck 4WD': 113,
   'Ranger Pickup 4WD': 113,
   'D100/D150 Pickup 2WD': 112,
   'Dakota Pickup 2WD': 108,
   'Colt': 107,
   'C1500 Pickup 2WD': 106,
   'F250 Pickup 2WD': 106,
   'E150 Econoline 2WD': 105,
   'Corvette': 104,
   'Cavalier': 101,
   'K1500 Pickup 4WD': 101,
   'Lancer': 100,
   'Passat': 100,
   'Tacoma 2WD': 98,
   'Ram 1500 Pickup 2WD': 98,
   'Mirage': 97,
   'G10/20 Van 2WD': 96,
   'Sonata': 96,
   '4Runner 4WD': 94,
   'Celica': 91,
   'Escort': 89,
   'Tacoma 4WD': 89,
   'Caravan/Grand Caravan 2WD': 88,
   'Regal': 88,
   'Dakota Pickup 4WD': 88,
   'Impreza AWD': 87,
   'Grand Prix': 87,
   'Bronco 4WD': 86,
   'B150/B250 Van 2WD': 85,
   'B150/B250 Wagon 2WD': 81,
   'Grand Am': 81,
   'Legacy AWD': 81,
   'Wrangler 4WD': 79,
   'E150 Club Wagon': 77,
   'Charger': 77,
   'Elantra': 75,
   '900': 75,
   'A4 quattro': 74,
   'Voyager/Grand Voyager 2WD': 74,
   'Golf': 74,
   'D250 Pickup 2WD': 73,
   'Pickup 2WD': 72,
   'Galant': 72,
   'Grand Cherokee 4WD': 71,
   'E250 Econoline 2WD': 71,
   'Colt Vista': 70,
   'Prelude': 70,
   'Thunderbird': 68,
   'C10 Pickup 2WD': 67,
   'C15 Pickup 2WD': 67,
   'G10/20 Sport Van 2WD': 65,
   'Passat Wagon': 64,
   'Frontier 2WD': 64,
   'Talon': 64,
   'G15/25 Rally 2WD': 64,
   'W100/W150 Pickup 4WD': 64,
   'Maxima': 64,
   'Sebring': 63,
   'Forester AWD': 63,
   'Altima': 61,
   'S15 Pickup 2WD': 61,
   'New Beetle': 61,
   'Taurus': 61,
   '300ZX': 60,
   'Optima': 60,
   'Century': 60,
   'CTS': 59,
   'G15/25 Vandura 2WD': 59,
   'Daytona': 59,
   'LeBaron': 59,
   'Ram 1500 Pickup 4WD': 58,
   'Sonoma 2WD': 58,
   'Cougar': 58,
   'Pathfinder 4WD': 57,
   'Escort Wagon': 57,
   'Impala': 57,
   'F250 Pickup 4WD': 57,
   'Cherokee 4WD': 57,
   'Sierra K15 4WD': 56,
   '6': 56,
   'Cherokee 2WD': 56,
   'A6 quattro': 56,
   'Tercel': 56,
   'GTI': 55,
   'Sierra C15 2WD': 55,
   'Silverado K15 4WD': 55,
   '9000': 55,
   '626': 55,
   'Laser': 55,
   'V70 FWD': 54,
   'Grand Cherokee 2WD': 54,
   'Stratus': 54,
   'S10 Blazer 2WD': 54,
   'Shadow': 54,
   'Montero': 53,
   'Pathfinder 2WD': 53,
   'Tundra 2WD': 53,
   'Cutlass Supreme': 53,
   'A4': 52,
   'Silverado C15 2WD': 52,
   'Colorado 2WD': 51,
   'Explorer 4WD': 51,
   'Sierra 2500 2WD': 51,
   'Canyon 2WD': 51,
   'LeBaron Convertible': 51,
   'Bonneville': 51,
   'Malibu': 51,
   '4Runner 2WD': 50,
   'Astro 2WD (cargo)': 50,
   'SC': 50,
   'Boxster': 50,
   'Safari 2WD (cargo)': 50,
   'Explorer 2WD': 50,
   'Cutlass Ciera': 50,
   'K2500 Pickup 4WD': 49,
   'K15 Pickup 4WD': 49,
   'K10 Pickup 4WD': 49,
   'A4 Avant quattro': 49,
   'Prizm': 49,
   'Supra': 48,
   'Range Rover': 48,
   'C2500 Pickup 2WD': 48,
   'SL': 48,
   'Silverado 1500 2WD': 48,
   'Sunbird': 47,
   'S60 FWD': 47,
   'Firebird/Trans Am': 47,
   'Ram 50 Pickup 2WD': 47,
   'Cavalier Wagon': 47,
   'Sunfire': 46,
   'Taurus Wagon': 46,
   'C2500 Sierra 2WD': 46,
   'Silverado 1500 4WD': 46,
   'K2500 Sierra 4WD': 46,
   'Colorado 4WD': 46,
   'Canyon 4WD': 46,
   'Skylark': 46,
   'Trooper': 46,
   'S15 Jimmy 2WD': 45,
   'Topaz': 45,
   'Sportage 2WD': 45,
   'Tempo': 45,
   'Beretta': 45,
   'Pickup 2500 2WD': 45,
   'Versa': 44,
   'Yukon 1500 2WD': 44,
   'Eclipse Spyder': 44,
   'Challenger': 44,
   'Rodeo 2WD': 44,
   'Tahoe 1500 2WD': 43,
   'Sable': 43,
   'Santa Fe 2WD': 43,
   'Tiburon': 43,
   'Tundra 4WD': 42,
   '3': 42,
   'Monte Carlo': 42,
   'Yukon 1500 4WD': 42,
   'Avenger': 42,
   'Integra': 42,
   'Sportage 4WD': 42,
   '9-3 Convertible': 42,
   'Sable Wagon': 42,
   'Matrix': 41,
   'W250 Pickup 4WD': 41,
   'Century Wagon': 41,
   'Probe': 41,
   'Astro 2WD (passenger)': 41,
   'Sebring Convertible': 40,
   'Boxster S': 40,
   'Skyhawk': 40,
   'Legacy Wagon AWD': 40,
   '6000': 40,
   'Lynx': 40,
   'Equinox FWD': 39,
   'Impulse': 39,
   'MX-5 Miata': 39,
   'Equinox AWD': 39,
   'Durango 2WD': 39,
   'Firenza': 38,
   'A8 L': 38,
   'Rio': 38,
   'MX-5': 38,
   'Swift': 38,
   'Protege': 38,
   'Suburban 1500 2WD': 38,
   'Focus': 38,
   'Fusion FWD': 38,
   'Accent': 38,
   '505 Sedan': 38,
   'Focus FWD': 37,
   'MR2': 37,
   'S10 Pickup 4WD': 37,
   'Capri': 37,
   'Celebrity Wagon': 37,
   'Tahoe 1500 4WD': 37,
   'T15 (S15) Pickup 4WD': 37,
   'Vandura G15/25 2WD': 37,
   'Soul': 37,
   'T10 (S10) Pickup 4WD': 37,
   'C25 Pickup 2WD': 37,
   'C20 Pickup 2WD': 37,
   'SW': 36,
   'B350 Van 2WD': 36,
   'Spirit': 36,
   'RX-7': 36,
   'Escape FWD': 36,
   '911 Carrera 4/2': 36,
   'Fit': 36,
   'Intrepid': 36,
   'Odyssey': 36,
   'Savana 1500/2500 2WD (cargo)': 35,
   'Reliant': 35,
   'RAV4 2WD': 35,
   'Safari 2WD (passenger)': 35,
   'T10 (S10) Blazer 4WD': 35,
   'LeSabre': 35,
   'Celebrity': 35,
   '9-3 Sport Sedan': 35,
   'Van 1500/2500 2WD': 35,
   'Corsica': 35,
   'AW100/AW150 Ramcharger 4WD': 35,
   '6000 Wagon': 35,
   'Aries': 35,
   'T15 (S15) Jimmy 4WD': 35,
   '911 Carrera': 35,
   'Firebird': 35,
   'M3': 35,
   'Summit Wagon': 34,
   'RAV4 4WD': 34,
   'Conquest': 34,
   'Outback Wagon AWD': 34,
   'Land Cruiser Wagon 4WD': 34,
   'Cherokee/Wagoneer 4WD': 34,
   'Camry Solara': 34,
   'Durango 4WD': 34,
   'Continental': 34,
   'Tracer': 34,
   'S4': 34,
   'Aerostar Wagon': 33,
   'Forte': 33,
   'Impreza Wagon AWD': 33,
   'Sierra 2500 4WD': 33,
   '300ZX 2x2': 33,
   'Tucson 2WD': 33,
   'Sonoma 4WD': 33,
   'Express 1500/2500 2WD': 33,
   'LeBaron GTS': 33,
   '3000 GT': 33,
   'Patriot 4WD': 33,
   'Omni': 33,
   'Achieva': 32,
   'S40 FWD': 32,
   'LS': 32,
   'Summit': 32,
   'PT Cruiser': 32,
   'Terrain FWD': 32,
   'Cavalier Convertible': 32,
   'Sundance': 32,
   'MPV': 32,
   'Touareg': 32,
   'Cruze': 32,
   '600': 32,
   'Aerostar Van': 32,
   'Colorado Crew Cab 2WD': 31,
   'Pickup 2500 4WD': 31,
   'Compass 4WD': 31,
   'Beetle': 31,
   '9-5': 31,
   'XC90 AWD': 31,
   'Highlander 2WD': 31,
   'Suburban 1500 4WD': 31,
   '9-5 Wagon': 31,
   'Canyon Crew Cab 2WD': 31,
   'B350 Wagon 2WD': 30,
   'Impreza Wagon/Outback SPT AWD': 30,
   '525i': 30,
   'Neon': 30,
   'Cooper S Convertible': 30,
   'CR-V 4WD': 30,
   'Cooper Convertible': 30,
   'Jetta Wagon': 30,
   'Stealth': 30,
   'Sunbird Convertible': 30,
   'Metro': 30,
   'Town Car': 30,
   'I-Mark': 29,
   'Cutlass Cruiser': 29,
   'F150 Pickup 4WD FFV': 29,
   '900 Convertible': 29,
   'Outlander 4WD': 29,
   'Terrain AWD': 29,
   'Civic CRX': 29,
   'Rodeo 4WD': 29,
   'Wagon': 29,
   '505 Station Wagon': 29,
   'F150 Pickup 2WD FFV': 29,
   'Edge FWD': 29,
   'G6': 29,
   'Acclaim': 29,
   'Wagon 4WD': 29,
   '323': 28,
   'S60 AWD': 28,
   'Jetta SportWagen': 28,
   'Aries Wagon': 28,
   'Legend': 28,
   'Skyhawk Wagon': 28,
   'NSX': 28,
   'Park Avenue': 28,
   'Pickup 4WD': 28,
   'Sonic': 28,
   'Z3 Roadster': 28,
   'Grand Vitara': 28,
   'C70 Convertible': 28,
   'Grand Marquis': 28,
   'Reliant Wagon': 28,
   'Sonic 5': 28,
   '300': 28,
   'Stage 3 Mustang': 28,
   'AD100/AD150 Ramcharger 2WD': 28,
   'Caprice': 28,
   'Range Rover Sport': 28,
   'TSX': 27,
   '535i': 27,
   'Sorento 2WD': 27,
   'Yukon XL 1500 2WD': 27,
   'Seville': 27,
   'Forte Koup': 27,
   'Focus Station Wagon': 27,
   'Vue FWD': 27,
   'Avalon': 27,
   'Excel': 27,
   'Firebird/Formula': 27,
   '530i': 27,
   'Santa Fe 4WD': 27,
   'Horizon': 27,
   'Mountaineer 2WD': 27,
   'Comanche Pickup 4WD': 26,
   'Cayman': 26,
   'Jimmy 4WD': 26,
   'HHR FWD': 26,
   'Frontier V6 4WD': 26,
   '200': 26,
   'Veloster': 26,
   'XT': 26,
   'Cayman S': 26,
   'Vibe': 26,
   'Edge AWD': 26,
   'B2200/B2600i': 26,
   'Caliber': 26,
   '3 Series': 26,
   'M3 Convertible': 26,
   'TT Coupe quattro': 26,
   'Quest': 26,
   'Comanche Pickup 2WD': 26,
   'ATS': 26,
   'Jimmy 2WD': 26,
   '1500 4WD': 25,
   'Town and Country/Voyager/Grand Voy. 2WD': 25,
   'A3': 25,
   'R8': 25,
   'K20 Pickup 4WD': 25,
   'Frontier 4WD': 25,
   '300 AWD': 25,
   '200SX': 25,
   'Celica Convertible': 25,
   'Charger AWD': 25,
   'Compass 2WD': 25,
   'Lemans': 25,
   'M5': 25,
   'Precis': 25,
   '328i': 25,
   '740/760': 25,
   'Insight': 25,
   'K25 Pickup 4WD': 25,
   'Patriot 2WD': 25,
   'Riviera': 25,
   'Caravan/Ram Van 2WD': 24,
   'Expedition 2WD': 24,
   'Cayenne': 24,
   'Cherokee/Wagoneer': 24,
   'Turismo': 24,
   'Mystique': 24,
   'B2300/B3000/B4000 Pickup 2WD': 24,
   'Outlander 2WD': 24,
   'Contour': 24,
   'XT 4WD': 24,
   '540i': 24,
   'Q7': 24,
   'Frontier V6 2WD': 24,
   'Yaris': 24,
   'Eldorado': 24,
   'Genesis Coupe': 24,
   'V50 FWD': 24,
   'Lynx Wagon': 24,
   '1500 2WD': 24,
   'T100 2WD': 24,
   'Mountaineer 4WD': 24,
   'El Camino Pickup 2WD': 24,
   'Sequoia 4WD': 24,
   'Rabbit': 24,
   'A6': 24,
   'Caballero Pickup 2WD': 24,
   'Cabriolet': 24,
   'XF': 24,
   'tC': 24,
   'Sienna 2WD': 24,
   'Caravan/Grand Caravan/Ram Van 2WD': 24,
   'A5 quattro': 24,
   'Tredia': 23,
   'Sorento 4WD': 23,
   'Sentra Wagon': 23,
   'XJ': 23,
   '911 Turbo': 23,
   'Dart': 23,
   'G20': 23,
   'TT Roadster quattro': 23,
   '5 Series': 23,
   'Cooper': 23,
   'Sequoia 2WD': 23,
   'Dakota Cab Chassis 2WD': 23,
   'Titan 2WD': 23,
   'XC60 AWD': 23,
   'Titan 4WD': 23,
   'Tracer Wagon': 22,
   'Cimarron': 22,
   'Accent/Brio': 22,
   'Fiero': 22,
   '911 Carrera 4 Cabriolet': 22,
   'Civic Hybrid': 22,
   '3 4-Door': 22,
   'Sephia': 22,
   'Stratus 4 Door': 22,
   'xB': 22,
   'Stanza': 22,
   'Xterra 4WD': 22,
   '9-3 SportCombi': 22,
   'Lanos': 22,
   '911 Carrera 4S': 22,
   '911 Carrera Cabriolet': 22,
   'Xterra 2WD': 22,
   'Passport 2WD': 22,
   'LTD Crown Victoria': 22,
   'Concorde': 22,
   'Blazer 2WD': 22,
   'Escape 4WD': 22,
   'Camry Wagon': 22,
   'Blazer 4WD': 22,
   'Firefly': 22,
   '3 5-Door': 22,
   'Expedition 4WD': 22,
   'Lancer Evolution': 22,
   '500': 22,
   '740 Wagon': 21,
   'Cobalt': 21,
   'Grand Vitara 4WD': 21,
   'M6 Convertible': 21,
   'XC70 AWD': 21,
   'Q5': 21,
   'Cordia': 21,
   'CR-V 2WD': 21,
   'V8 Vantage': 21,
   'Flex AWD': 21,
   'Outback AWD': 21,
   'SRX AWD': 21,
   '550i': 21,
   '911 Carrera 4': 21,
   'Cutlass Calais': 21,
   'Alliance/Encore': 21,
   '1000': 21,
   'Quantum Wagon': 21,
   'Beetle Convertible': 21,
   'Altima Coupe': 21,
   'New Beetle Convertible': 21,
   'Civic Wagon': 21,
   'S5': 21,
   'Esteem': 21,
   'Lumina': 21,
   'X5': 21,
   'Bronco II 4WD': 21,
   '528i': 21,
   'MX-6': 20,
   'CC': 20,
   'Shadow Convertible': 20,
   '626/MX-6': 20,
   'Tribute 4WD': 20,
   'SC 300/SC 400': 20,
   'MX-3': 20,
   'Accord Coupe': 20,
   'Cooper S': 20,
   '370Z': 20,
   'A8': 20,
   'Highlander 4WD': 20,
   'Sprint': 20,
   'B4000 4WD': 20,
   'Armada 2WD': 20,
   '740': 20,
   'Ion': 20,
   'S600': 20,
   'Accord Wagon': 20,
   'Taurus AWD': 20,
   '911 Carrera S': 20,
   'Tucson 4WD': 20,
   '911 Carrera S Cabriolet': 20,
   '911 Carrera 4S Cabriolet': 20,
   'Starion': 20,
   'Firenza Cruiser Wagon': 20,
   'Outlander Sport 2WD': 20,
   'Sundance/Duster': 20,
   'Sentra/200SX': 20,
   'G35': 20,
   'Armada 4WD': 20,
   'Lancer Sportback': 20,
   '330i': 20,
   'Limousine': 19,
   'Canyon Crew Cab 4WD': 19,
   'Quantum': 19,
   'SLK350': 19,
   'Colorado Crew Cab 4WD': 19,
   'B2300/B3000/B4000 Pickup 4WD': 19,
   'Vue AWD': 19,
   'G30 Van 2WD': 19,
   'Sebring 4 Door': 19,
   'Taurus FWD': 19,
   'Compass FWD': 19,
   'Yukon K1500 4WD': 19,
   'S10 Cab Chassis 2WD': 19,
   '240SX': 19,
   'Silhouette 2WD': 19,
   'Evora': 19,
   'Avalanche 1500 2WD': 19,
   '370Z Roadster': 19,
   'EXP': 19,
   'CTS AWD': 19,
   'Continental GT': 19,
   'Diplomat': 18,
   'Caravelle': 18,
   'Golf/GTI': 18,
   'HHR Panel FWD': 18,
   'Colt Vista 4WD': 18,
   'Prius': 18,
   'Impala/Caprice': 18,
   'Focus FWD FFV': 18,
   'B2300 2WD': 18,
   'Corolla Wagon': 18,
   'Hombre Pickup 2WD': 18,
   'Cutlass Cruiser Wagon': 18,
   'Expo.LRV': 18,
   '335i': 18,
   'Cooper S Countryman All4': 18,
   'Cooper Countryman': 18,
   'C30 FWD': 18,
   'Escalade 2WD': 18,
   'MKZ AWD': 18,
   'Spider': 18,
   'MKZ FWD': 18,
   'Sidekick Hardtop 4WD': 18,
   'Town and Country 2WD': 18,
   'Suburban V1500 4WD': 18,
   'Parisienne': 18,
   'Comanche 2WD': 18,
   'X-Type': 18,
   'Gran Fury': 18,
   'Amigo 2WD': 18,
   'Solstice': 18,
   'Element 4WD': 18,
   'CTS Wagon': 18,
   'Mustang Convertible': 18,
   'Savana 1500/2500 2WD (Passenger)': 18,
   'Yukon XL 1500 4WD': 18,
   'Passport 4WD': 18,
   'Passat 4motion': 18,
   '600 Convertible': 18,
   'Element 2WD': 17,
   '300/SRT-8': 17,
   'Expo': 17,
   'Power Ram 50 Pickup 4WD': 17,
   'Cherokee FWD': 17,
   '740i': 17,
   '944': 17,
   'E350': 17,
   'Astro AWD (cargo)': 17,
   'SRX 2WD': 17,
   'XC90 FWD': 17,
   'Passat Wagon 4motion': 17,
   'Z4 Roadster': 17,
   'Q45': 17,
   'Sedan': 17,
   'Patriot FWD': 17,
   'Eos': 17,
   'Pulsar NX': 17,
   'S80 AWD': 17,
   'Alero': 17,
   'Sedona': 17,
   'Astro AWD (passenger)': 17,
   'R8 Spyder': 17,
   'Comanche 4WD': 17,
   'Safari AWD (cargo)': 17,
   '911 Turbo Cabriolet': 17,
   '5': 17,
   'Caprice Wagon': 17,
   'Gallardo Coupe': 16,
   'TrailBlazer 2WD': 16,
   'Azera': 16,
   'Coupe GT': 16,
   'Citation II': 16,
   'C300 4matic': 16,
   'S15 Cab Chassis 2WD': 16,
   'Millenia': 16,
   'Yukon K1500 XL 4WD': 16,
   'CL600': 16,
   'Cabrio': 16,
   'Dynasty': 16,
   'Caravan C/V/Grand Caravan 2WD': 16,
   'LTD': 16,
   'Marquis': 16,
   'E350 4matic': 16,
   'Laser/Daytona': 16,
   'Grand Caravan': 16,
   'Lumina/Monte Carlo': 16,
   'Electra/Park Avenue': 16,
   'SKY': 16,
   '1-Ton Truck 2WD': 16,
   'Cayenne S': 16,
   'Cab/Chassis 2WD': 16,
   'Acadia AWD': 16,
   'Forte 5': 16,
   'Raider Pickup 2WD': 16,
   'Tiguan': 16,
   'Town and Country': 16,
   'Justy': 16,
   'John Cooper Works Convertible': 16,
   'Navigator 2WD': 16,
   'Acadia FWD': 16,
   'Montero Sport 2WD': 16,
   'Explorer FWD': 16,
   'PT Cruiser Convertible': 16,
   'Carrera 2 Cabriolet': 16,
   'RX-8': 16,
   'FJ Cruiser 4WD': 16,
   'Challenger SRT8': 16,
   'Diamante': 16,
   'Carrera 2 Coupe': 16,
   'Legacy': 16,
   'Montero Sport 4WD': 16,
   'Eclipse Convertible': 16,
   'Gallardo Spyder': 16,
   '318i/318is': 16,
   'B4000 2WD': 16,
   '968': 16,
   'Hatchback': 16,
   'Paseo': 16,
   'Explorer AWD': 16,
   'Cooper S Clubman': 16,
   'Cayenne Turbo': 16,
   'Storm': 16,
   'C350': 16,
   'Xterra V6 4WD': 15,
   'S70': 15,
   'MKX AWD': 15,
   'Murano AWD': 15,
   'Phantom': 15,
   '325i': 15,
   'MKS AWD': 15,
   'RSX': 15,
   'Xterra V6 2WD': 15,
   'Pilot 4WD': 15,
   'Esteem Wagon': 15,
   'V70 AWD': 15,
   'Liberty/Cherokee 4WD': 15,
   'ATS AWD': 15,
   'MDX 4WD': 15,
   'MKX FWD': 15,
   '750i': 15,
   'Bravada AWD': 15,
   '850': 15,
   'SL600': 15,
   'Suburban C10 2WD': 15,
   '850 Wagon': 15,
   'LaCrosse': 15,
   'Cooper S Countryman': 15,
   'Yukon C1500 XL 2WD': 15,
   'Crown Victoria': 15,
   'WRX': 15,
   'Chevette': 15,
   'Yukon C1500 2WD': 15,
   '405 Sedan': 15,
   'Windstar FWD Wagon': 15,
   'Commander 2WD': 15,
   'Suburban K10 4WD': 15,
   'XK': 15,
   'Legacy Wagon': 15,
   'Cressida': 15,
   'Commander 4WD': 15,
   'Wrangler Unlimited 4WD': 14,
   'S10 Blazer 4WD': 14,
   '323/323 Protege': 14,
   'Sedan/3 Door': 14,
   'Wrangler/TJ 4WD': 14,
   'Venza': 14,
   'Town and Country Wagon': 14,
   'Tribute 2WD': 14,
   'xD': 14,
   'Civic CRX HF': 14,
   '240 DL/GL/Turbo': 14,
   'Cooper Clubman': 14,
   'Explorer Sport Trac 4WD': 14,
   'ES 300': 14,
   'Spectra': 14,
   '612 Scaglietti': 14,
   'Lacrosse/Allure': 14,
   'Avalanche 1500 4WD': 14,
   'Explorer Sport Trac 2WD': 14,
   'Tracker 2WD Convertible': 14,
   'C15 Suburban 2WD': 14,
   'Breeze': 14,
   'Power Ram 50 4WD': 14,
   'Lucerne': 14,
   'Discovery': 14,
   'Aveo 5': 14,
   'Corolla Sport': 14,
   'Raider Pickup 4WD': 14,
   'Quattroporte': 14,
   '240 DL/GL/Turbo Wagon': 14,
   'Spark': 14,
   'Tracker 4WD Convertible': 14,
   'Tracker 4WD Hardtop': 14,
   'S4 Avant': 14,
   'Tracker 2WD Hardtop': 14,
   'K15 Jimmy 4WD': 14,
   'Fiesta FWD': 14,
   'Savana 1500 AWD (Passenger)': 14,
   'Toronado': 14,
   'CJ7': 14,
   'Cirrus': 14,
   'Aveo': 14,
   'Testarossa': 14,
   '330ci': 14,
   'New Yorker': 14,
   'XK Convertible': 14,
   'Magnum': 14,
   '750il': 14,
   'Firenza Cruiser': 14,
   'Milan': 14,
   'K15 Suburban 4WD': 14,
   'SX4 AWD': 14,
   'Crown Victoria Police': 14,
   '330ci Convertible': 14,
   '5000S Wagon': 14,
   'Sedan/3 Door 4WD': 14,
   'Civic Del Sol': 14,
   'Z4 sDrive35i': 14,
   'Nubira': 14,
   'Elantra GT': 14,
   'Voyager 2WD': 14,
   'Brat 4WD': 14,
   'Fusion AWD': 14,
   'SX4': 14,
   'Juke': 14,
   'Vitara 2Door 4WD': 14,
   'Verano': 14,
   'M6 Gran Coupe': 14,
   '2': 14,
   'Vision': 14,
   'Baja AWD': 14,
   'Carrera 4 Cabriolet': 14,
   '190E 2.6': 13,
   '9-3': 13,
   'S80 FWD': 13,
   'XTS AWD': 13,
   'Escape AWD': 13,
   'R15 Pickup 2WD': 13,
   'Camry Solara Convertible': 13,
   'SL500': 13,
   'GranTurismo': 13,
   'Festiva': 13,
   'Highlander Hybrid 4WD': 13,
   'MPV 4x4': 13,
   'Tracker Convertible 4WD': 13,
   '325i Convertible': 13,
   'Scoupe': 13,
   'Amigo 4WD': 13,
   'A3 quattro': 13,
   'Charade': 13,
   'Envoy 2WD': 13,
   'Trans Sport 2WD': 13,
   'Taurus SHO': 13,
   'Liberty/Cherokee 2WD': 13,
   'MKC AWD': 13,
   '330xi': 13,
   'DB9': 13,
   'Murano FWD': 13,
   'Grand Wagoneer 4WD': 13,
   'Sorento FWD': 13,
   'Express 2500 2WD Passenger': 13,
   'XJ6': 13,
   '325ci': 13,
   'Civic CNG': 13,
   'S320': 13,
   '740/760 Wagon': 13,
   '325ci Convertible': 13,
   'Sorento AWD': 13,
   'ML350 4matic': 13,
   '164': 13,
   '350z': 13,
   'Envoy 4WD': 13,
   'Traverse FWD': 13,
   'DeVille': 13,
   'R10 Pickup 2WD': 13,
   'Tracker Convertible 2WD': 13,
   'Echo': 12,
   'R1500 Suburban 2WD': 12,
   'H3 4WD': 12,
   '325xi': 12,
   'Ram 2500 Pickup 4WD': 12,
   'Q70': 12,
   'XJS Convertible': 12,
   'TL': 12,
   'Ram 2500 Pickup 2WD': 12,
   'V50 AWD': 12,
   '325xi Sport Wagon': 12,
   'Savana 1500 2WD (cargo)': 12,
   'Grand Vitara XL7 4WD': 12,
   'Carrera 4 S Coupe': 12,
   'Carrera 4 S Cabriolet': 12,
   'CLS550': 12,
   'Mariner 4WD': 12,
   'S4 Cabriolet': 12,
   '328ci': 12,
   'GS 300/GS 430': 12,
   'Equator 2WD': 12,
   'V10 (K10) Pickup 4WD': 12,
   'G30 Sport Van 2WD': 12,
   '240 DL/240 GL Wagon': 12,
   'Outlander Sport 4WD': 12,
   '911 Targa 4S': 12,
   'Grand Vitara XL7': 12,
   'Saloon/Vantage/Volante': 12,
   'Sedan 4WD': 12,
   'Q70 AWD': 12,
   'S6': 12,
   'Pilot 2WD': 12,
   '318ti': 12,
   'Mark VII': 12,
   '350z Roadster': 12,
   '335i xDrive': 12,
   '325i/325is': 12,
   '320i': 12,
   'Phantom EWB': 12,
   'Savana 3500 2WD (Passenger)': 12,
   'Cherokee Trailhawk 4WD': 12,
   '650ci Convertible': 12,
   'Nitro 4WD': 12,
   'Leganza': 12,
   '650ci': 12,
   'Colt Wagon': 12,
   'F-Type Convertible': 12,
   'Previa': 12,
   'Aerio': 12,
   '335ci Convertible': 12,
   'SX4 Sedan': 12,
   'XJR': 12,
   '6 Series': 12,
   'STS': 12,
   'Civic HX': 12,
   'C300': 12,
   '929': 12,
   'Yukon XL 1500 AWD': 12,
   'Tahoe K1500 4WD': 12,
   'CJ7 4WD': 12,
   'Escalade AWD': 12,
   '911 Targa 4': 12,
   '328ci Convertible': 12,
   'V15 Pickup 4WD': 12,
   'E150 Club Wagon 2WD': 12,
   'Bronco II 2WD': 12,
   'Rondo': 12,
   'Enclave FWD': 12,
   'Vanagon/Camper 2WD': 12,
   'Enclave AWD': 12,
   'Nitro 2WD': 12,
   '240 Wagon': 12,
   'G35 Rally 2WD': 12,
   'Savana 1500/2500 2WD (passenger)': 12,
   'Scrambler Pickup 4WD': 12,
   'TrailBlazer 4WD': 12,
   'S40 AWD': 12,
   'Sunbird Wagon': 12,
   'Express 3500 2WD Passenger': 12,
   'C230 Kompressor': 12,
   'Savana 15/25 2WD Conversion (cargo)': 12,
   'Cube': 12,
   'CX-9 2WD': 12,
   'Stylus': 12,
   '100': 12,
   'fortwo coupe': 12,
   '240': 12,
   'A6 Avant quattro': 12,
   'Viper Convertible': 12,
   '128i': 12,
   '135i': 12,
   '135i Convertible': 12,
   'RX 350 AWD': 12,
   'M3 Coupe': 12,
   'ES 350': 12,
   'B1500/B2500 Van 2WD': 12,
   'Fox': 12,
   '300E': 12,
   '9-7X AWD': 12,
   'CX-5 2WD': 12,
   'Corolla LE Eco': 12,
   'MKT AWD': 12,
   'G37 Coupe': 12,
   'Savana 2500 2WD (Passenger)': 12,
   'BRZ': 12,
   'Cherokee 4WD Active Drive II': 12,
   'Escort ZX2': 12,
   'JCW Countryman All4': 12,
   'Scrambler': 12,
   'CX-9 4WD': 12,
   'Arnage': 12,
   'Coachbuilder Wagon': 12,
   'CR-Z': 12,
   'Van 15/25 2WD Conversion': 11,
   'DTS': 11,
   'LS 400': 11,
   'LS 460': 11,
   'Tahoe C1500 2WD': 11,
   'Ninety-Eight': 11,
   'Esprit Turbo': 11,
   'GS 350 AWD': 11,
   'GS 350': 11,
   'X6 xDrive35i': 11,
   'LS 460 L': 11,
   'Cayenne GTS': 11,
   'X6 xDrive50i': 11,
   'Genesis': 11,
   'Flex FWD': 11,
   '500 Abarth': 11,
   'Civic 4Dr': 11,
   'Rogue AWD': 11,
   'Rogue FWD': 11,
   'Civic 2Dr': 11,
   'F-Type Coupe': 11,
   'Escape 2WD': 11,
   'Suburban K1500 4WD': 11,
   'Previa All-Trac': 11,
   'Safari AWD (passenger)': 11,
   'Traverse AWD': 11,
   '911 GT3': 11,
   '325i Sport Wagon': 11,
   'Fuego': 11,
   'Rally G15/25 2WD': 11,
   'S550': 11,
   'STS AWD': 11,
   'Silverado 15 Hybrid 4WD': 11,
   'Eagle SX/4 4WD': 11,
   'Tercel Wagon 4WD': 11,
   '300SL': 11,
   'Truck Cab Chassis 2WD': 11,
   'Elise/Exige': 11,
   'Sierra 15 Hybrid 4WD': 11,
   'E350 Coupe': 11,
   'Caravan/Grand Caravan AWD': 11,
   '2000 Sunbird': 11,
   'Eagle 4WD': 11,
   'Azure': 11,
   'Sierra 15 Hybrid 2WD': 11,
   'Windstar FWD Van': 11,
   'Silverado 15 Hybrid 2WD': 11,
   'B1500/B2500 Wagon 2WD': 11,
   'Continental GTC': 11,
   'Space Wagon': 11,
   'Aurora': 11,
   'TL 4WD': 11,
   'S500': 11,
   'GS 450h': 11,
   'S550 4matic': 11,
   'XF AWD': 11,
   'SL550': 11,
   'XC60 FWD': 11,
   'RDX 4WD': 10,
   '456 MGT/MGA': 10,
   'LX 570': 10,
   'ML350': 10,
   …}},
 'mpgData': {'Unique Values': 2,
  'Most Frequent': 'N',
  'Frequency': {'N': 27367, 'Y': 12714}},
 'phevBlended': {'Unique Values': 2,
  'Most Frequent': np.False_,
  'Frequency': {False: 40005, True: 76}},
 'trany': {'Unique Values': 37,
  'Most Frequent': 'Automatic 4-spd',
  'Frequency': {'Automatic 4-spd': 11045,
   'Manual 5-spd': 8351,
   'Automatic 3-spd': 3151,
   'Automatic (S6)': 2984,
   'Manual 6-spd': 2671,
   'Automatic 5-spd': 2198,
   'Automatic 6-spd': 1579,
   'Manual 4-spd': 1483,
   'Automatic (S8)': 1421,
   'Automatic (S5)': 830,
   'Automatic (variable gear ratios)': 766,
   'Automatic 7-spd': 708,
   'Automatic (AM-S7)': 374,
   'Automatic 8-spd': 347,
   'Automatic (S7)': 307,
   'Automatic (S4)': 233,
   'Automatic (AM7)': 216,
   'Automatic 9-spd': 208,
   'Automatic (AV-S6)': 189,
   'Automatic (A1)': 163,
   'Automatic (AM6)': 136,
   'Automatic (AV-S7)': 126,
   'Manual 7-spd': 113,
   'Automatic (AM-S6)': 113,
   'Manual 3-spd': 77,
   'Automatic (S10)': 69,
   'Automatic (S9)': 65,
   'Automatic (AV-S8)': 43,
   'Automatic (AM-S8)': 40,
   'Manual 4-spd Doubled': 17,
   'Automatic 10-spd': 16,
   'Automatic (AM5)': 14,
   'Automatic (AM8)': 6,
   'Automatic (AV-S10)': 5,
   'Automatic (L3)': 2,
   'Automatic (AM-S9)': 2,
   'Automatic (L4)': 2}},
 'VClass': {'Unique Values': 34,
  'Most Frequent': 'Compact Cars',
  'Frequency': {'Compact Cars': 5751,
   'Subcompact Cars': 5036,
   'Midsize Cars': 4697,
   'Standard Pickup Trucks': 2354,
   'Sport Utility Vehicle - 4WD': 2090,
   'Large Cars': 2072,
   'Two Seaters': 2025,
   'Sport Utility Vehicle - 2WD': 1627,
   'Small Station Wagons': 1579,
   'Special Purpose Vehicles': 1455,
   'Minicompact Cars': 1367,
   'Standard Pickup Trucks 2WD': 1221,
   'Vans': 1141,
   'Standard Pickup Trucks 4WD': 1044,
   'Small Sport Utility Vehicle 4WD': 734,
   'Midsize-Large Station Wagons': 656,
   'Special Purpose Vehicle 2WD': 643,
   'Standard Sport Utility Vehicle 4WD': 577,
   'Small Pickup Trucks': 538,
   'Midsize Station Wagons': 536,
   'Small Sport Utility Vehicle 2WD': 530,
   'Small Pickup Trucks 2WD': 457,
   'Vans, Cargo Type': 438,
   'Minivan - 2WD': 356,
   'Vans, Passenger Type': 314,
   'Special Purpose Vehicle 4WD': 310,
   'Standard Sport Utility Vehicle 2WD': 237,
   'Small Pickup Trucks 4WD': 237,
   'Minivan - 4WD': 48,
   'Standard Pickup Trucks/2wd': 4,
   'Vans Passenger': 2,
   'Special Purpose Vehicles/2wd': 2,
   'Special Purpose Vehicles/4wd': 2,
   'Special Purpose Vehicle': 1}},
 'guzzler': {'Unique Values': 3,
  'Most Frequent': 'G',
  'Frequency': {'G': 1398, 'T': 964, 'S': 15}},
 'trans_dscr': {'Unique Values': 52,
  'Most Frequent': 'CLKUP',
  'Frequency': {'CLKUP': 7809,
   'SIL': 2189,
   '2MODE CLKUP': 1235,
   'Creeper': 525,
   'EMS 2MODE CLKUP': 520,
   '3MODE CLKUP': 517,
   '2MODE': 448,
   '2MODE 2LKUP': 383,
   'EMS': 252,
   '3MODE': 166,
   'CMODE': 150,
   'CMODE CLKUP': 130,
   'VMODE CLKUP': 105,
   'SIL Creeper': 72,
   '2LKUP': 63,
   'DC/FW': 53,
   'VLKUP': 52,
   'EMS CLKUP': 50,
   'EMS 2MODE': 46,
   '6MODE CLKUP': 41,
   '4MODE': 35,
   'VMODE VLKUP': 26,
   '2MODE 3LKUP': 22,
   '2MODE DC/FW': 19,
   'CMODE VLKUP': 17,
   '3LKUP': 15,
   'EMS 3MODE': 11,
   'Overdrive': 10,
   '2MODE VLKUP': 9,
   'Lockup': 9,
   'SIL EMS': 7,
   '6MODE': 6,
   '4MODE CLKUP': 6,
   'SIL 3MODE': 6,
   'SIL 2MODE CLKUP': 5,
   'EMS 2MODE CLKU': 5,
   '3MODE VLKUP': 5,
   'Elec Overdrive': 3,
   'Mech Overdrive': 3,
   'EMS 3MODE CLKUP': 3,
   'SIL 3MODE CLKUP': 2,
   'LONG RATIO': 2,
   'EMS CMODE CLKUP': 2,
   '3MODE 2LKUP': 2,
   'SIL CMODE': 2,
   '2MODE CLKUP FW': 2,
   'VMODE': 2,
   '3MODE CLKUP FW': 1,
   'SIL CLKUP': 1,
   'EMS 5MODE': 1,
   'fuel injection': 1,
   'Lockup A3': 1}},
 'tCharger': {'Unique Values': 1,
  'Most Frequent': 'T',
  'Frequency': {'T': 6302}},
 'sCharger': {'Unique Values': 1,
  'Most Frequent': 'S',
  'Frequency': {'S': 796}},
 'atvType': {'Unique Values': 8,
  'Most Frequent': 'FFV',
  'Frequency': {'FFV': 1412,
   'Diesel': 1070,
   'Hybrid': 539,
   'EV': 168,
   'Plug-in Hybrid': 107,
   'CNG': 50,
   'Bifuel (CNG)': 20,
   'Bifuel (LPG)': 8}},
 'fuelType2': {'Unique Values': 4,
  'Most Frequent': 'E85',
  'Frequency': {'E85': 1412,
   'Electricity': 107,
   'Natural Gas': 20,
   'Propane': 8}},
 'rangeA': {'Unique Values': 223,
  'Most Frequent': '290',
  'Frequency': {'290': 74,
   '270': 57,
   '280': 54,
   '310': 41,
   '277': 38,
   '260': 36,
   '307': 33,
   '300': 30,
   '320': 28,
   '280/380': 26,
   '240': 24,
   '413': 24,
   '308': 24,
   '329': 22,
   '269': 21,
   '304': 20,
   '338': 20,
   '14': 20,
   '340': 19,
   '296': 17,
   '279': 17,
   '364/476': 16,
   '295': 16,
   '273': 15,
   '280-350': 14,
   '310/390': 14,
   '290/410': 14,
   '332': 13,
   '353': 13,
   '330': 13,
   '262': 13,
   '310-340': 12,
   '310/340': 12,
   '337/451': 12,
   '338/442': 12,
   '250': 12,
   '287': 12,
   '264': 11,
   '310-420': 11,
   '230': 11,
   '298': 11,
   '310/460': 11,
   '340-400': 10,
   '16': 10,
   '346': 10,
   '314': 10,
   '318': 9,
   '20': 9,
   '253': 9,
   '383': 9,
   '238': 9,
   '360': 9,
   '370': 9,
   '120': 8,
   '350': 8,
   '340-440': 8,
   '290/340': 8,
   '344': 8,
   '331': 8,
   '294': 7,
   '338/416': 7,
   '283': 7,
   '310/440': 7,
   '274': 7,
   '310/480': 7,
   '364/448': 7,
   '278': 6,
   '284': 6,
   '286/382': 6,
   '347': 6,
   '307/420': 6,
   '366': 6,
   '305': 6,
   '313/465': 6,
   '382': 6,
   '11': 5,
   '312': 5,
   '15': 5,
   '292': 5,
   '333': 5,
   '12': 5,
   '285': 5,
   '302': 4,
   '266': 4,
   '314/434': 4,
   '339/469': 4,
   '337/467': 4,
   '361/501': 4,
   '308/366': 4,
   '337/406': 4,
   '325': 4,
   '362': 4,
   '309/313': 4,
   '271': 4,
   '270/240/340': 4,
   '250/220/320': 4,
   '407': 4,
   '245': 4,
   '241': 4,
   '310-540': 4,
   '282': 4,
   '445': 4,
   '338/468': 4,
   '335/445': 4,
   '317': 4,
   '339/441': 4,
   '361/479': 4,
   '330/450': 4,
   '380': 4,
   '310-410': 4,
   '290/418': 4,
   '357': 4,
   '310/330': 4,
   '369': 4,
   '170': 4,
   '409': 4,
   '336': 4,
   '337': 4,
   '316': 3,
   '33': 3,
   '254': 3,
   '37': 3,
   '306': 3,
   '234': 3,
   '38': 3,
   '72': 3,
   '365': 3,
   '19': 3,
   '17': 3,
   '18': 3,
   '335/378': 3,
   '280/370': 3,
   '29': 3,
   '97': 3,
   '311/351': 3,
   '410': 3,
   '310/540': 3,
   '265': 3,
   '276': 3,
   '53': 3,
   '21': 3,
   '358': 2,
   '444': 2,
   '312/432': 2,
   '10': 2,
   '358/405': 2,
   '382/432': 2,
   '364/504': 2,
   '31': 2,
   '69': 2,
   '9': 2,
   '240/280': 2,
   '281': 2,
   '25': 2,
   '264/380': 2,
   '22': 2,
   '27': 2,
   '400': 2,
   '333/337': 2,
   '357/362': 2,
   '330-450': 2,
   '368': 2,
   '220': 2,
   '260/320': 2,
   '214': 2,
   '248': 2,
   '236': 2,
   '311': 2,
   '288/351': 2,
   '272': 2,
   '361': 2,
   '35': 2,
   '310/420': 2,
   '217': 2,
   '350-370': 2,
   '130': 2,
   '339': 2,
   '291': 2,
   '267': 2,
   '354': 2,
   '286/352': 2,
   '300-420': 2,
   '310/410': 2,
   '345': 2,
   '336/402': 2,
   '308/368': 2,
   '340-350': 1,
   '303': 1,
   '220/260': 1,
   '250/370': 1,
   '362/537': 1,
   '337/501': 1,
   '240-420': 1,
   '415': 1,
   '230/350': 1,
   '310/370': 1,
   '310/320': 1,
   '48': 1,
   '26': 1,
   '219': 1,
   '40': 1,
   '36': 1,
   '328': 1,
   '286': 1,
   '220/270/260': 1,
   '13': 1,
   '351': 1,
   '240/290/280': 1,
   '301': 1,
   '230/270': 1,
   '334': 1,
   '140': 1,
   '258': 1,
   '256': 1,
   '341': 1,
   '240/290/290': 1,
   '210': 1,
   '110': 1,
   '119': 1,
   '180': 1,
   '230/270/270': 1,
   '289/430': 1,
   '28': 1}},
 'evMotor': {'Unique Values': 140,
  'Most Frequent': '288V Ni-MH',
  'Frequency': {'288V Ni-MH': 122,
   '245V Ni-MH': 48,
   '144V Li-Ion': 28,
   '270V Li-Ion': 26,
   '330V Ni-MH': 26,
   '202V Ni-MH': 23,
   '346V Li-Ion': 21,
   '101V Ni-MH': 17,
   '280V Li-Ion': 17,
   '86V Li-Ion': 14,
   '115V Li-Ion': 13,
   '70 kW DC Brushless': 12,
   '80 kW DCPM': 12,
   '126V Li-Ion': 11,
   '374V Li-Ion': 11,
   '240V Li-Ion': 10,
   '125 kW AC Induction': 10,
   '259V Li-Ion': 9,
   '275V Ni-MH': 8,
   '55 kW DCPM': 8,
   '36V Ni-MH': 8,
   '144V Ni-MH': 7,
   '68 kW DCPM': 7,
   '85 kW DCPM': 7,
   '83kW DCPM': 6,
   '193 (front) 375 (rear) (100 kW-hr battery pack)': 6,
   '107 kW AC PMSM': 6,
   '34 and 65kW 3-phase Sync.': 6,
   '193 (front) 193 (rear) (75 kW-hr battery pack)': 6,
   '260V Li-Ion': 5,
   '49 kW DCPM': 5,
   '266V Li-Ion': 4,
   '60 KW AC Synchronous': 4,
   '68 kW': 4,
   '220V Li-Ion': 4,
   '60 kW DCPM': 4,
   '193 (front) 193 (rear) (90 kW-hr battery pack)': 4,
   '81 kW AC PMSM': 4,
   '193 (front) 375 (rear) (90 kW-hr battery pack)': 4,
   '202V Li-Ion': 4,
   '82 kW AC Induction': 4,
   '111 kW': 4,
   '207V Li-Ion': 4,
   '132 kW AC Induction': 4,
   '48V Li-Ion': 4,
   '193 (front) 193 (rear) (60 kW-hr battery pack)': 4,
   '50 KW DC': 4,
   '158V Ni-MH': 4,
   '193 (front) 193 (rear) (100 kW-hr battery pack)': 4,
   '125kW': 3,
   '96kW': 3,
   '311V Li-Ion': 3,
   '75 kW AC PM/B': 3,
   '75 kW AC PMSM': 3,
   '96kW AC Induction': 3,
   '300V Li-Ion': 3,
   '67 KW AC': 3,
   '83 kW AC Induction': 3,
   '88 kW AC PMSM': 3,
   '50 kW IPMSM': 3,
   '48 and 87 kW 3-Phase AC': 3,
   '115 kW AC Induction': 3,
   '18 kW': 3,
   '100 kW AC PMSM': 2,
   '85 kW AC PMSM': 2,
   '89 kW AC Induction': 2,
   '55 and 174kW 3-Phase AC': 2,
   '34 and 65kW DCPM': 2,
   '150 kW ACPM': 2,
   '285 kW AC Induction (90 kW-hr battery pack)': 2,
   '16 and 37 kW AC Induction': 2,
   '285 kW AC Induction (60 kW-hr battery pack)': 2,
   '50kW IPMSM': 2,
   '285 kW AC Induction (75 kW-hr battery pack)': 2,
   '82 kW ACIPM': 2,
   '30 kW DCPM': 2,
   '56kW AC Induction': 2,
   '67 KW AC  Induction': 2,
   '85 kW AC Induction': 2,
   '312V Ni-MH': 2,
   '65 kW AC Induction': 2,
   '52 kW AC Induction': 2,
   '100 kW DCPM': 2,
   '300V Ni-MH': 2,
   '222 kW AC PMSM': 2,
   '92 kW DCPM': 2,
   '225 kW AC Induction': 2,
   '32kW IPMSM': 2,
   '67 KW AC Induction': 2,
   '270 kW AC Induction (85 kW-hr battery pack)': 2,
   '225 kW AC Induction (60 kW-hr battery pack)': 2,
   '120 kW AC PMSM': 2,
   '102kW AC Induction': 2,
   'Two 444V Li-Ion': 1,
   '147 and 188 kW AC 3-Phase': 1,
   '59 and 61kW 3-Phase AC': 1,
   '110 kW DCPM': 1,
   '125kW AC Induction': 1,
   '270 kW AC Induction (75 kW-hr battery pack)': 1,
   '211 kW AC 3-Phase': 1,
   '135kW AC PMSM': 1,
   '192 kW AC 3-Phase': 1,
   '59 and 61 kW 3-Phase AC': 1,
   '150 and 150 kw DCPM': 1,
   '62 KW AC Induction': 1,
   '18 kW AC Induction': 1,
   '65kW': 1,
   '260 kW AC Induction': 1,
   '95 kW and 116 kW DC Brushless': 1,
   '70 kW': 1,
   '126 kW': 1,
   '104 kW AC Induction': 1,
   '124 kW': 1,
   '270 kW AC Induction': 1,
   '2 @ 150 kw (300 kw)': 1,
   '193 (front) 375 (rear) (85 kW-hr battery pack)': 1,
   '107 kW AC Induction': 1,
   '150 kW': 1,
   '49 kW DC Brushless': 1,
   '49kW DC Brushless': 1,
   '24 KW AC Synchronous': 1,
   '27 KW AC Induction': 1,
   '104 kW ACPM': 1,
   '111 kW 3-Phase AC': 1,
   '126 kW 3-Phase AC': 1,
   '132kW': 1,
   '132 kW AC Synchronous': 1,
   'Two 480V Li-Ion': 1,
   '180 and 350 kW AC Induction (85 kW-hr battery pack)': 1,
   '164 (front) 350 (rear) (85 kW-hr battery pack)': 1,
   '140 (front) 140 (rear) (85 kW-hr battery pack)': 1,
   '140 (front) 140 (rear) (70 kW-hr battery pack)': 1,
   '140 (front) 140 (rear) (90 kW-hr battery pack)': 1,
   '164 (front) 350 (rear) (90 kW-hr battery pack)': 1,
   '105 kW ACPM': 1,
   '285 kW AC Induction (70 kW-hr battery pack)': 1,
   '285 kW AC Induction (85 kW-hr battery pack)': 1,
   '193 (front) 193 (rear) (70 kW-hr battery pack)': 1,
   '193 (front) 193 (rear) (85 kW-hr battery pack)': 1,
   '147 and 211 kW AC 3-Phase': 1}},
 'mfrCode': {'Unique Values': 47,
  'Most Frequent': 'GMX',
  'Frequency': {'GMX': 1366,
   'BMX': 1072,
   'FMX': 792,
   'CRX': 703,
   'TYX': 662,
   'MBX': 593,
   'NSX': 546,
   'PRX': 486,
   'VGA': 383,
   'HNX': 358,
   'HYX': 338,
   'KMX': 303,
   'JLX': 241,
   'FJX': 172,
   'TKX': 167,
   'VVX': 163,
   'ADX': 138,
   'VWX': 132,
   'MTX': 124,
   'FEX': 99,
   'TSL': 58,
   'ASX': 56,
   'RRG': 48,
   'MAX': 46,
   'SKX': 33,
   'JCX': 27,
   'RII': 22,
   'SAX': 21,
   'LTX': 20,
   'BEX': 16,
   'NLX': 15,
   'LRX': 14,
   'MLN': 13,
   'BYD': 6,
   'DSX': 6,
   'MBV': 4,
   'PGN': 3,
   'BGT': 3,
   'TVP': 3,
   'CDA': 2,
   'QTM': 2,
   'AZD': 2,
   'FSK': 1,
   'SKR': 1,
   'FTG': 1,
   'KAL': 1,
   'KGG': 1}},
 'c240Dscr': {'Unique Values': 5,
  'Most Frequent': 'standard charger',
  'Frequency': {'standard charger': 54,
   '3.6 kW charger': 4,
   'single charger': 3,
   '7.2 kW charger': 2,
   '6.6 kW charger': 2}},
 'c240bDscr': {'Unique Values': 4,
  'Most Frequent': '80 amp dual charger',
  'Frequency': {'80 amp dual charger': 54,
   '6.6 kW charger': 4,
   'dual charger': 3,
   '3.6 kW charger': 2}},
 'createdOn': {'Unique Values': 234,
  'Most Frequent': 'Tue Jan 01 00:00:00 EST 2013',
  'Frequency': {'Tue Jan 01 00:00:00 EST 2013': 34217,
   'Fri Jun 06 00:00:00 EDT 2014': 100,
   'Thu Jul 07 00:00:00 EDT 2016': 100,
   'Thu Jul 31 00:00:00 EDT 2014': 95,
   'Wed Jun 13 00:00:00 EDT 2018': 93,
   'Tue Aug 02 00:00:00 EDT 2016': 86,
   'Thu Aug 03 00:00:00 EDT 2017': 86,
   'Wed Apr 04 00:00:00 EDT 2018': 84,
   'Thu Jul 16 00:00:00 EDT 2015': 80,
   'Thu Jul 02 00:00:00 EDT 2015': 77,
   'Thu Aug 10 00:00:00 EDT 2017': 77,
   'Wed Jul 26 00:00:00 EDT 2017': 76,
   'Thu Aug 06 00:00:00 EDT 2015': 75,
   'Mon Jul 27 00:00:00 EDT 2015': 74,
   'Mon Sep 11 00:00:00 EDT 2017': 74,
   'Wed Aug 20 00:00:00 EDT 2014': 73,
   'Mon Aug 04 00:00:00 EDT 2014': 73,
   'Thu Jul 13 00:00:00 EDT 2017': 71,
   'Tue Jun 05 00:00:00 EDT 2018': 69,
   'Wed Jul 27 00:00:00 EDT 2016': 68,
   'Thu Jun 26 00:00:00 EDT 2014': 65,
   'Mon Aug 28 00:00:00 EDT 2017': 63,
   'Wed Jul 19 00:00:00 EDT 2017': 61,
   'Wed Oct 05 00:00:00 EDT 2016': 61,
   'Wed Jul 20 00:00:00 EDT 2016': 60,
   'Fri May 29 00:00:00 EDT 2015': 59,
   'Fri Nov 03 00:00:00 EDT 2017': 59,
   'Wed Jun 24 00:00:00 EDT 2015': 56,
   'Thu Sep 11 00:00:00 EDT 2014': 54,
   'Thu Jun 02 00:00:00 EDT 2016': 54,
   'Mon Apr 07 00:00:00 EDT 2014': 52,
   'Wed Aug 16 00:00:00 EDT 2017': 52,
   'Wed Dec 18 00:00:00 EST 2013': 52,
   'Thu Sep 01 00:00:00 EDT 2016': 52,
   'Mon Jun 16 00:00:00 EDT 2014': 51,
   'Thu Sep 04 00:00:00 EDT 2014': 51,
   'Fri Sep 04 00:00:00 EDT 2015': 50,
   'Wed Aug 10 00:00:00 EDT 2016': 50,
   'Thu Oct 09 00:00:00 EDT 2014': 45,
   'Thu Sep 21 00:00:00 EDT 2017': 44,
   'Fri Nov 10 00:00:00 EST 2017': 44,
   'Mon Nov 07 00:00:00 EST 2016': 43,
   'Mon Sep 26 00:00:00 EDT 2016': 43,
   'Wed Aug 17 00:00:00 EDT 2016': 42,
   'Mon Oct 19 00:00:00 EDT 2015': 42,
   'Mon Feb 17 00:00:00 EST 2014': 42,
   'Tue May 29 00:00:00 EDT 2018': 41,
   'Tue Aug 12 00:00:00 EDT 2014': 41,
   'Tue Sep 13 00:00:00 EDT 2016': 40,
   'Wed Apr 05 00:00:00 EDT 2017': 39,
   'Mon Jul 07 00:00:00 EDT 2014': 39,
   'Wed Oct 19 00:00:00 EDT 2016': 38,
   'Tue Mar 17 00:00:00 EDT 2015': 38,
   'Wed Jul 23 00:00:00 EDT 2014': 38,
   'Fri May 30 00:00:00 EDT 2014': 38,
   'Mon Oct 23 00:00:00 EDT 2017': 38,
   'Mon Jun 12 00:00:00 EDT 2017': 38,
   'Fri Jun 24 00:00:00 EDT 2016': 37,
   'Tue Jul 14 00:00:00 EDT 2015': 36,
   'Wed Jul 13 00:00:00 EDT 2016': 36,
   'Wed Oct 04 00:00:00 EDT 2017': 35,
   'Wed Jul 05 00:00:00 EDT 2017': 35,
   'Thu Aug 25 00:00:00 EDT 2016': 35,
   'Fri Sep 25 00:00:00 EDT 2015': 35,
   'Thu May 21 00:00:00 EDT 2015': 33,
   'Mon Jun 08 00:00:00 EDT 2015': 32,
   'Thu Sep 25 00:00:00 EDT 2014': 32,
   'Fri Dec 02 00:00:00 EST 2016': 32,
   'Wed Jul 16 00:00:00 EDT 2014': 31,
   'Wed May 03 00:00:00 EDT 2017': 30,
   'Fri Jun 16 00:00:00 EDT 2017': 30,
   'Tue Oct 29 00:00:00 EDT 2013': 29,
   'Thu Nov 20 00:00:00 EST 2014': 29,
   'Fri Jan 17 00:00:00 EST 2014': 29,
   'Tue Nov 22 00:00:00 EST 2016': 29,
   'Wed Aug 12 00:00:00 EDT 2015': 29,
   'Thu Dec 17 00:00:00 EST 2015': 29,
   'Wed Dec 10 00:00:00 EST 2014': 28,
   'Tue Jan 05 00:00:00 EST 2016': 28,
   'Mon Oct 12 00:00:00 EDT 2015': 28,
   'Wed Sep 27 00:00:00 EDT 2017': 27,
   'Wed Apr 20 00:00:00 EDT 2016': 27,
   'Mon Apr 04 00:00:00 EDT 2016': 27,
   'Wed Jul 25 00:00:00 EDT 2018': 27,
   'Mon Oct 06 00:00:00 EDT 2014': 27,
   'Wed May 24 00:00:00 EDT 2017': 26,
   'Wed Aug 26 00:00:00 EDT 2015': 25,
   'Thu Jul 12 00:00:00 EDT 2018': 25,
   'Wed Jun 27 00:00:00 EDT 2018': 25,
   'Fri Sep 01 00:00:00 EDT 2017': 24,
   'Wed Jun 17 00:00:00 EDT 2015': 24,
   'Tue Apr 07 00:00:00 EDT 2015': 23,
   'Wed Apr 22 00:00:00 EDT 2015': 23,
   'Mon Nov 04 00:00:00 EST 2013': 23,
   'Mon Jun 26 00:00:00 EDT 2017': 23,
   'Tue Jan 06 00:00:00 EST 2015': 23,
   'Tue Feb 02 00:00:00 EST 2016': 23,
   'Wed May 16 00:00:00 EDT 2018': 23,
   'Thu Sep 18 00:00:00 EDT 2014': 22,
   'Mon Oct 27 00:00:00 EDT 2014': 22,
   'Fri Jun 10 00:00:00 EDT 2016': 22,
   'Tue Apr 12 00:00:00 EDT 2016': 22,
   'Wed Oct 28 00:00:00 EDT 2015': 22,
   'Mon Nov 14 00:00:00 EST 2016': 21,
   'Wed Jan 03 00:00:00 EST 2018': 21,
   'Tue Jul 17 00:00:00 EDT 2018': 21,
   'Thu Nov 12 00:00:00 EST 2015': 20,
   'Wed Nov 04 00:00:00 EST 2015': 20,
   'Tue Aug 18 00:00:00 EDT 2015': 20,
   'Fri Mar 02 00:00:00 EST 2018': 20,
   'Wed Sep 09 00:00:00 EDT 2015': 19,
   'Tue Nov 28 00:00:00 EST 2017': 19,
   'Fri Oct 02 00:00:00 EDT 2015': 18,
   'Thu Apr 09 00:00:00 EDT 2015': 18,
   'Thu Oct 16 00:00:00 EDT 2014': 18,
   'Wed Feb 05 00:00:00 EST 2014': 18,
   'Thu Jun 16 00:00:00 EDT 2016': 18,
   'Wed Oct 12 00:00:00 EDT 2016': 18,
   'Tue Apr 25 00:00:00 EDT 2017': 17,
   'Thu Dec 14 00:00:00 EST 2017': 17,
   'Mon Apr 21 00:00:00 EDT 2014': 17,
   'Tue Jul 03 00:00:00 EDT 2018': 17,
   'Mon Jan 06 00:00:00 EST 2014': 17,
   'Wed May 14 00:00:00 EDT 2014': 17,
   'Mon Jun 05 00:00:00 EDT 2017': 17,
   'Tue Dec 05 00:00:00 EST 2017': 17,
   'Wed May 18 00:00:00 EDT 2016': 17,
   'Thu Jun 19 00:00:00 EDT 2014': 17,
   'Tue May 24 00:00:00 EDT 2016': 16,
   'Mon Mar 19 00:00:00 EDT 2018': 16,
   'Wed Oct 26 00:00:00 EDT 2016': 15,
   'Thu Nov 06 00:00:00 EST 2014': 15,
   'Mon May 22 00:00:00 EDT 2017': 14,
   'Mon May 05 00:00:00 EDT 2014': 14,
   'Thu Mar 24 00:00:00 EDT 2016': 14,
   'Wed Oct 11 00:00:00 EDT 2017': 14,
   'Fri Sep 16 00:00:00 EDT 2016': 14,
   'Wed May 02 00:00:00 EDT 2018': 14,
   'Wed Jun 10 00:00:00 EDT 2015': 13,
   'Mon May 07 00:00:00 EDT 2018': 13,
   'Tue Apr 10 00:00:00 EDT 2018': 13,
   'Wed Mar 25 00:00:00 EDT 2015': 13,
   'Tue May 10 00:00:00 EDT 2016': 13,
   'Tue Mar 03 00:00:00 EST 2015': 13,
   'Wed Dec 02 00:00:00 EST 2015': 13,
   'Tue Feb 06 00:00:00 EST 2018': 13,
   'Mon May 19 00:00:00 EDT 2014': 12,
   'Thu Mar 16 00:00:00 EDT 2017': 12,
   'Tue May 12 00:00:00 EDT 2015': 12,
   'Thu Nov 19 00:00:00 EST 2015': 12,
   'Tue May 05 00:00:00 EDT 2015': 12,
   'Fri Mar 21 00:00:00 EDT 2014': 12,
   'Wed May 04 00:00:00 EDT 2016': 12,
   'Thu Jan 11 00:00:00 EST 2018': 12,
   'Thu Jan 18 00:00:00 EST 2018': 12,
   'Thu Mar 17 00:00:00 EDT 2016': 12,
   'Mon Feb 19 00:00:00 EST 2018': 12,
   'Wed Feb 18 00:00:00 EST 2015': 12,
   'Thu Mar 12 00:00:00 EDT 2015': 11,
   'Thu Mar 02 00:00:00 EST 2017': 11,
   'Mon Mar 26 00:00:00 EDT 2018': 11,
   'Wed Jan 11 00:00:00 EST 2017': 11,
   'Fri Apr 25 00:00:00 EDT 2014': 10,
   'Wed Apr 19 00:00:00 EDT 2017': 10,
   'Fri Mar 10 00:00:00 EST 2017': 10,
   'Wed Mar 02 00:00:00 EST 2016': 10,
   'Thu Dec 18 00:00:00 EST 2014': 10,
   'Tue Apr 26 00:00:00 EDT 2016': 10,
   'Wed Jan 13 00:00:00 EST 2016': 10,
   'Tue May 09 00:00:00 EDT 2017': 9,
   'Wed Sep 16 00:00:00 EDT 2015': 9,
   'Fri Apr 11 00:00:00 EDT 2014': 9,
   'Wed Jan 24 00:00:00 EST 2018': 9,
   'Fri Feb 20 00:00:00 EST 2015': 9,
   'Mon Feb 12 00:00:00 EST 2018': 8,
   'Mon Dec 12 00:00:00 EST 2016': 8,
   'Thu Dec 22 00:00:00 EST 2016': 8,
   'Wed Feb 17 00:00:00 EST 2016': 8,
   'Thu Jan 28 00:00:00 EST 2016': 8,
   'Wed Apr 18 00:00:00 EDT 2018': 8,
   'Tue Mar 28 00:00:00 EDT 2017': 8,
   'Wed May 10 00:00:00 EDT 2017': 7,
   'Wed Jan 14 00:00:00 EST 2015': 7,
   'Wed Dec 20 00:00:00 EST 2017': 7,
   'Thu Feb 02 00:00:00 EST 2017': 7,
   'Mon Mar 12 00:00:00 EDT 2018': 7,
   'Tue Nov 19 00:00:00 EST 2013': 7,
   'Mon Feb 26 00:00:00 EST 2018': 7,
   'Wed Dec 17 00:00:00 EST 2014': 7,
   'Tue Jan 27 00:00:00 EST 2015': 7,
   'Thu Jan 22 00:00:00 EST 2015': 6,
   'Fri Feb 06 00:00:00 EST 2015': 6,
   'Thu Nov 16 00:00:00 EST 2017': 6,
   'Thu Dec 19 00:00:00 EST 2013': 6,
   'Fri Feb 24 00:00:00 EST 2017': 6,
   'Wed Jan 18 00:00:00 EST 2017': 6,
   'Wed Jan 04 00:00:00 EST 2017': 6,
   'Fri Dec 16 00:00:00 EST 2016': 6,
   'Wed Apr 02 00:00:00 EDT 2014': 6,
   'Thu Mar 10 00:00:00 EST 2016': 5,
   'Wed Jan 22 00:00:00 EST 2014': 5,
   'Wed Jan 25 00:00:00 EST 2017': 5,
   'Thu Dec 10 00:00:00 EST 2015': 5,
   'Mon Mar 10 00:00:00 EDT 2014': 5,
   'Fri Feb 07 00:00:00 EST 2014': 4,
   'Mon Apr 06 00:00:00 EDT 2015': 4,
   'Thu Jan 21 00:00:00 EST 2016': 4,
   'Wed Feb 10 00:00:00 EST 2016': 4,
   'Wed Feb 24 00:00:00 EST 2016': 4,
   'Wed Apr 16 00:00:00 EDT 2014': 4,
   'Thu Feb 20 00:00:00 EST 2014': 3,
   'Wed Nov 16 00:00:00 EST 2016': 3,
   'Wed Sep 10 00:00:00 EDT 2014': 3,
   'Fri Feb 10 00:00:00 EST 2017': 3,
   'Thu Nov 07 00:00:00 EST 2013': 3,
   'Wed Aug 27 00:00:00 EDT 2014': 2,
   'Tue Apr 11 00:00:00 EDT 2017': 2,
   'Mon Jan 13 00:00:00 EST 2014': 2,
   'Mon Aug 31 00:00:00 EDT 2015': 2,
   'Thu May 04 00:00:00 EDT 2017': 2,
   'Wed Dec 04 00:00:00 EST 2013': 2,
   'Fri Feb 27 00:00:00 EST 2015': 2,
   'Thu May 15 00:00:00 EDT 2014': 2,
   'Mon Oct 13 00:00:00 EDT 2014': 1,
   'Fri Jun 03 00:00:00 EDT 2016': 1,
   'Fri Feb 28 00:00:00 EST 2014': 1,
   'Tue Mar 04 00:00:00 EST 2014': 1,
   'Wed Apr 25 00:00:00 EDT 2018': 1,
   'Thu Jul 17 00:00:00 EDT 2014': 1,
   'Fri May 27 00:00:00 EDT 2016': 1,
   'Tue Nov 15 00:00:00 EST 2016': 1,
   'Wed Jun 25 00:00:00 EDT 2014': 1,
   'Fri Jun 27 00:00:00 EDT 2014': 1,
   'Mon Jan 12 00:00:00 EST 2015': 1}},
 'modifiedOn': {'Unique Values': 107,
  'Most Frequent': 'Tue Jan 01 00:00:00 EST 2013',
  'Frequency': {'Tue Jan 01 00:00:00 EST 2013': 29437,
   'Mon Sep 26 00:00:00 EDT 2016': 6346,
   'Wed Apr 05 00:00:00 EDT 2017': 911,
   'Wed Dec 20 00:00:00 EST 2017': 280,
   'Tue Nov 22 00:00:00 EST 2016': 191,
   'Fri Dec 02 00:00:00 EST 2016': 183,
   'Thu Sep 21 00:00:00 EDT 2017': 129,
   'Wed Apr 04 00:00:00 EDT 2018': 112,
   'Mon Feb 26 00:00:00 EST 2018': 106,
   'Wed Jan 24 00:00:00 EST 2018': 97,
   'Fri Sep 09 00:00:00 EDT 2016': 94,
   'Mon Feb 12 00:00:00 EST 2018': 88,
   'Wed Jan 25 00:00:00 EST 2017': 87,
   'Tue Apr 11 00:00:00 EDT 2017': 87,
   'Wed Oct 11 00:00:00 EDT 2017': 87,
   'Wed Jun 13 00:00:00 EDT 2018': 84,
   'Wed Jul 25 00:00:00 EDT 2018': 78,
   'Tue Jun 05 00:00:00 EDT 2018': 73,
   'Wed Jan 18 00:00:00 EST 2017': 70,
   'Fri Sep 16 00:00:00 EDT 2016': 65,
   'Wed Apr 18 00:00:00 EDT 2018': 63,
   'Wed Oct 05 00:00:00 EDT 2016': 58,
   'Fri Nov 10 00:00:00 EST 2017': 58,
   'Thu Aug 10 00:00:00 EDT 2017': 56,
   'Wed Jul 26 00:00:00 EDT 2017': 55,
   'Tue Feb 21 00:00:00 EST 2017': 54,
   'Tue Feb 06 00:00:00 EST 2018': 52,
   'Fri Nov 03 00:00:00 EDT 2017': 51,
   'Mon Oct 23 00:00:00 EDT 2017': 50,
   'Mon Feb 17 00:00:00 EST 2014': 49,
   'Wed Jul 19 00:00:00 EDT 2017': 46,
   'Tue May 29 00:00:00 EDT 2018': 42,
   'Fri Jun 16 00:00:00 EDT 2017': 40,
   'Mon Aug 28 00:00:00 EDT 2017': 36,
   'Wed Apr 25 00:00:00 EDT 2018': 35,
   'Wed Jul 05 00:00:00 EDT 2017': 31,
   'Mon May 05 00:00:00 EDT 2014': 30,
   'Mon Mar 26 00:00:00 EDT 2018': 29,
   'Tue Jul 17 00:00:00 EDT 2018': 27,
   'Mon Jun 05 00:00:00 EDT 2017': 26,
   'Wed Jun 27 00:00:00 EDT 2018': 25,
   'Thu Mar 16 00:00:00 EDT 2017': 25,
   'Mon Dec 12 00:00:00 EST 2016': 24,
   'Wed Oct 26 00:00:00 EDT 2016': 22,
   'Thu Jul 12 00:00:00 EDT 2018': 20,
   'Wed May 16 00:00:00 EDT 2018': 20,
   'Wed May 04 00:00:00 EDT 2016': 20,
   'Tue Nov 28 00:00:00 EST 2017': 19,
   'Wed Sep 27 00:00:00 EDT 2017': 18,
   'Fri Apr 01 00:00:00 EDT 2016': 17,
   'Tue Jul 03 00:00:00 EDT 2018': 17,
   'Wed Apr 19 00:00:00 EDT 2017': 17,
   'Mon Feb 19 00:00:00 EST 2018': 16,
   'Wed Aug 17 00:00:00 EDT 2016': 15,
   'Tue Apr 10 00:00:00 EDT 2018': 15,
   'Tue Sep 26 00:00:00 EDT 2017': 13,
   'Mon Apr 04 00:00:00 EDT 2016': 13,
   'Thu Jul 07 00:00:00 EDT 2016': 12,
   'Mon Jun 12 00:00:00 EDT 2017': 12,
   'Thu Jul 13 00:00:00 EDT 2017': 12,
   'Mon Mar 19 00:00:00 EDT 2018': 12,
   'Fri Mar 02 00:00:00 EST 2018': 12,
   'Fri Apr 11 00:00:00 EDT 2014': 11,
   'Wed May 02 00:00:00 EDT 2018': 11,
   'Wed Jan 04 00:00:00 EST 2017': 11,
   'Tue May 09 00:00:00 EDT 2017': 11,
   'Fri Nov 07 00:00:00 EST 2014': 10,
   'Mon Sep 11 00:00:00 EDT 2017': 10,
   'Wed Mar 25 00:00:00 EDT 2015': 9,
   'Thu Aug 25 00:00:00 EDT 2016': 9,
   'Thu Jan 11 00:00:00 EST 2018': 8,
   'Fri Jul 20 00:00:00 EDT 2018': 8,
   'Thu Mar 10 00:00:00 EST 2016': 8,
   'Fri Sep 01 00:00:00 EDT 2017': 8,
   'Wed Jan 11 00:00:00 EST 2017': 8,
   'Mon Mar 12 00:00:00 EDT 2018': 7,
   'Thu Dec 14 00:00:00 EST 2017': 7,
   'Tue Dec 05 00:00:00 EST 2017': 6,
   'Mon May 07 00:00:00 EDT 2018': 6,
   'Fri Jan 12 00:00:00 EST 2018': 5,
   'Mon Jun 26 00:00:00 EDT 2017': 5,
   'Thu Aug 03 00:00:00 EDT 2017': 5,
   'Wed Jan 03 00:00:00 EST 2018': 5,
   'Thu Mar 02 00:00:00 EST 2017': 4,
   'Wed May 18 00:00:00 EDT 2016': 4,
   'Tue Sep 13 00:00:00 EDT 2016': 4,
   'Fri Feb 24 00:00:00 EST 2017': 3,
   'Thu Aug 14 00:00:00 EDT 2014': 3,
   'Wed Dec 04 00:00:00 EST 2013': 3,
   'Wed Apr 16 00:00:00 EDT 2014': 2,
   'Wed Oct 12 00:00:00 EDT 2016': 2,
   'Mon May 22 00:00:00 EDT 2017': 2,
   'Wed Aug 16 00:00:00 EDT 2017': 2,
   'Thu Nov 16 00:00:00 EST 2017': 2,
   'Wed Oct 19 00:00:00 EDT 2016': 1,
   'Mon Jul 07 00:00:00 EDT 2014': 1,
   'Fri Feb 10 00:00:00 EST 2017': 1,
   'Mon Nov 14 00:00:00 EST 2016': 1,
   'Mon Nov 07 00:00:00 EST 2016': 1,
   'Tue Aug 02 00:00:00 EDT 2016': 1,
   'Mon Jan 06 00:00:00 EST 2014': 1,
   'Thu Jun 02 00:00:00 EDT 2016': 1,
   'Wed May 24 00:00:00 EDT 2017': 1,
   'Wed Jul 27 00:00:00 EDT 2016': 1,
   'Fri Dec 16 00:00:00 EST 2016': 1,
   'Thu Oct 26 00:00:00 EDT 2017': 1,
   'Fri Mar 21 00:00:00 EDT 2014': 1}},
 'startStop': {'Unique Values': 2,
  'Most Frequent': 'N',
  'Frequency': {'N': 5695, 'Y': 2682}}}
```

[99]:

```
sdf.plot_summary()
```



1.2 Dependent Variable:

[100]:

```
sdf.plot_boxplot_with_swarm(x_col='year', y_col='UCity')
```

```
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 94.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 93.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 91.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
```

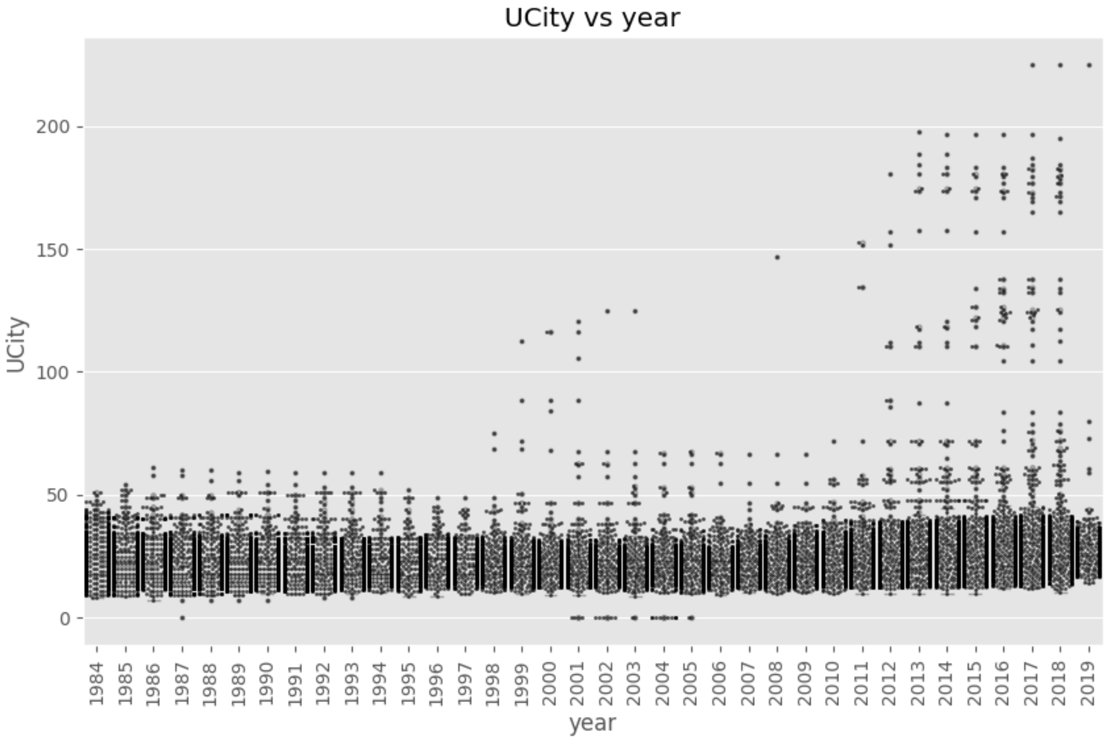
```
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 91.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 91.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
```

```
UserWarning: 86.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 86.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 88.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 88.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 88.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

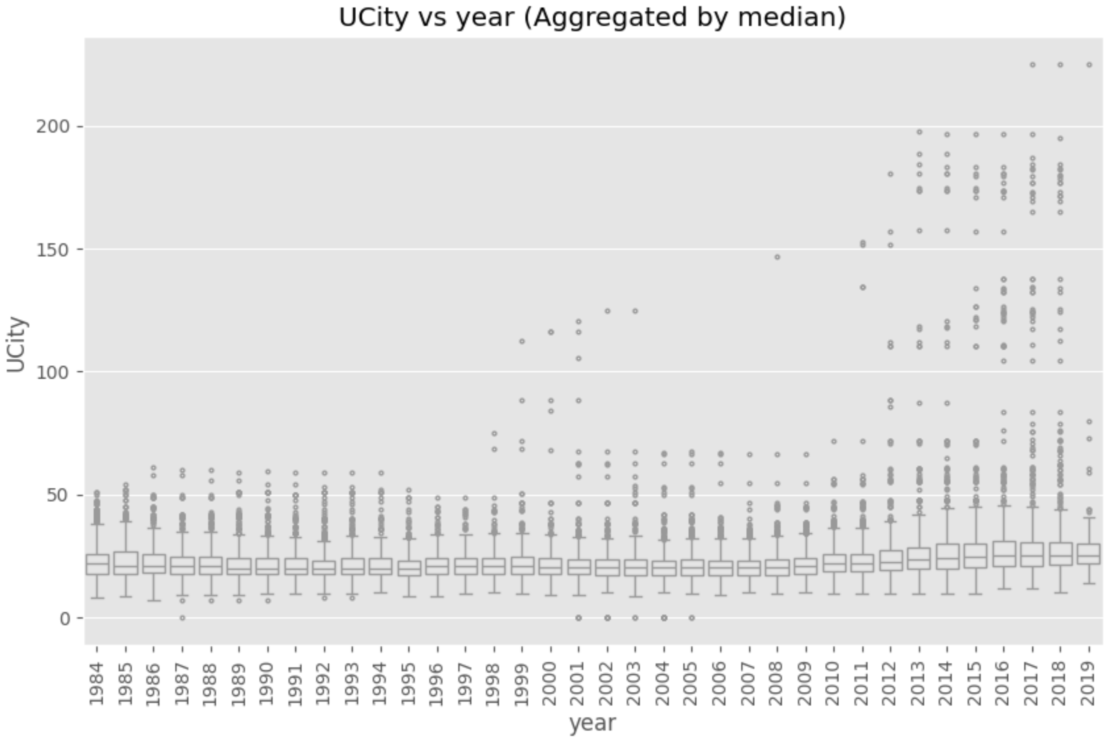
```
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 85.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 85.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 86.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 79.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 93.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 90.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.7% of the points cannot be placed; you may want to decrease the
```

```
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 86.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 85.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 86.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 88.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
```

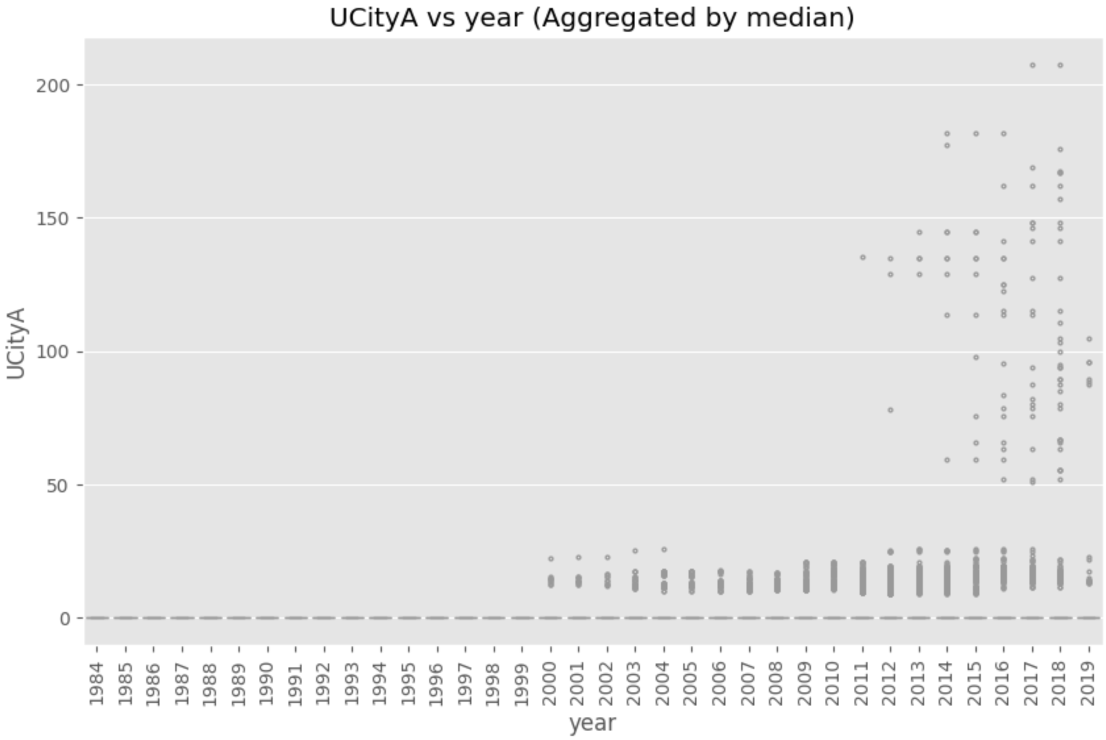
```
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 88.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 89.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 88.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 87.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 86.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 84.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/Users/alkhalifas/Documents/Github/cs6140-machine-
learning/.venv/lib/python3.11/site-packages/seaborn/categorical.py:3399:
UserWarning: 77.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



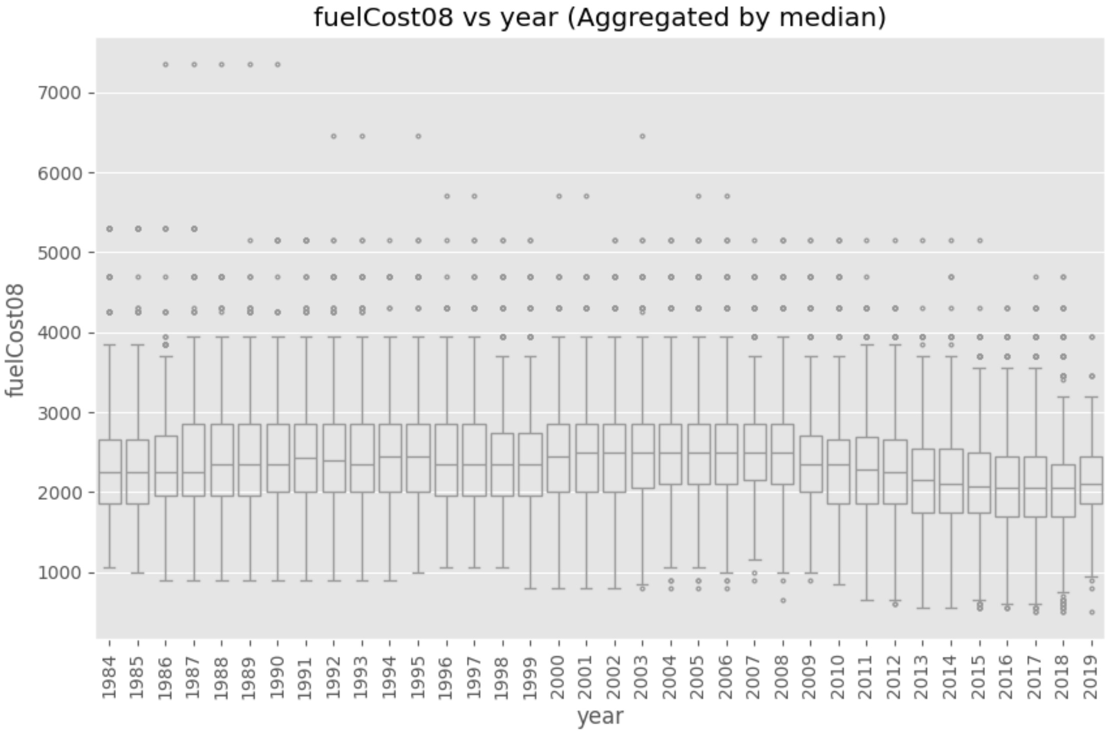
```
[101]: sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='UCity')
```



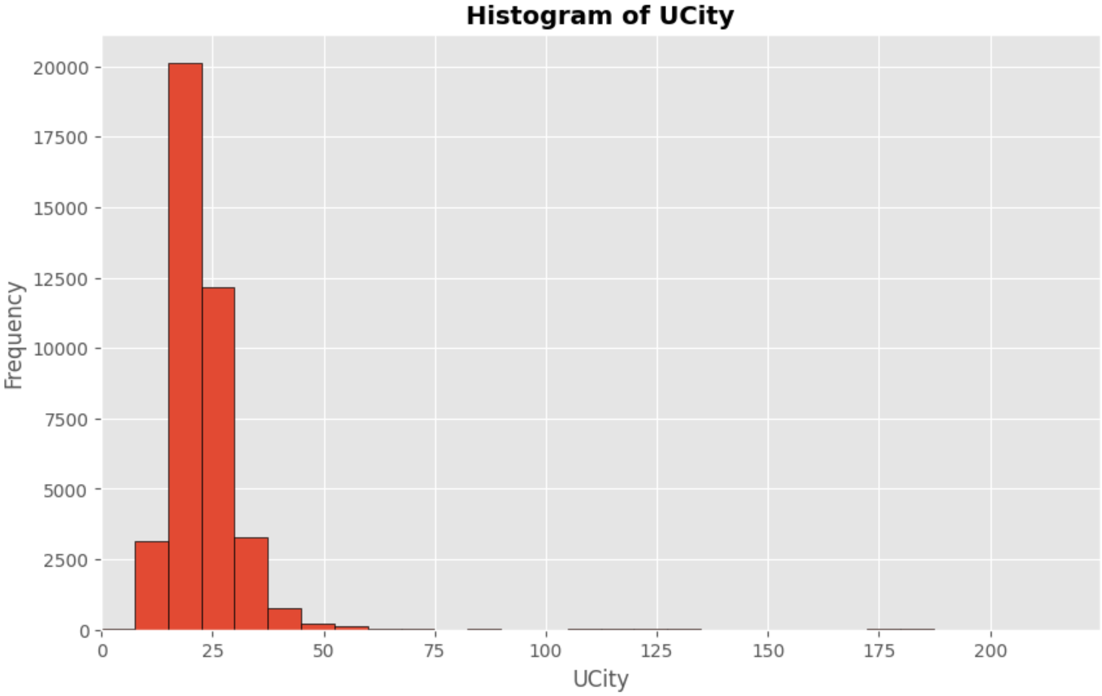
[102]:
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='UCityA')
```



[103]:
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='fuelCost08')
```

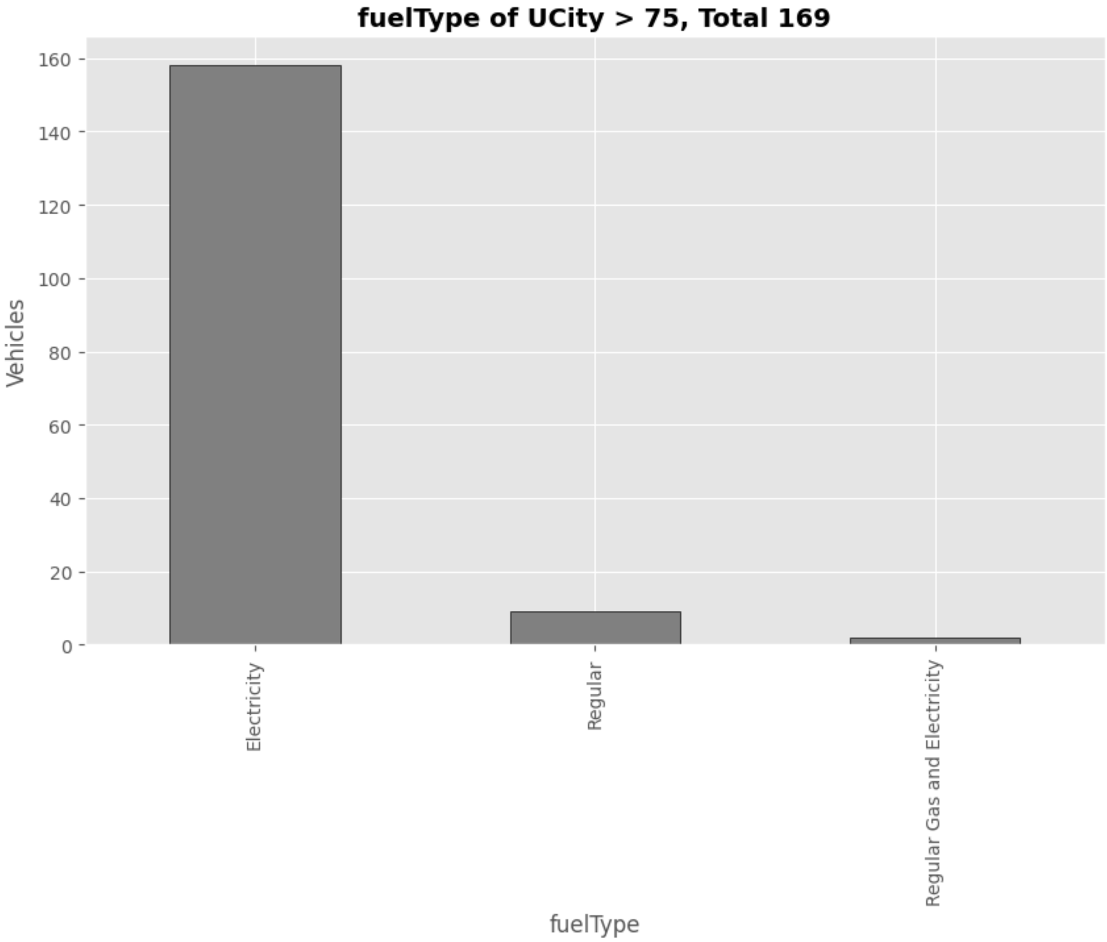


[108]:
```
sdf.plot_histogram(column='UCity', bins=30)
```



[110]:

```
sdf.plot_fuelType_distribution(y_col='UCity', filter_value=75)
```



[113]:

```
zero_UCity_vehicles = sdf.get_zero_UCity_vehicles()
```

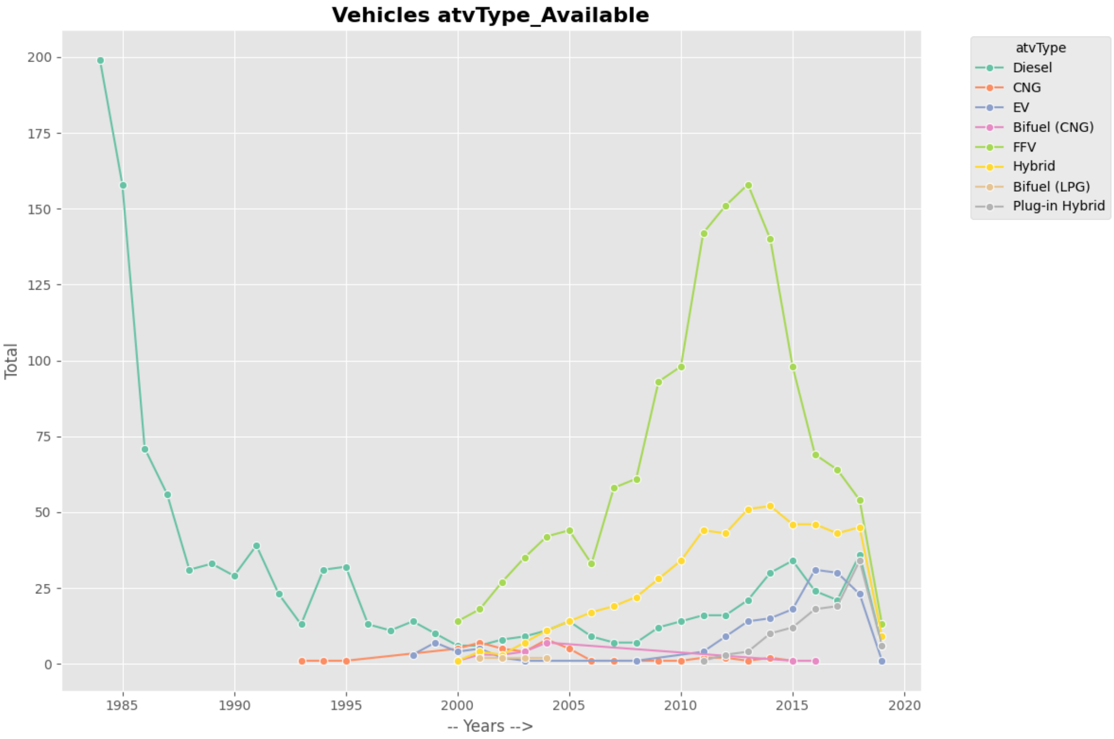
```
25 zero UCity vehicles:

         make                          model                fuelType
atvType
8127     Ford        F150 Dual-fuel 2WD (CNG)  Gasoline or natural gas
Bifuel (CNG)
8128     Ford        F150 Dual-fuel 4WD (CNG)  Gasoline or natural gas
Bifuel (CNG)
8129     Ford        F150 Dual-fuel 2WD (LPG)      Gasoline or propane
Bifuel (LPG)
8130     Ford        F150 Dual-fuel 4WD (LPG)      Gasoline or propane
```

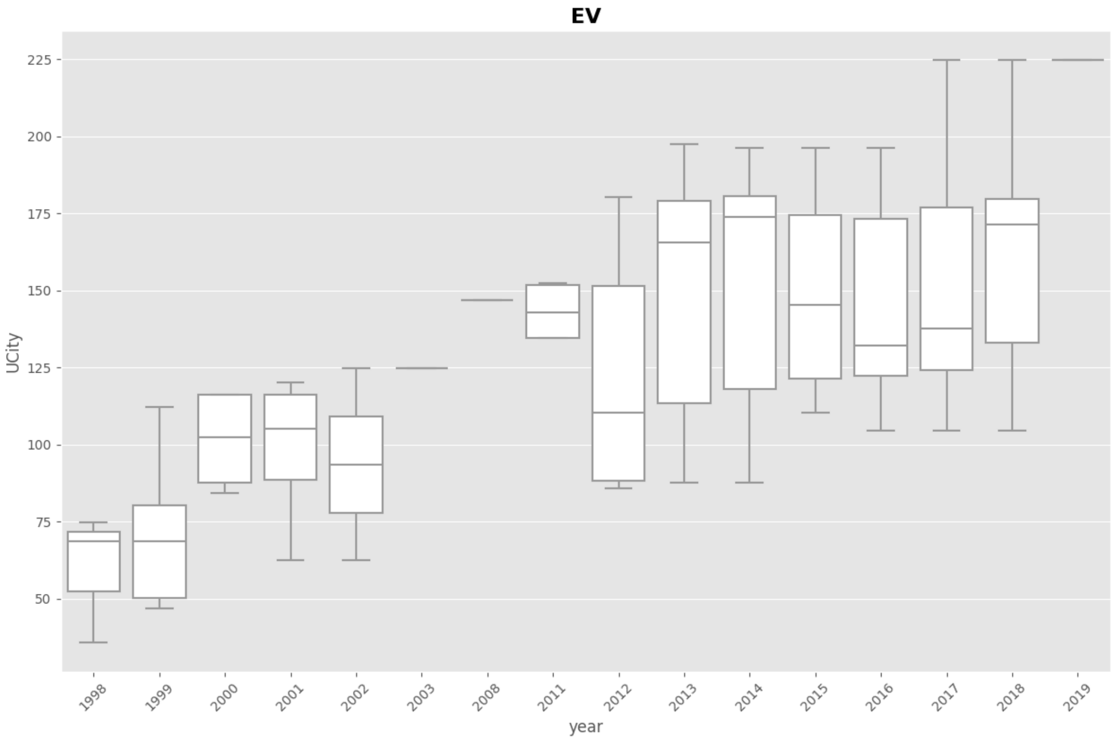
```
Bifuel (LPG)
9174       Dodge               Ram Van 2500 2WD CNG                      CNG
CNG
9175       Dodge             Ram Wagon 2500 2WD CNG                      CNG
CNG
9183        Ford           F150 Dual-fuel 2WD (CNG)  Gasoline or natural gas
Bifuel (CNG)
9184        Ford           F150 Dual-fuel 4WD (CNG)  Gasoline or natural gas
Bifuel (CNG)
9185        Ford           F150 Dual-fuel 2WD (LPG)      Gasoline or propane
Bifuel (LPG)
9186        Ford           F150 Dual-fuel 4WD (LPG)      Gasoline or propane
Bifuel (LPG)
10282       Ford           F150 Dual-fuel 2WD (LPG)      Gasoline or propane
Bifuel (LPG)
10283       Ford           F150 Dual-fuel 4WD (LPG)      Gasoline or propane
Bifuel (LPG)
11584       Ford           F150 Dual-fuel 2WD (LPG)      Gasoline or propane
Bifuel (LPG)
11585       Ford           F150 Dual-fuel 4WD (LPG)      Gasoline or propane
Bifuel (LPG)
11586  Chevrolet            Express Cargo (Bi-fuel)  Gasoline or natural gas
Bifuel (CNG)
11587  Chevrolet        Express Passenger (Bi-fuel)  Gasoline or natural gas
Bifuel (CNG)
11588        GMC           Savana (cargo) (Bi-fuel)  Gasoline or natural gas
Bifuel (CNG)
11589        GMC         Savana Passenger (Bi-fuel)  Gasoline or natural gas
Bifuel (CNG)
11591  Chevrolet      Express Cargo (dedicated CNG)                      CNG
CNG
11592  Chevrolet  Express Passenger (dedicated CNG)                      CNG
CNG
11593        GMC       Savana Cargo (dedicated CNG)                      CNG
CNG
11594        GMC   Savana Passenger (dedicated CNG)                      CNG
CNG
12814      Dodge          Caravan/Grand Caravan 2WD          Gasoline or E85
FFV
12815   Chrysler       Voyager/Town and Country 2WD          Gasoline or E85
FFV
21505    Porsche                              924 S                  Regular
NaN
```

1.3 Independent Variables:

[118]:
```
sdf.plot_atvType_over_years()
```

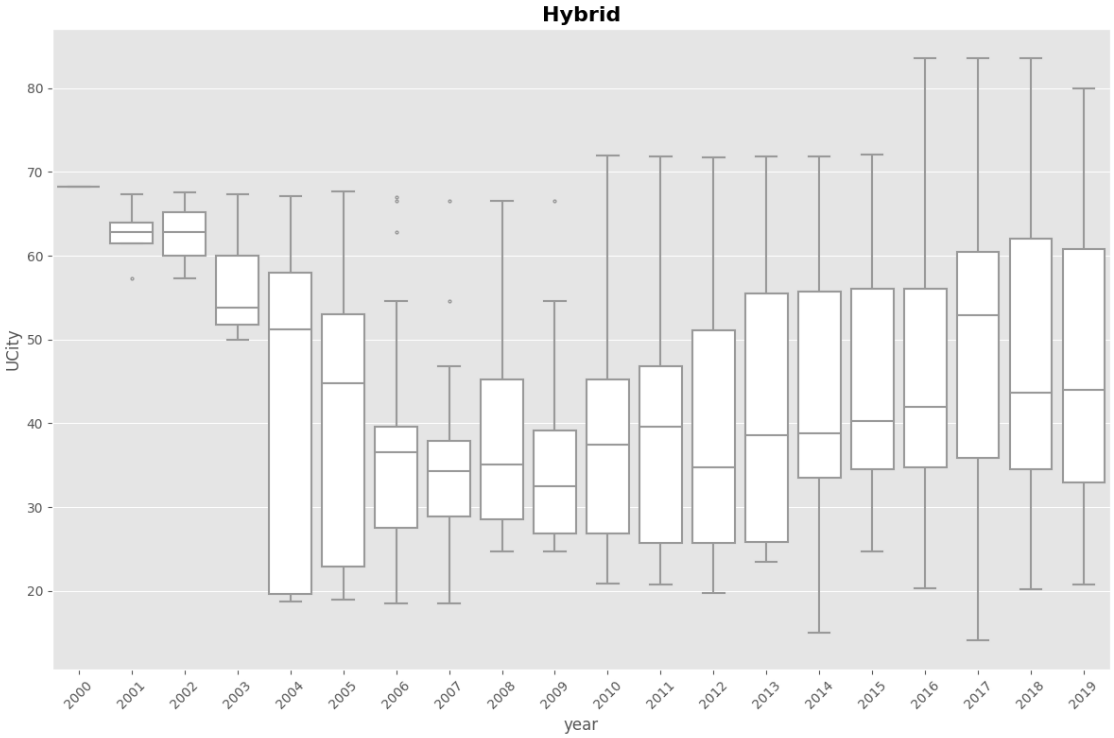


[124]:
```
sdf.plot_ucity_over_years_by_atvType('EV')
```

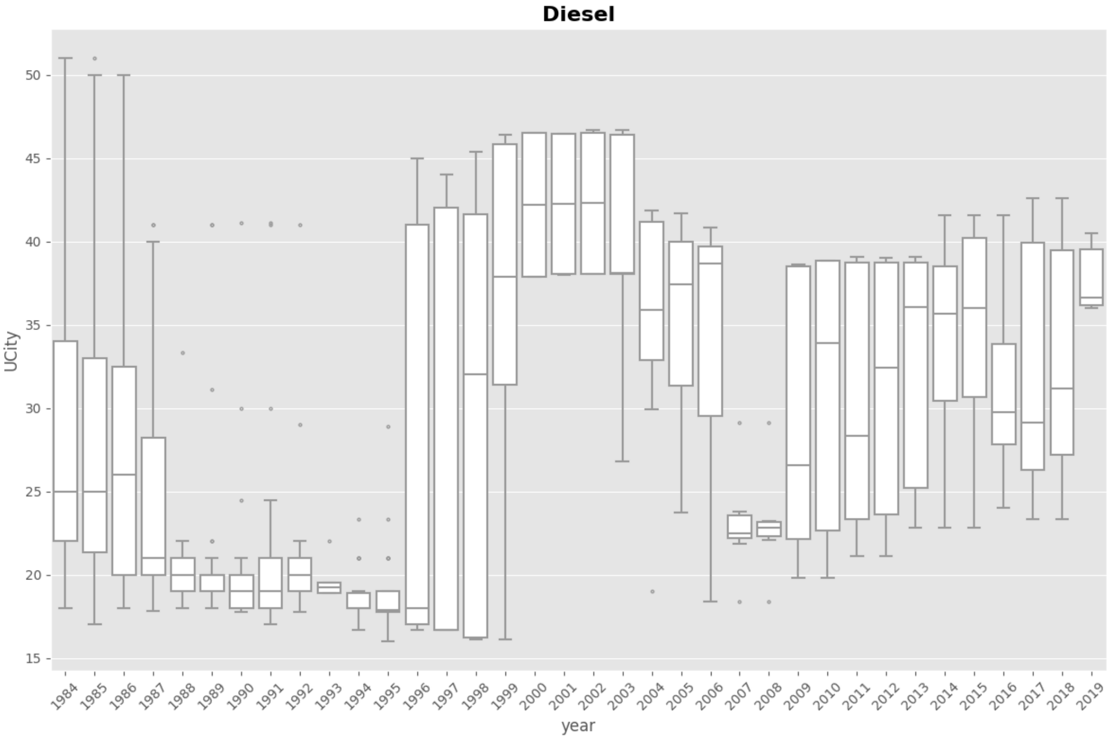


[125]:

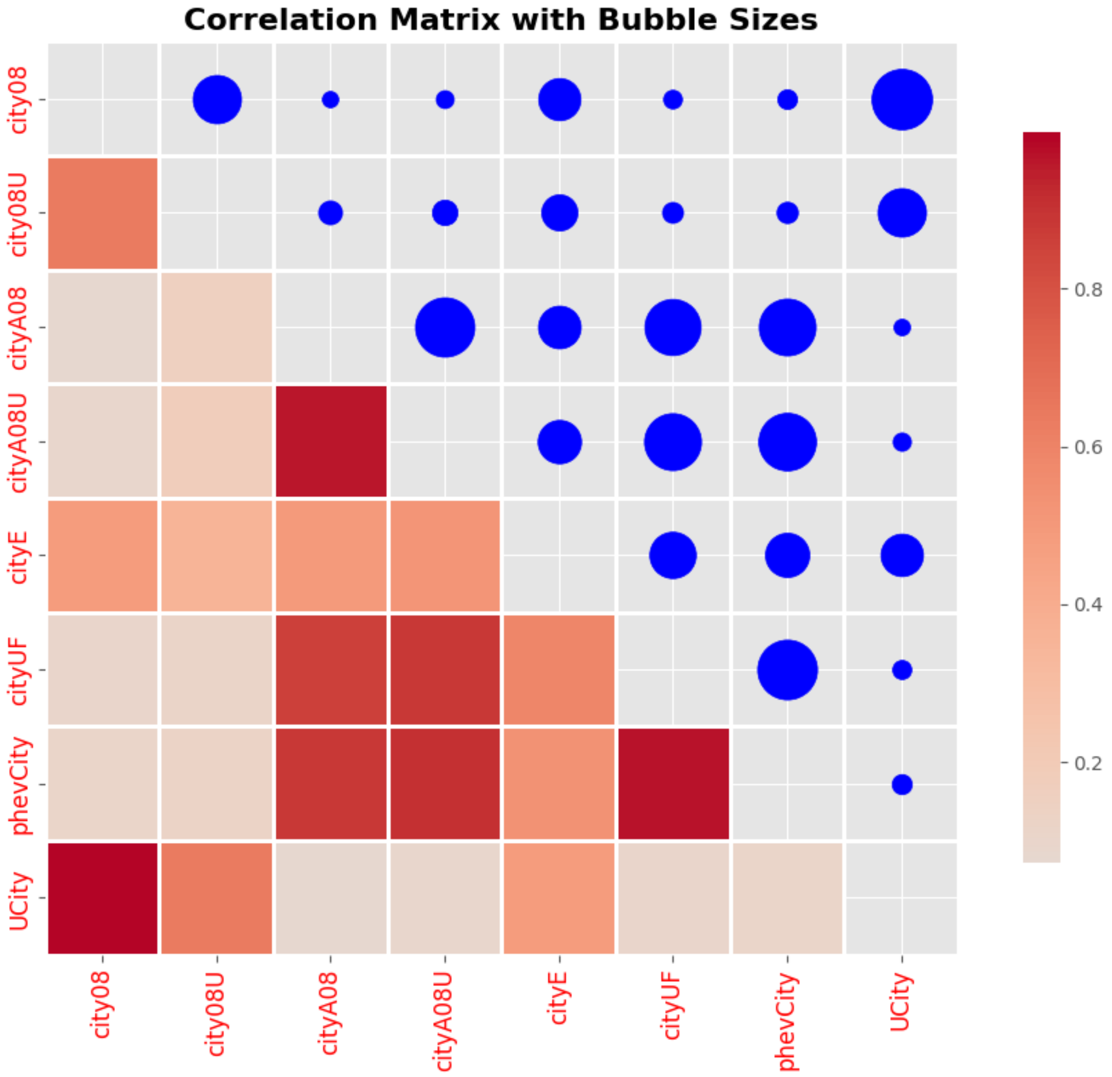
```
sdf.plot_ucity_over_years_by_atvType('Hybrid')
```



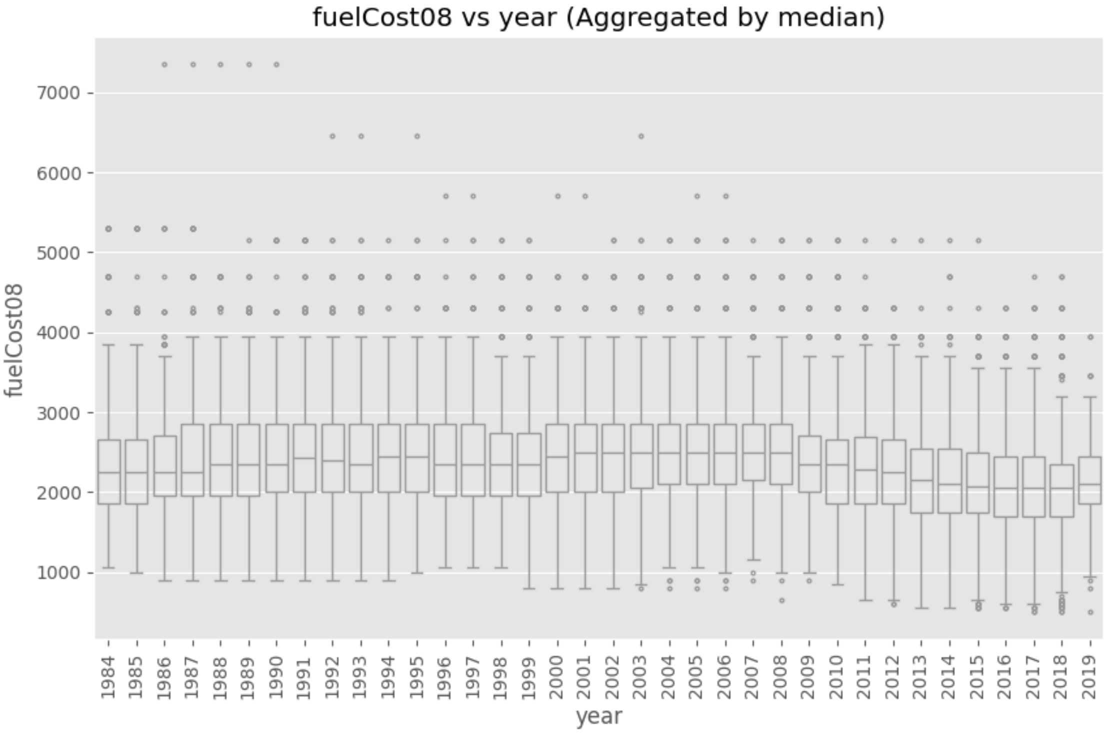
[126]:
```
sdf.plot_ucity_over_years_by_atvType('Diesel')
```



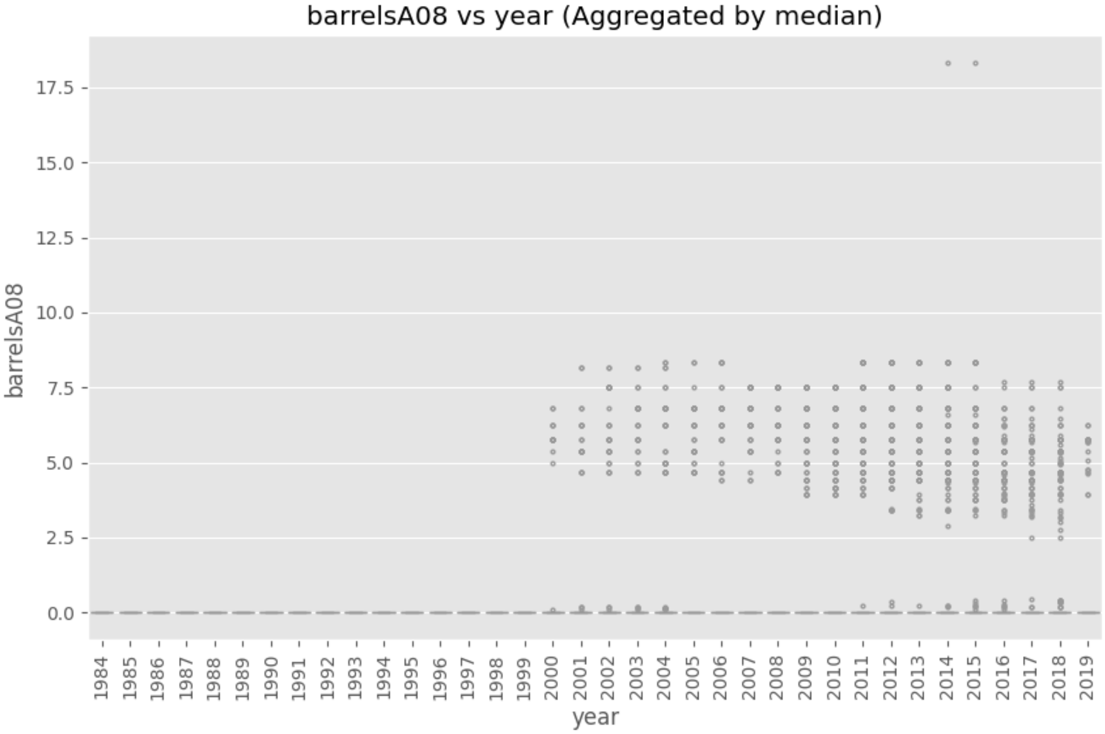
[131]:
```
sdf.plot_correlation_matrix()
```



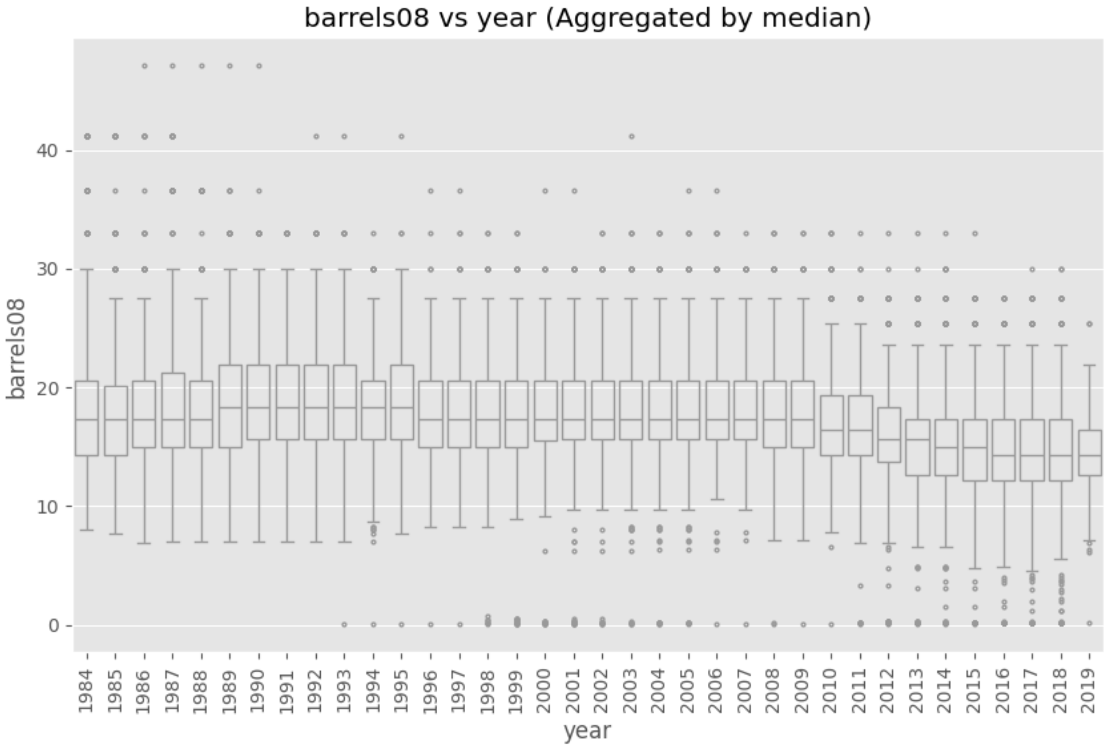
```
[153]: sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='fuelCost08')
```



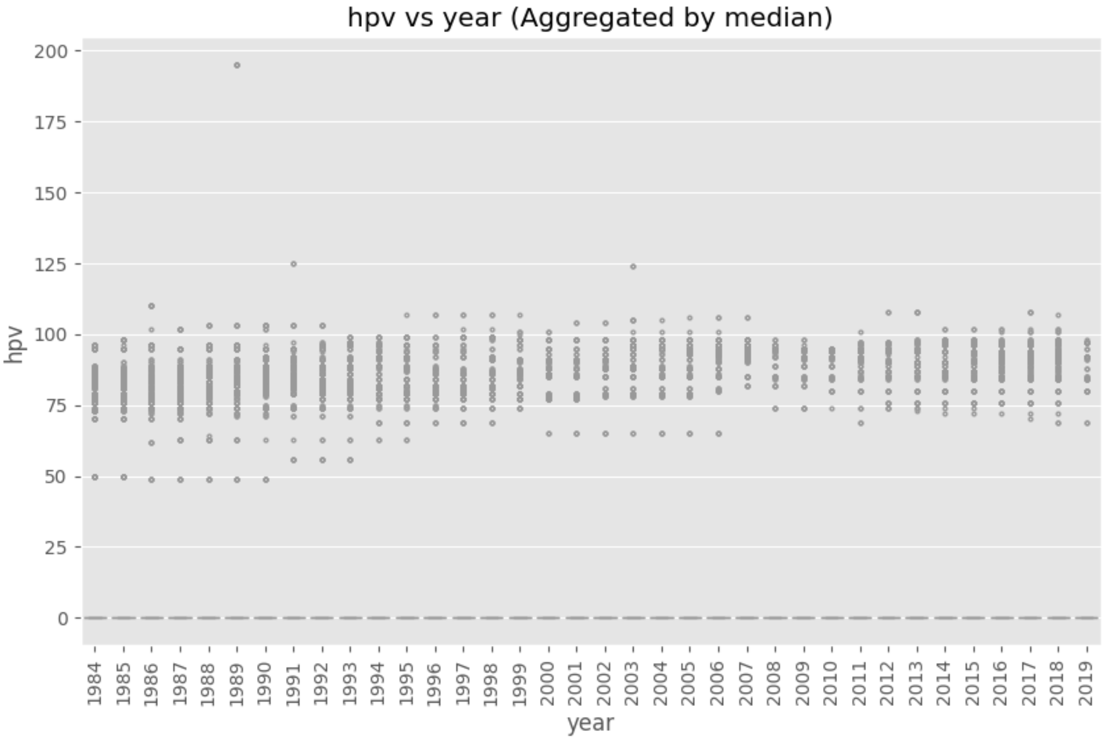
[151]:
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='barrelsA08')
```



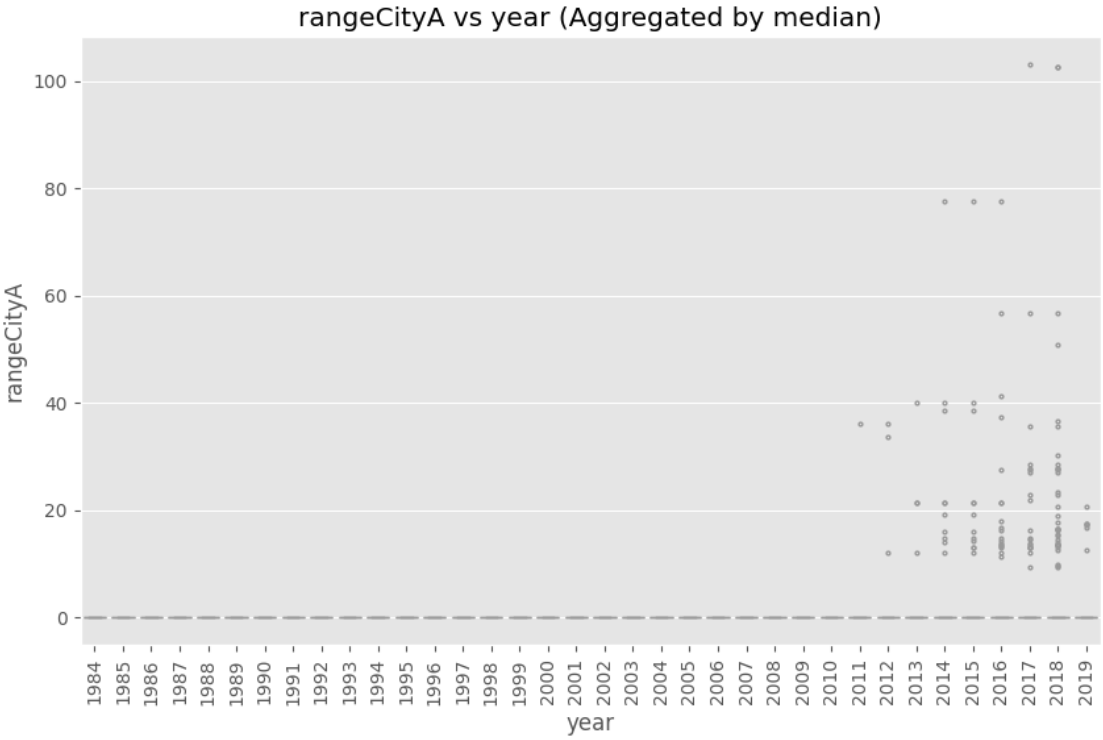
[150]:
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='barrels08')
```



[156]:
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='hpv')
```

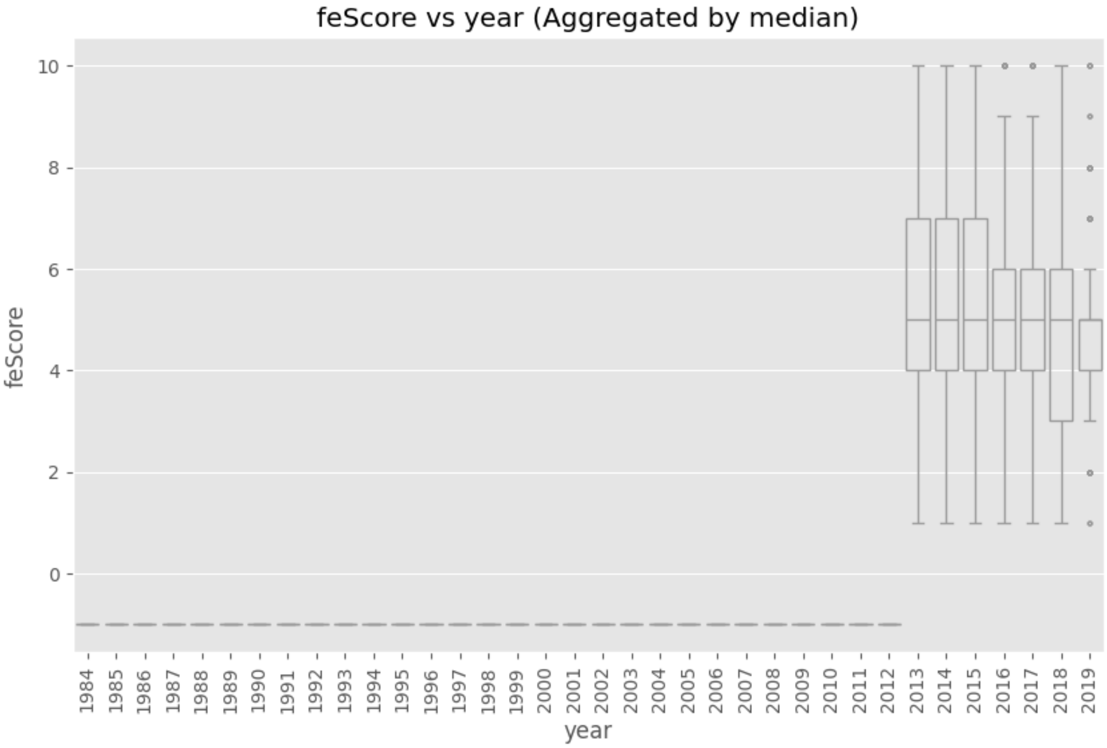


[155]:
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='rangeCityA')
```



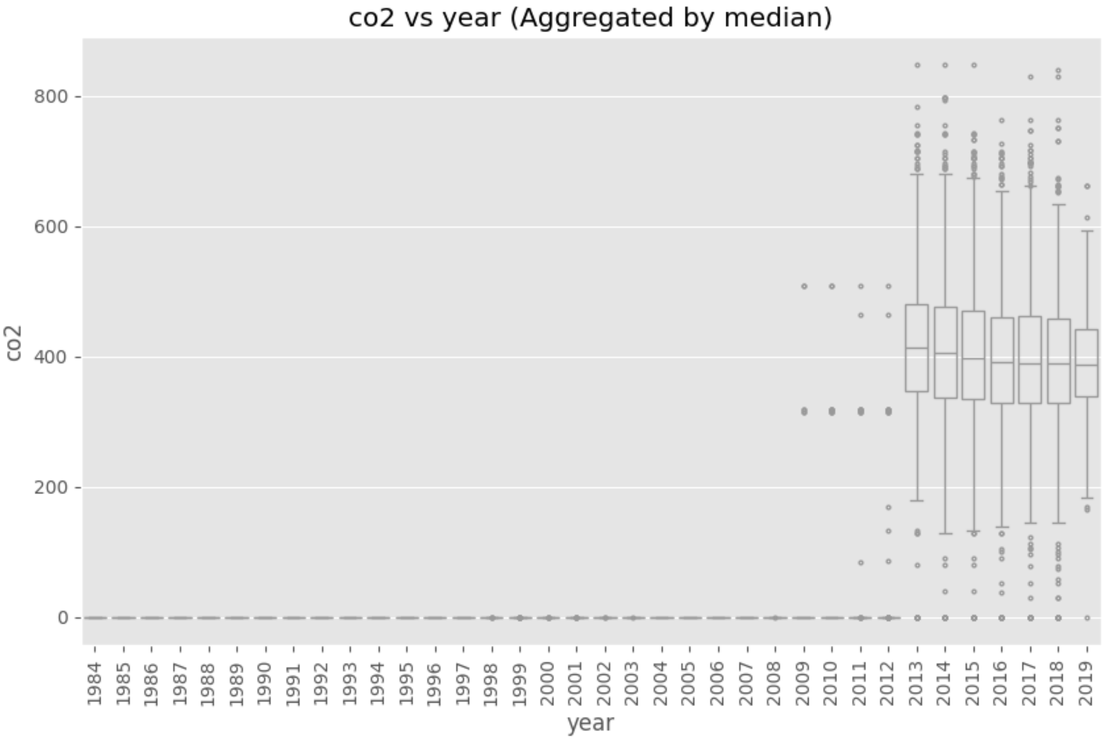
[149]:

```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='feScore')
```

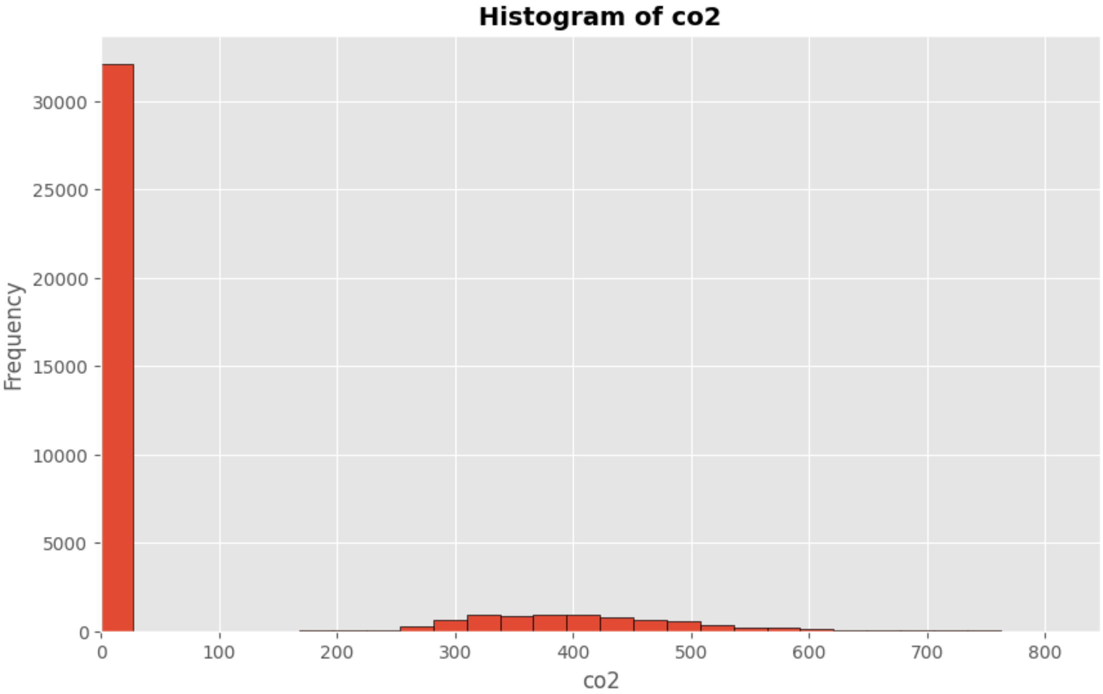


[148]:

```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='co2')
```

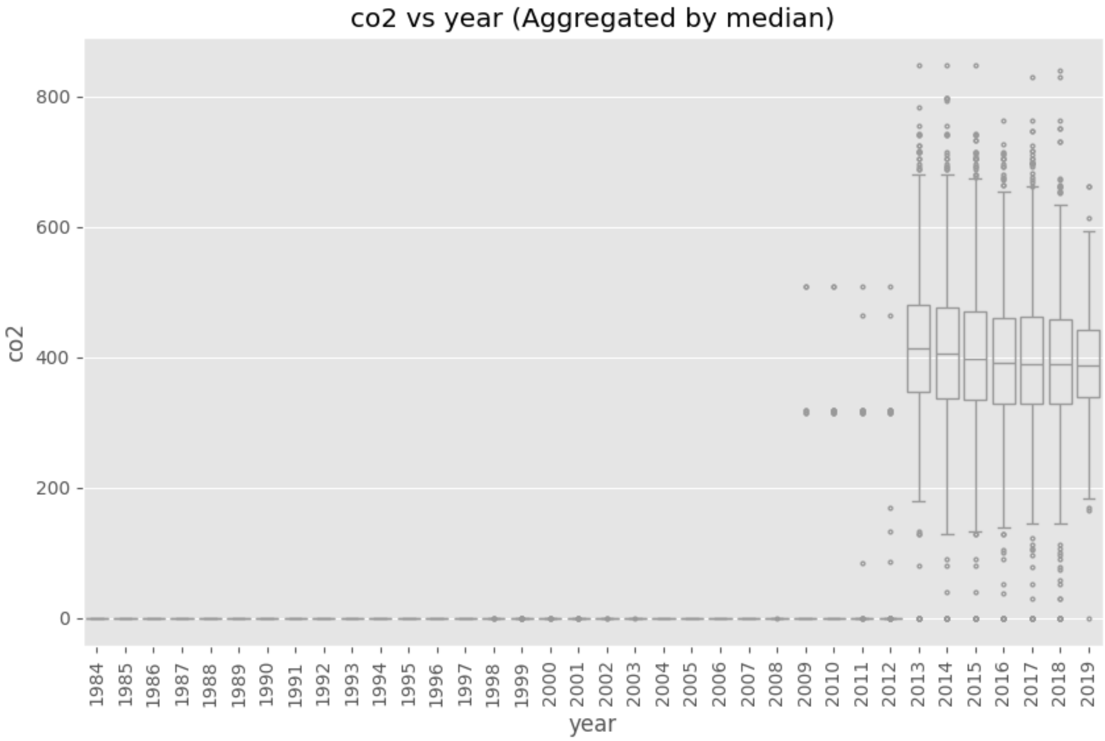


```
[114]: sdf.plot_histogram(column='co2', bins=30)
```



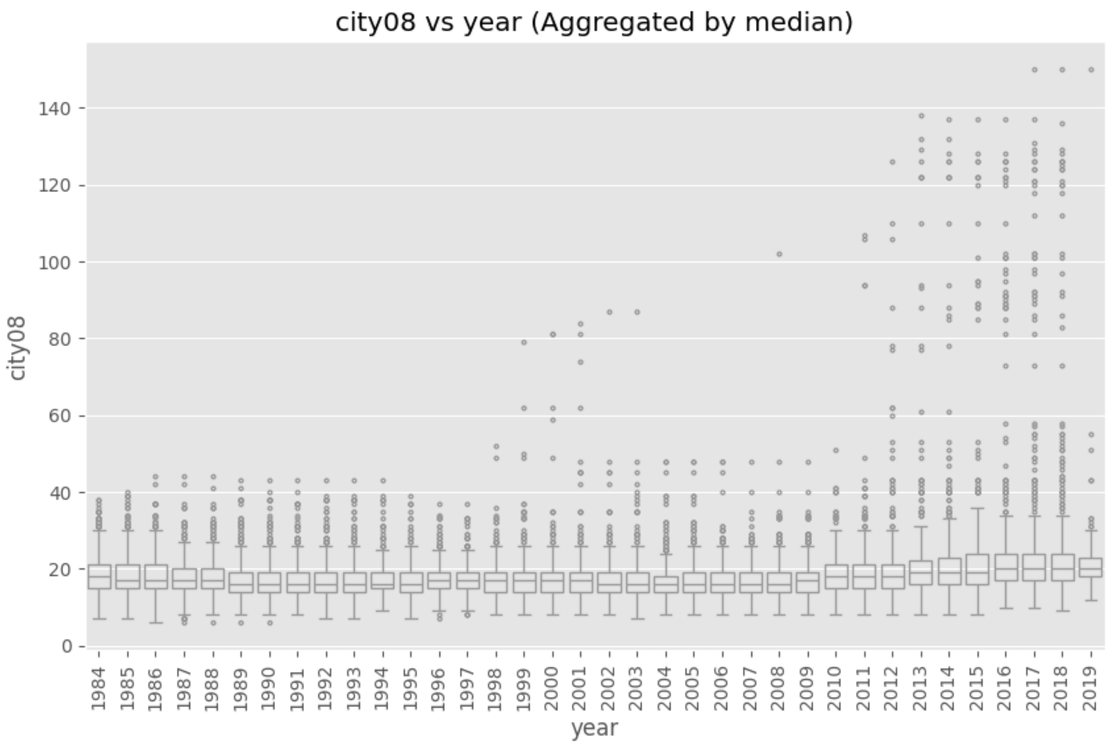
[115]:

```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='co2')
```

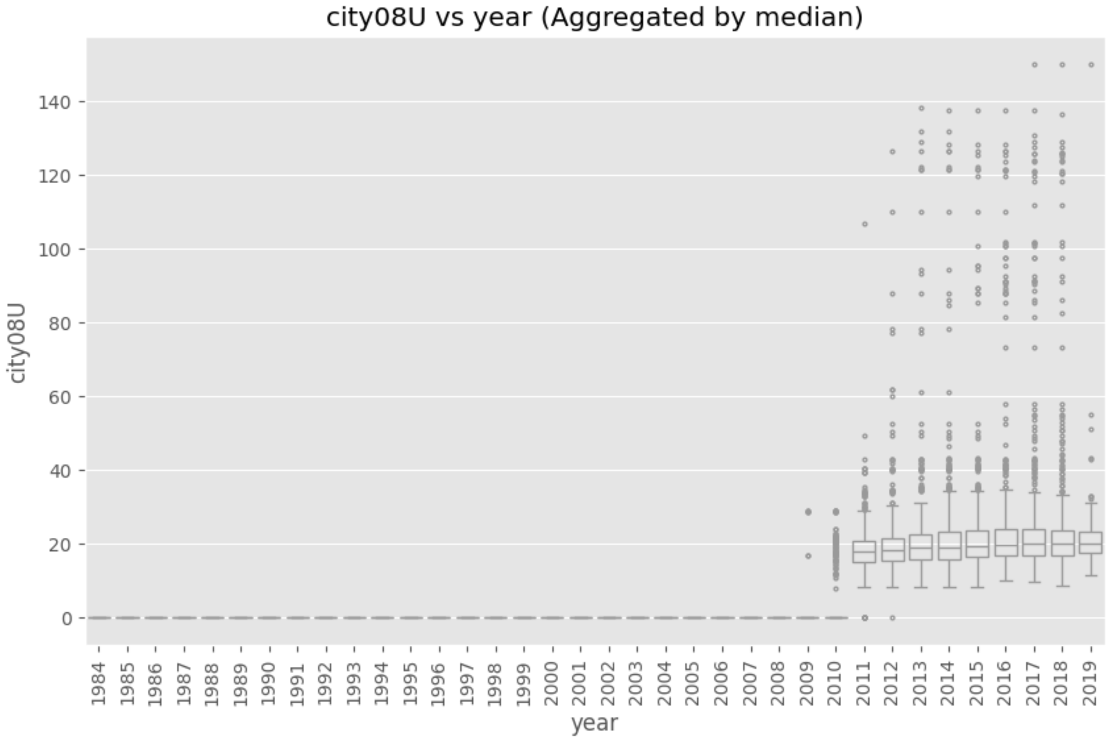


[105]:

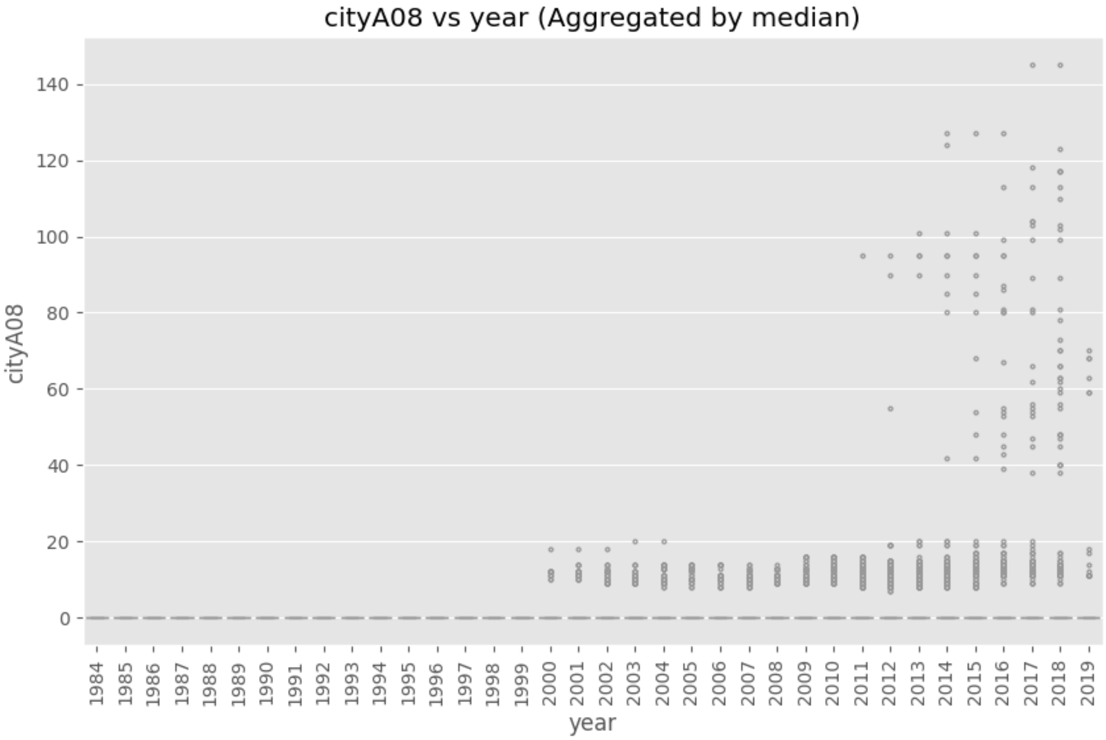
```
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='city08')
```



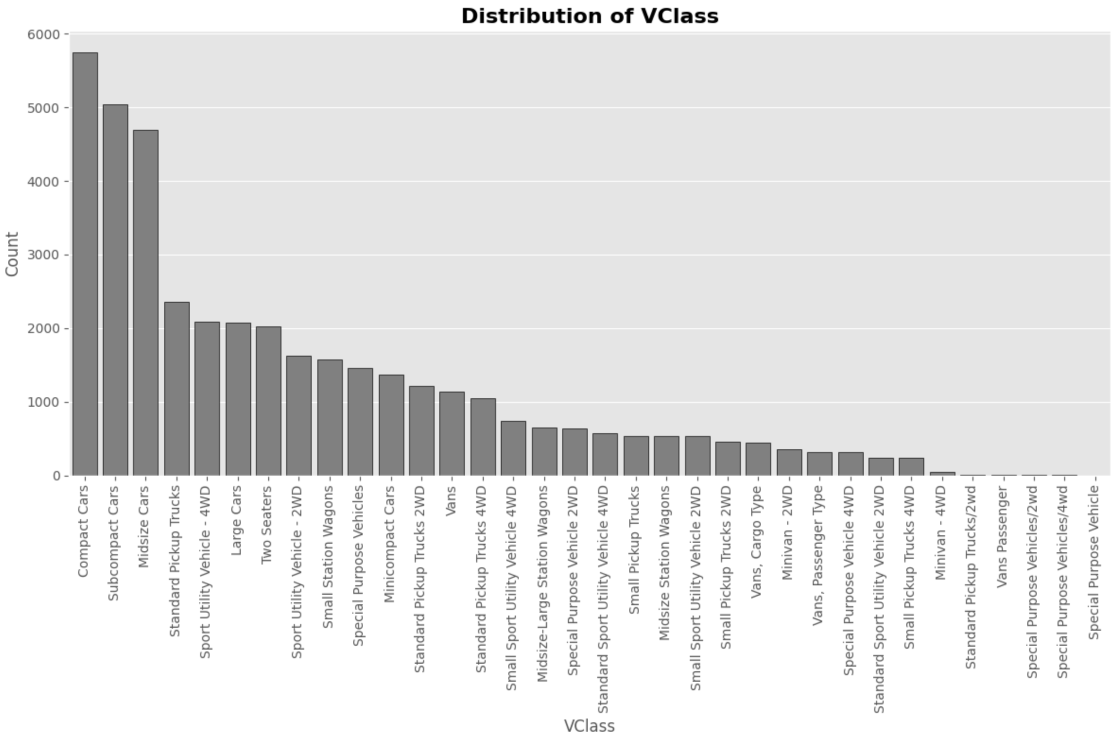
```
[106]: sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='city08U')
```



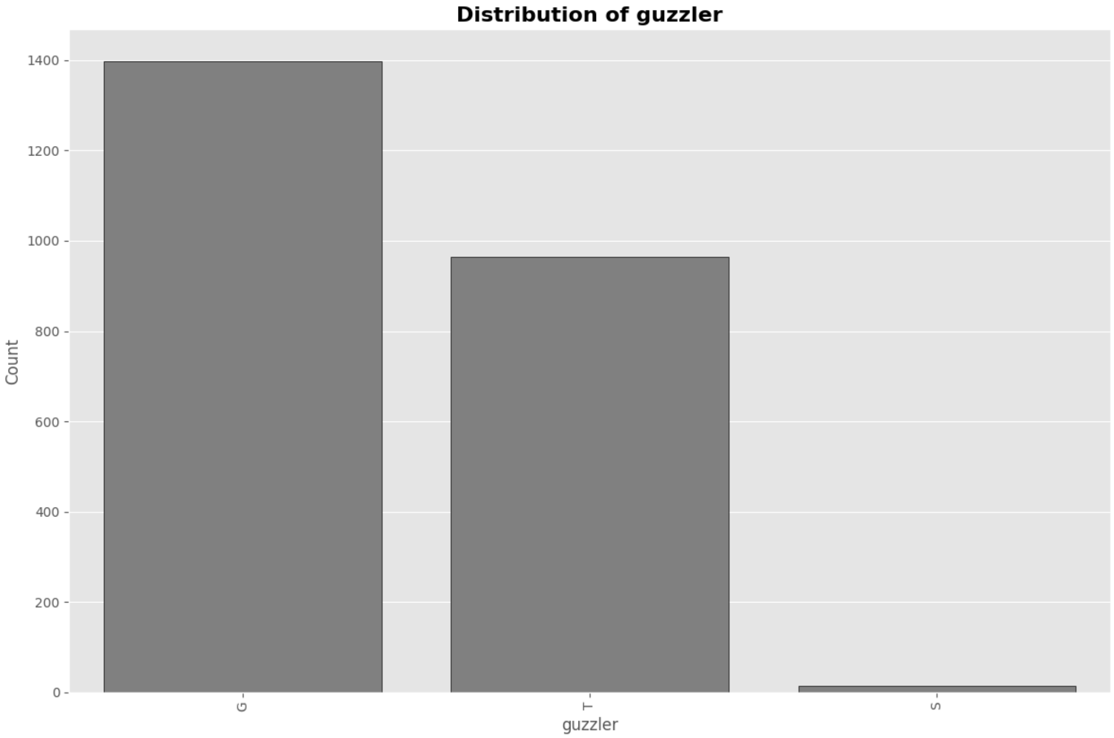
[107]:
```python
sdf.plot_scatter_with_boxplot_aggregated(x_col='year', y_col='cityA08')
```



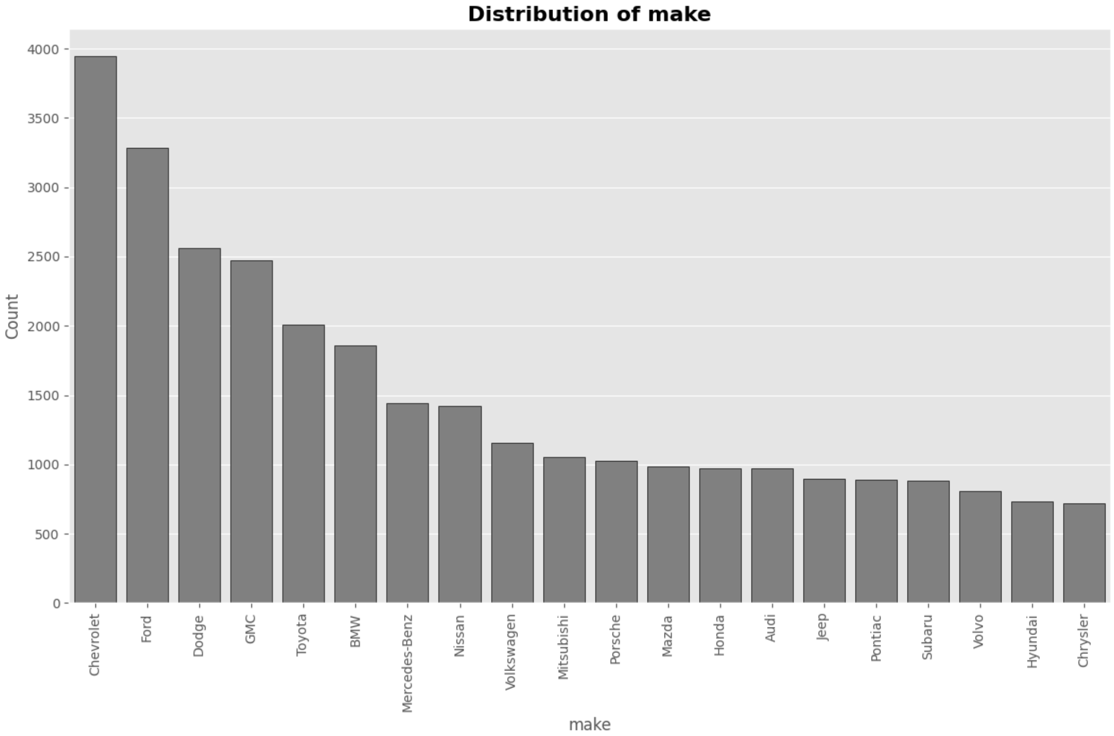
[136]:
```python
sdf.plot_feature_distribution('VClass')
```



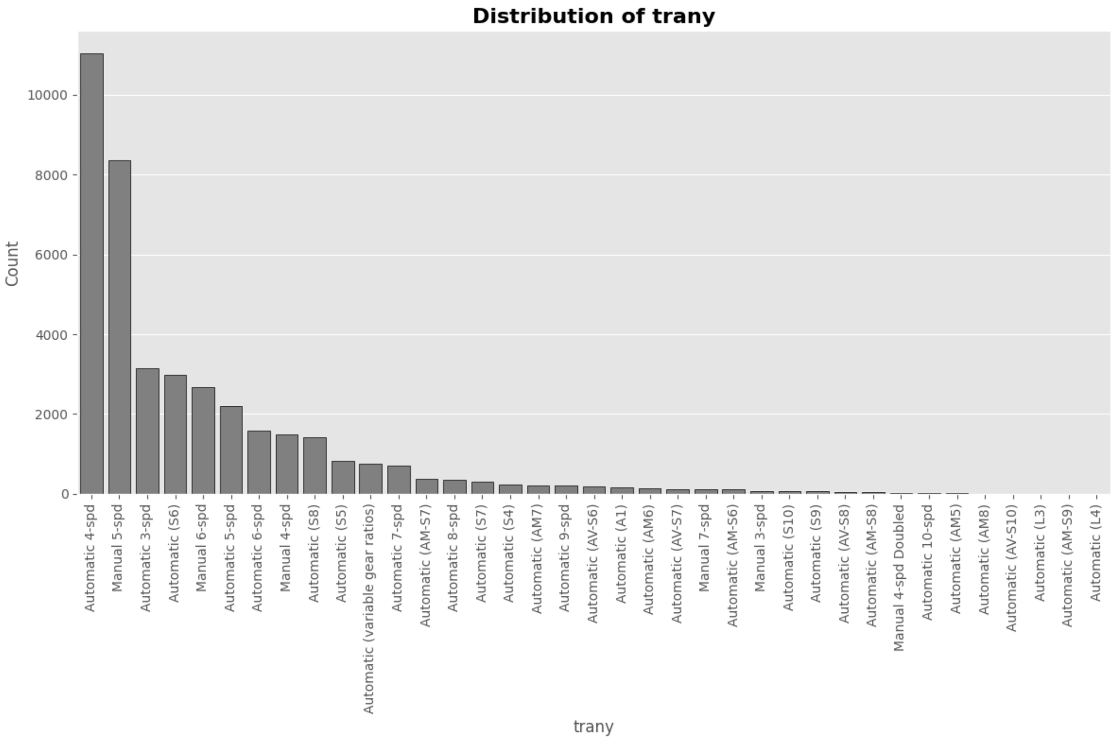
[137]:
```
sdf.plot_feature_distribution('guzzler')
```



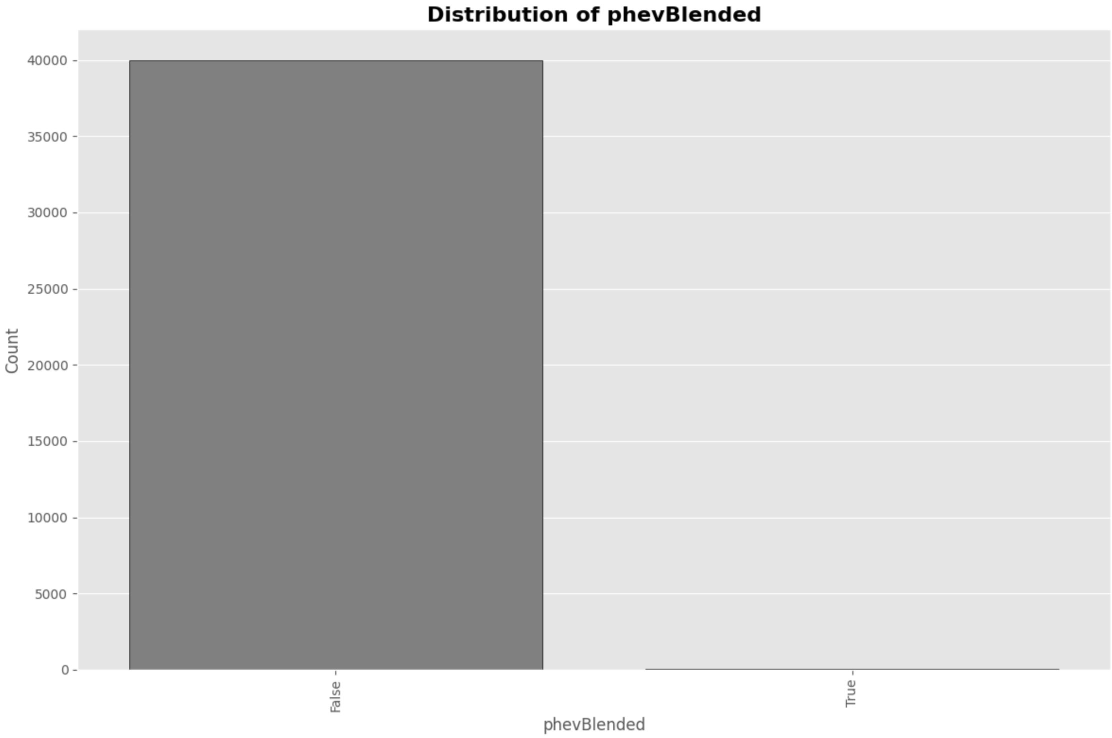
[140]:
```
sdf.plot_feature_distribution('make', 20)
```



[141]:
```
sdf.plot_feature_distribution('trany')
```

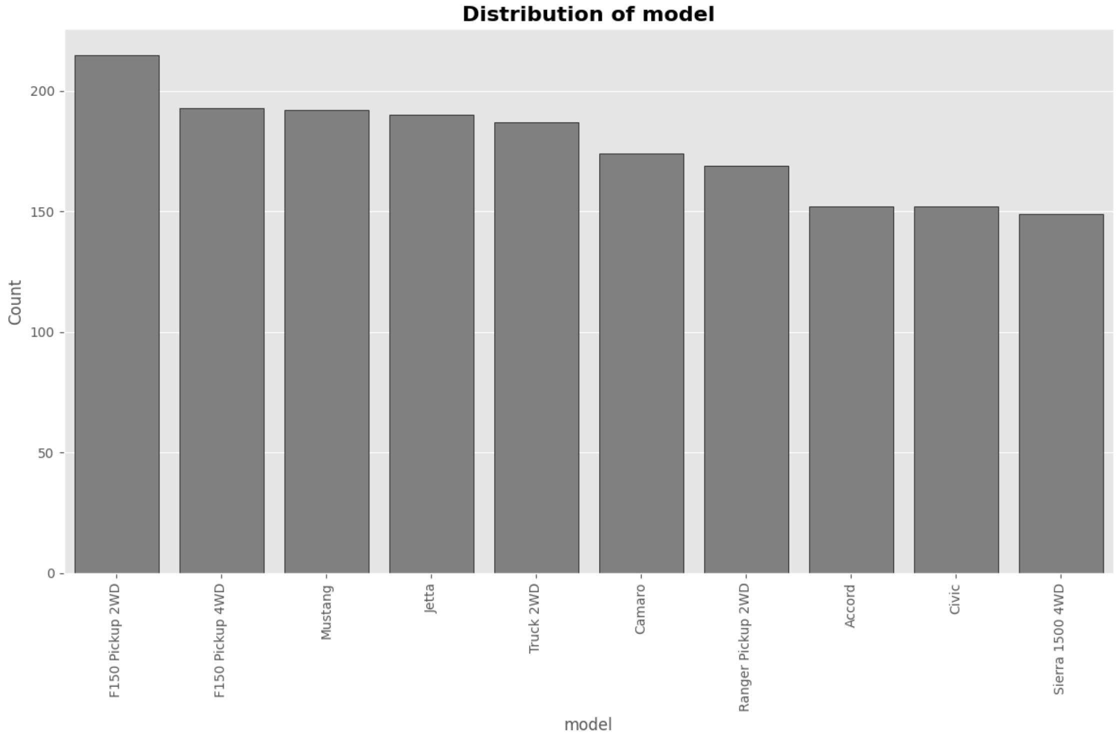


```
[145]: sdf.plot_feature_distribution('phevBlended')
```



[147]:

```
sdf.plot_feature_distribution('model', 10)
```



[37]:

```
# Get some descriptive statistics about the data
df.describe()
```

[37]:

```
          barrels08    barrelsA08  charge120    charge240        city08  \
count  40081.000000  40081.000000    40081.0 40081.000000  40081.000000
mean      17.363564      0.220069        0.0     0.036086     18.213318
std        4.597119      1.143270        0.0     0.534894      7.397433
min        0.060000      0.000000        0.0     0.000000      6.000000
25%       14.330870      0.000000        0.0     0.000000     15.000000
50%       16.480500      0.000000        0.0     0.000000     17.000000
75%       19.388824      0.000000        0.0     0.000000     20.000000
max       47.087143     18.311667        0.0    12.000000    150.000000

           city08U       cityA08      cityA08U        cityCD         cityE  \
count  40081.000000  40081.000000  40081.000000  40081.000000  40081.000000
mean       5.494777      0.616077      0.466164      0.000471      0.274113
std       11.027993      4.739349      4.563736      0.039282      3.513989
min        0.000000      0.000000      0.000000      0.000000      0.000000
```

```
25%       0.000000      0.000000      0.000000      0.000000      0.000000
50%       0.000000      0.000000      0.000000      0.000000      0.000000
75%      12.273600      0.000000      0.000000      0.000000      0.000000
max     150.000000    145.000000    145.083500      5.350000    122.000000

       …         UCity        UCityA      UHighway     UHighwayA  \
count  …  40081.000000  40081.000000  40081.000000  40081.000000
mean   …     22.981798      0.789437     34.105932      1.076877
std    …     10.473444      6.612445     10.790921      7.205627
min    …      0.000000      0.000000      0.000000      0.000000
25%    …     18.110500      0.000000     27.661300      0.000000
50%    …     21.296500      0.000000     33.024600      0.000000
75%    …     25.700000      0.000000     38.839200      0.000000
max    …    224.800000    207.262200    182.700000    173.143600

               year  youSaveSpend    charge240b      phevCity       phevHwy  \
count  40081.000000  40081.000000  40081.000000  40081.000000  40081.000000
mean    2001.068586  -4134.565006      0.007497      0.122851      0.123375
std       10.908967   3256.499139      0.195365      2.599224      2.510273
min     1984.000000 -29000.000000      0.000000      0.000000      0.000000
25%     1991.000000  -5750.000000      0.000000      0.000000      0.000000
50%     2002.000000  -4000.000000      0.000000      0.000000      0.000000
75%     2011.000000  -2000.000000      0.000000      0.000000      0.000000
max     2019.000000   5250.000000      8.500000     97.000000     81.000000

           phevComb
count  40081.000000
mean       0.122527
std        2.542274
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max       88.000000

[8 rows x 59 columns]
```

[27]:

```
# Understand the null values
df.isnull().sum()
```

[27]:

```
barrels08         0
barrelsA08        0
charge120         0
charge240         0
city08            0
                 …
modifiedOn        0
```

```
startStop     31704
phevCity          0
phevHwy           0
phevComb          0
Length: 83, dtype: int64
```

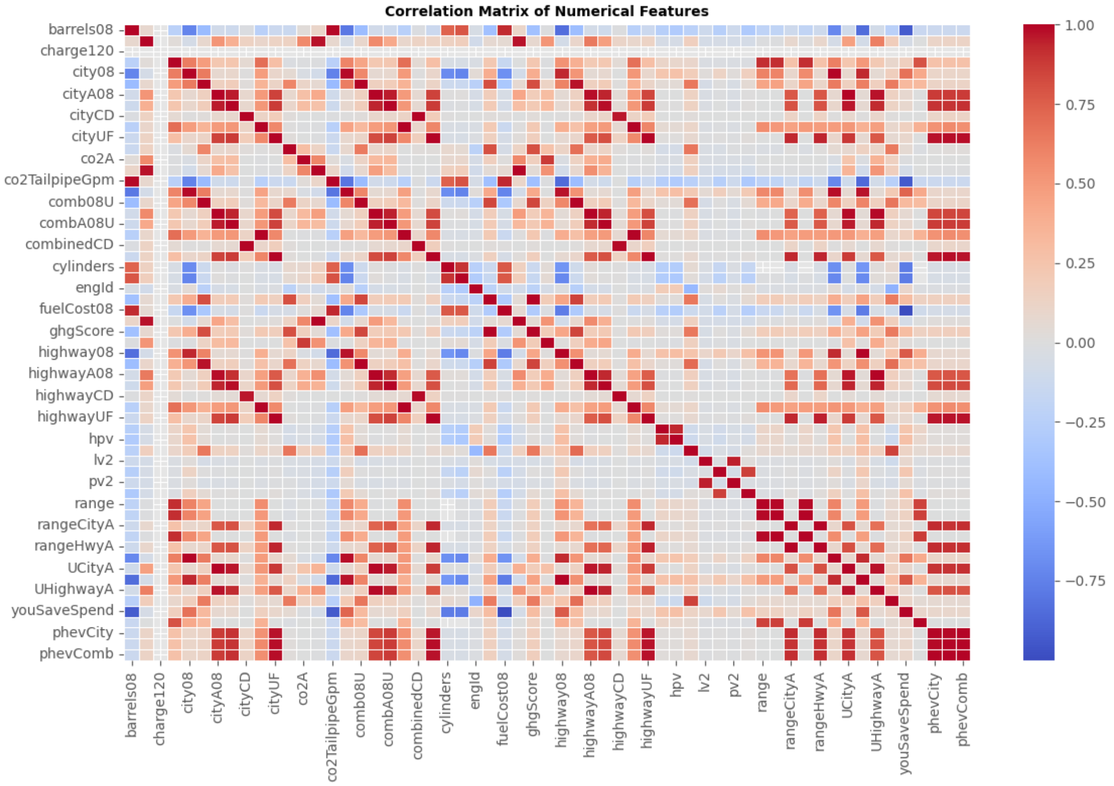
```
[28]: df['fuelType'].value_counts()
```

```
[28]: fuelType
      Regular                        25997
      Premium                        11067
      Gasoline or E85                 1287
      Diesel                          1142
      Electricity                      168
      Premium or E85                   125
      Midgrade                         100
      CNG                               60
      Premium and Electricity           47
      Regular Gas and Electricity       29
      Premium Gas or Electricity        28
      Gasoline or natural gas           20
      Gasoline or propane                8
      Regular Gas or Electricity         3
      Name: count, dtype: int64
```

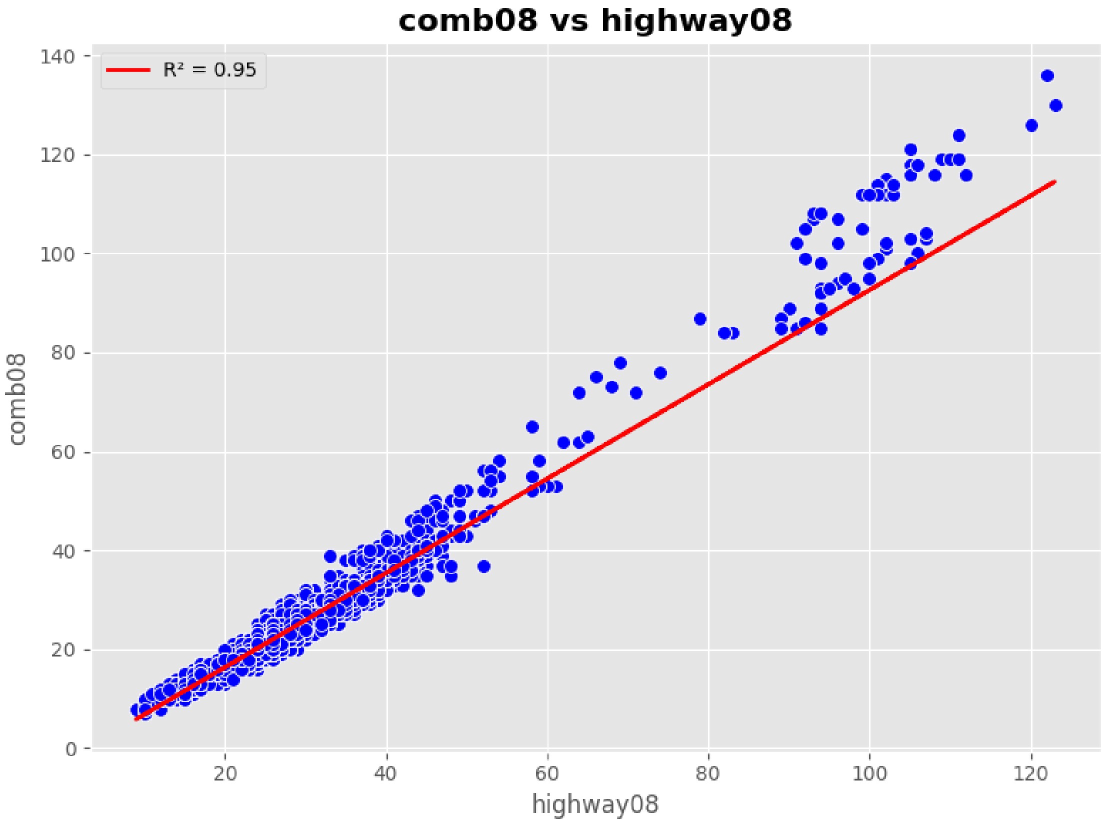
```
[29]: df['fuelType1'].value_counts()
```

```
[29]: fuelType1
      Regular Gasoline     27344
      Premium Gasoline     11267
      Diesel                1142
      Electricity            168
      Midgrade Gasoline      100
      Natural Gas             60
      Name: count, dtype: int64
```

```
[171]: sdf.plot_full_correlation_matrix()
```

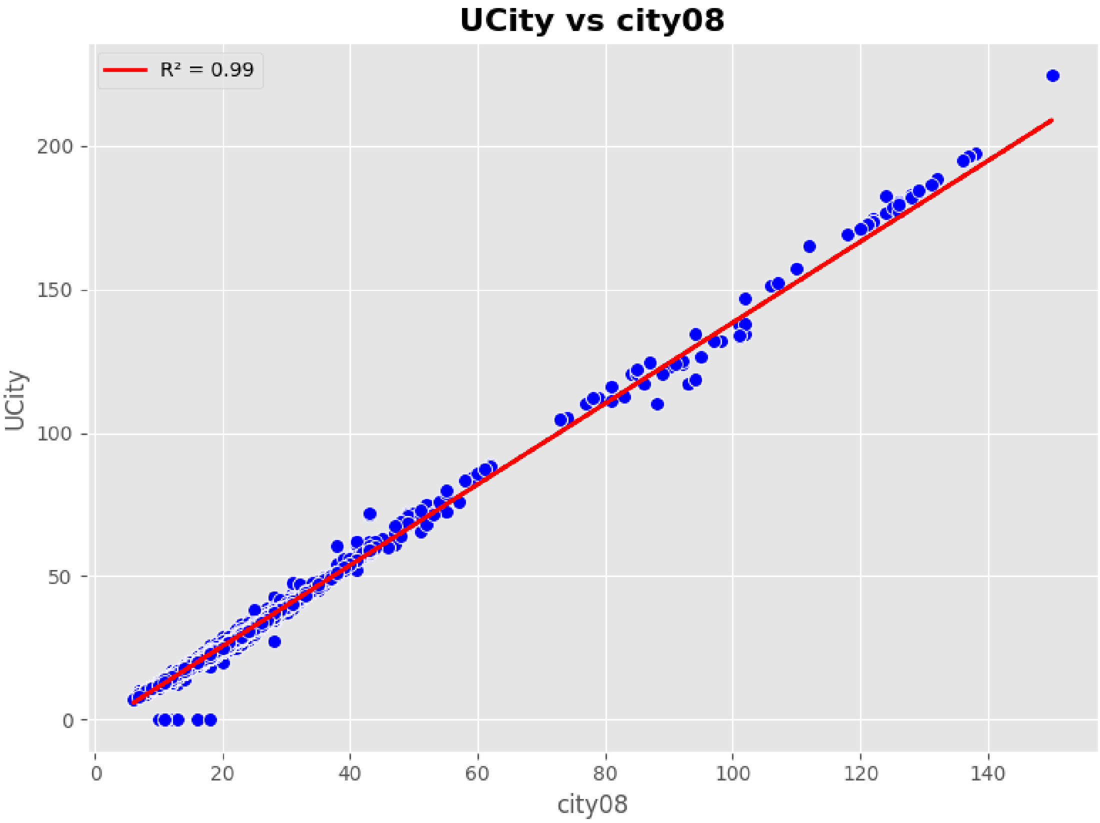


[177]:
```
sdf.plot_and_calculate_r2('highway08', 'comb08')
```

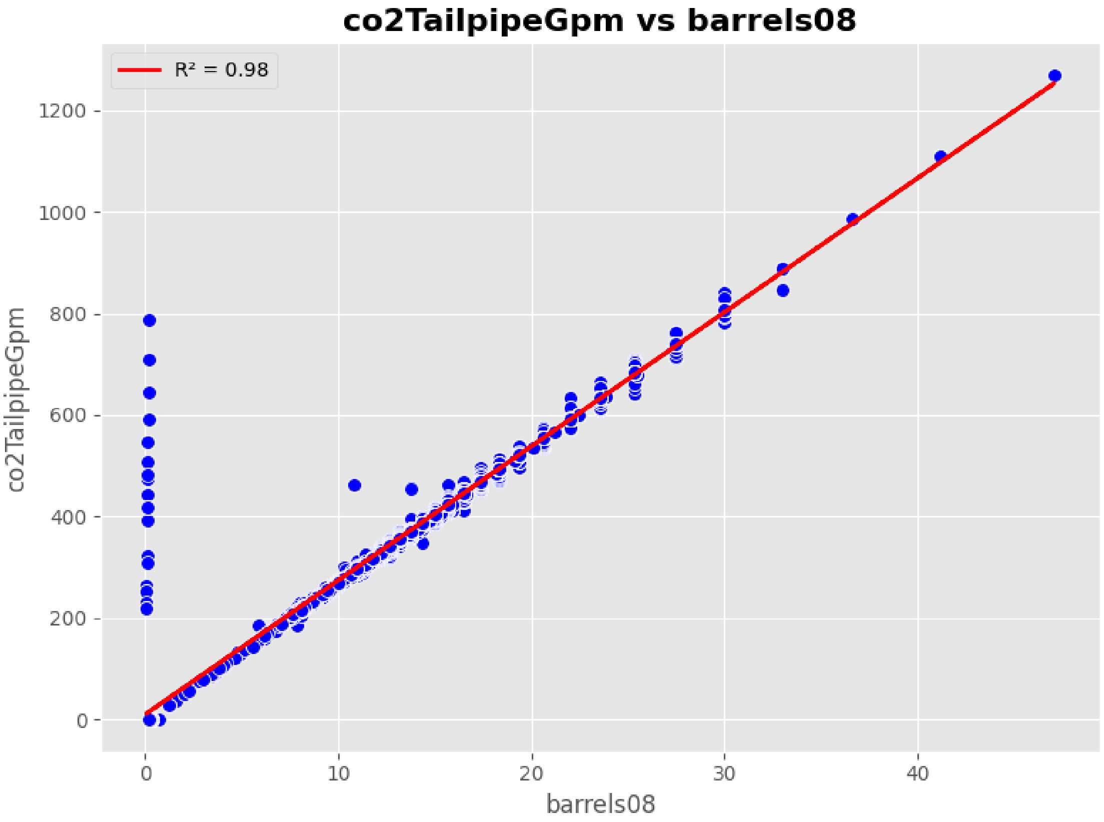


[178]:

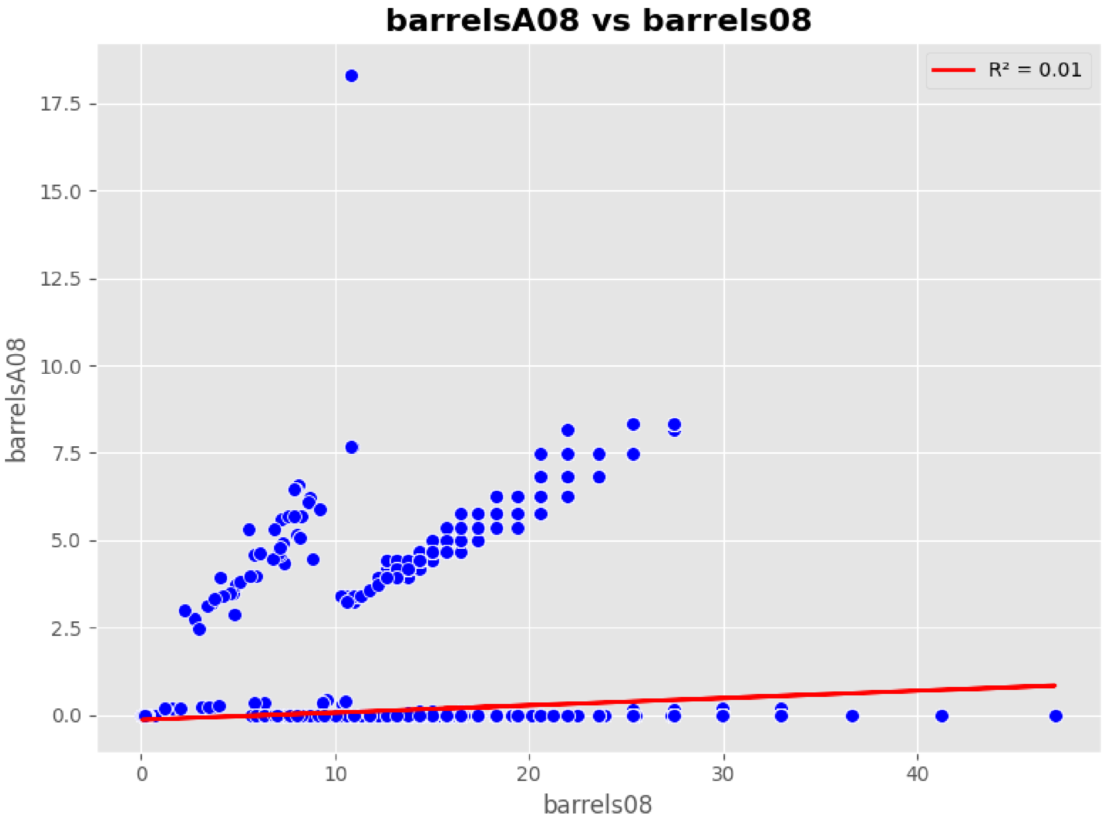
```
sdf.plot_and_calculate_r2('city08', 'UCity')
```



[179]:
```
sdf.plot_and_calculate_r2('barrels08', 'co2TailpipeGpm')
```

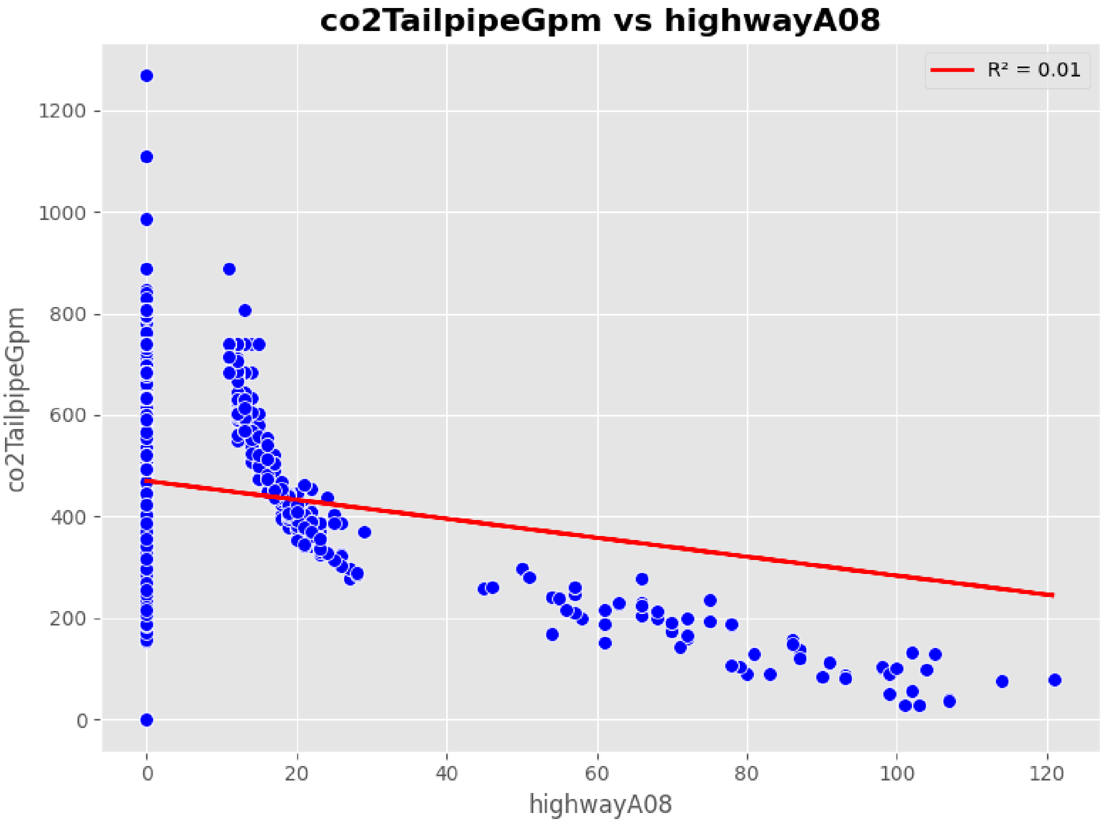


[181]:
```
sdf.plot_and_calculate_r2('barrels08', 'barrelsA08')
```

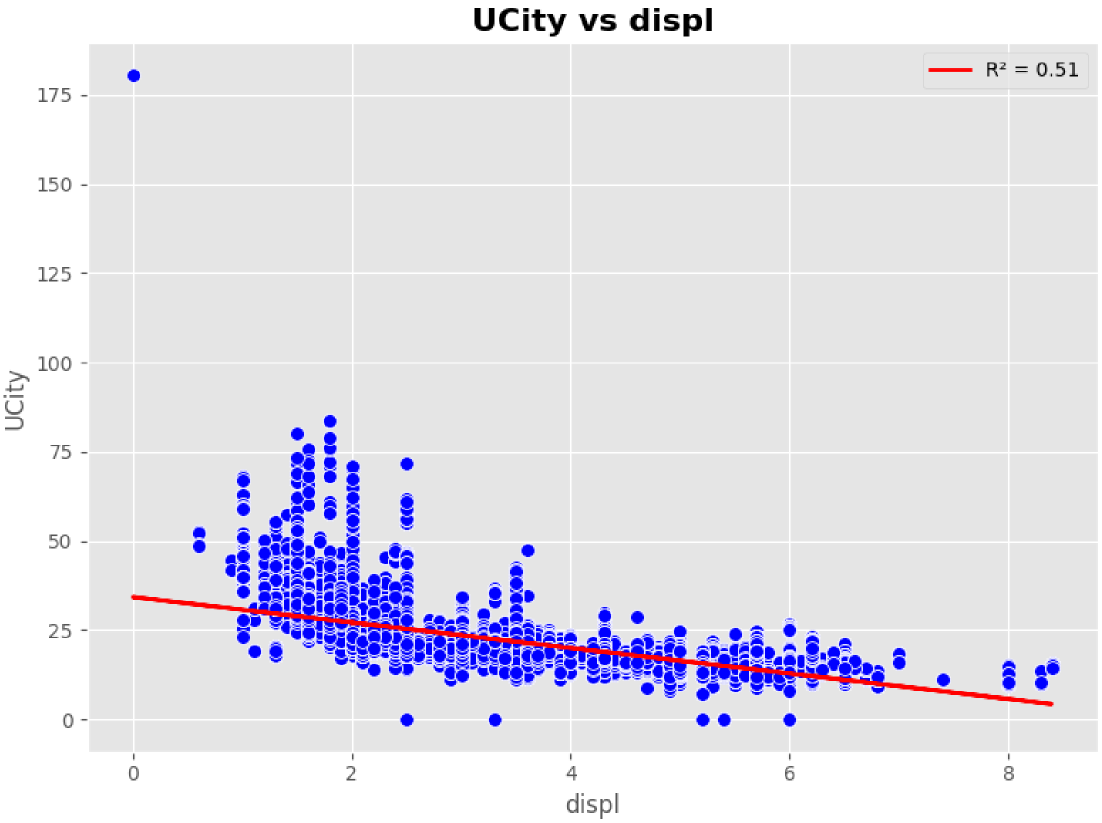


[180]:

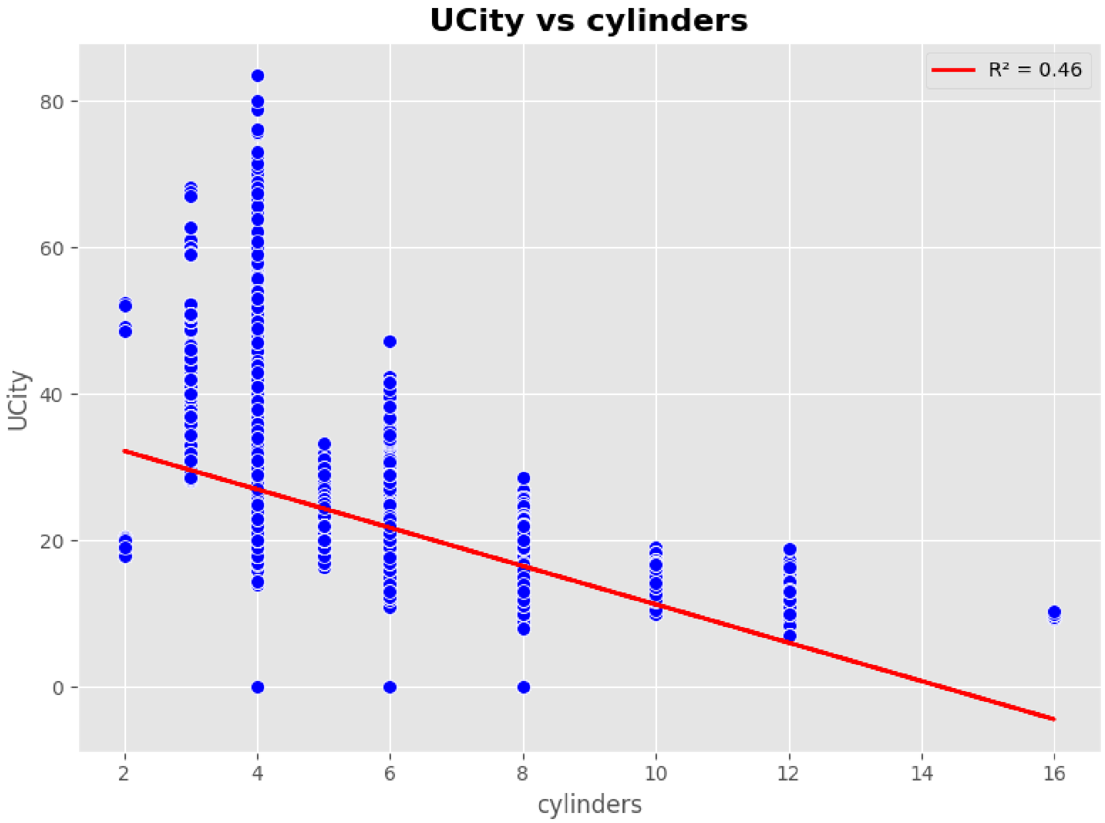
```
sdf.plot_and_calculate_r2('highwayA08', 'co2TailpipeGpm')
```



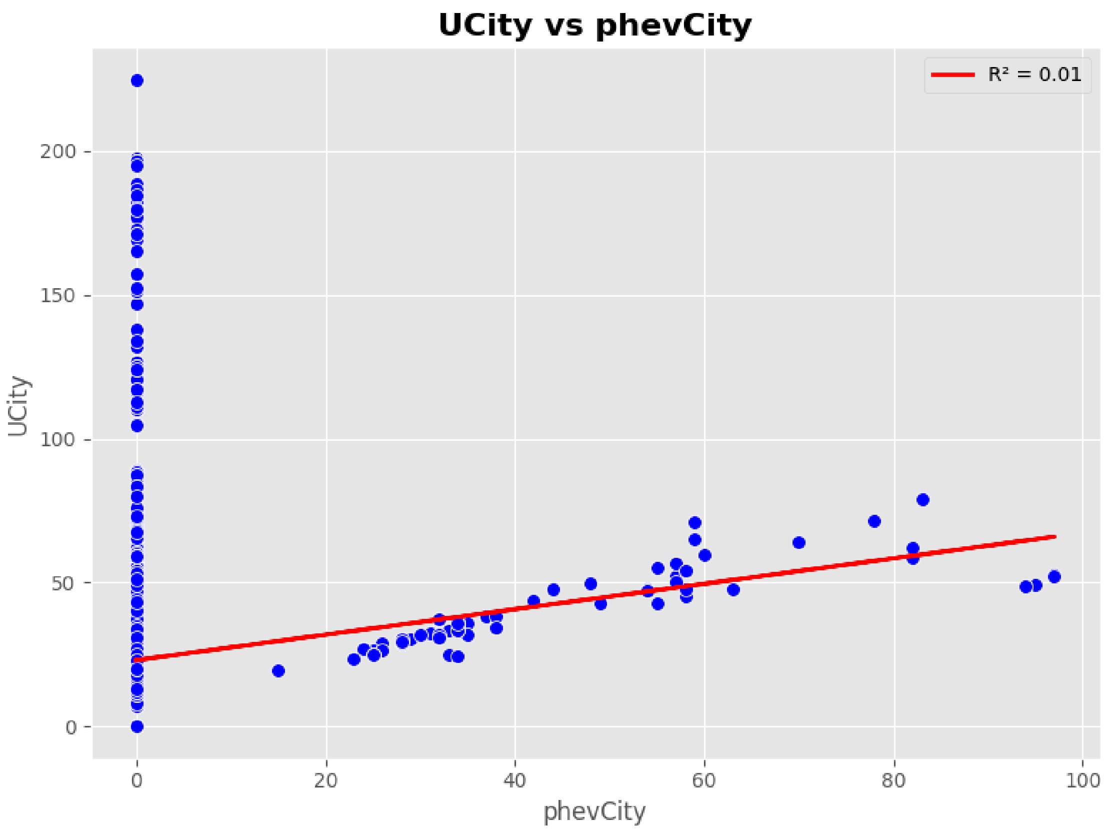
```
[184]: sdf.plot_and_calculate_r2('displ', 'UCity')
```



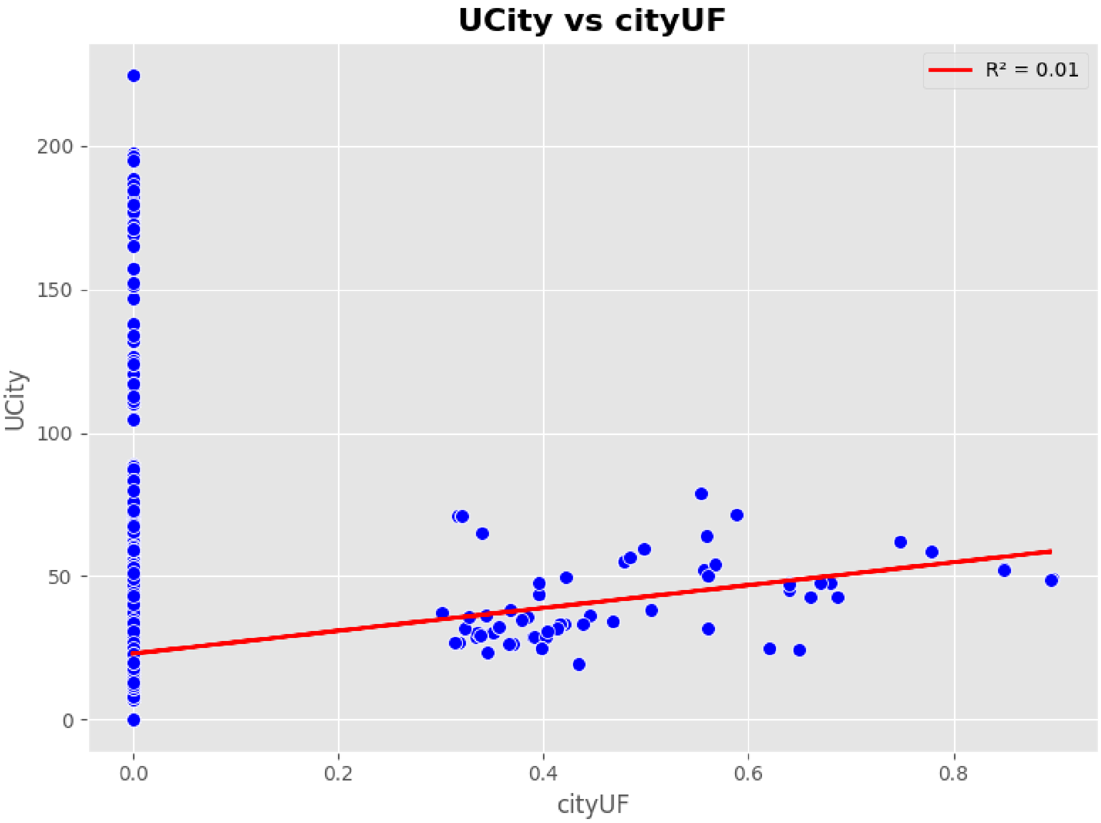
[185]:
```
sdf.plot_and_calculate_r2('cylinders', 'UCity')
```



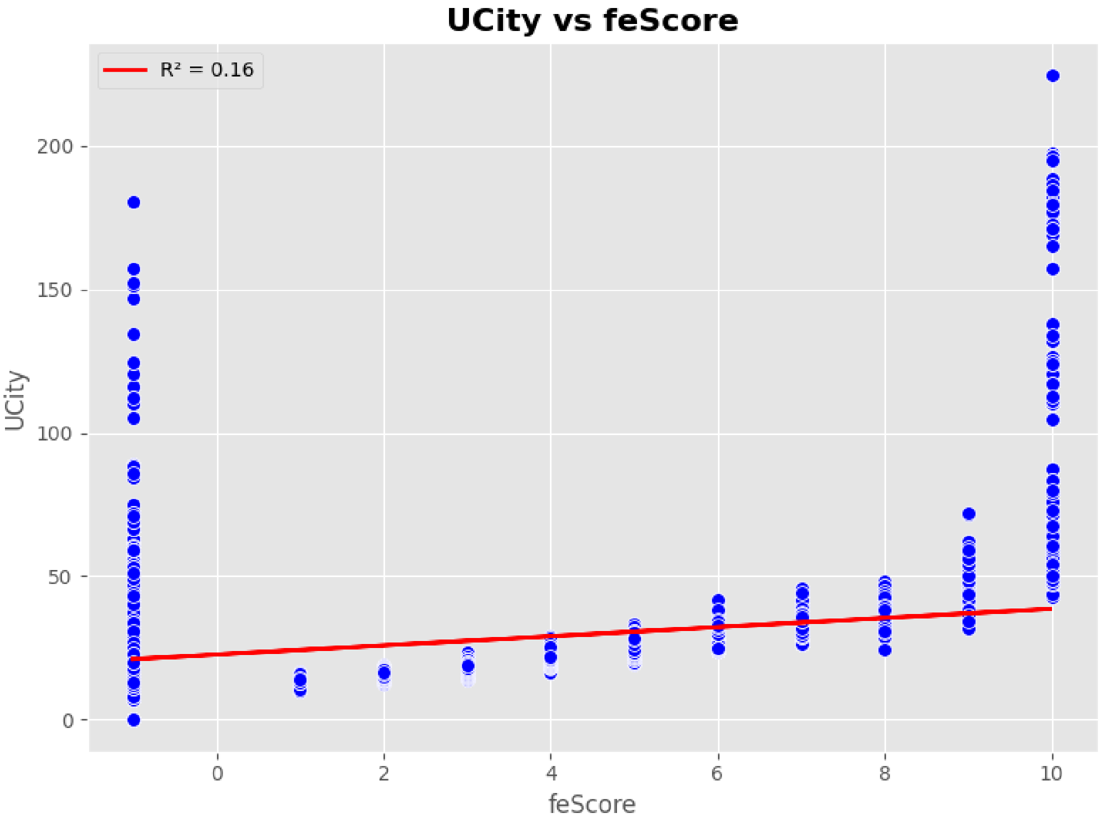
[189]:
```
sdf.plot_and_calculate_r2('phevCity', 'UCity')
```



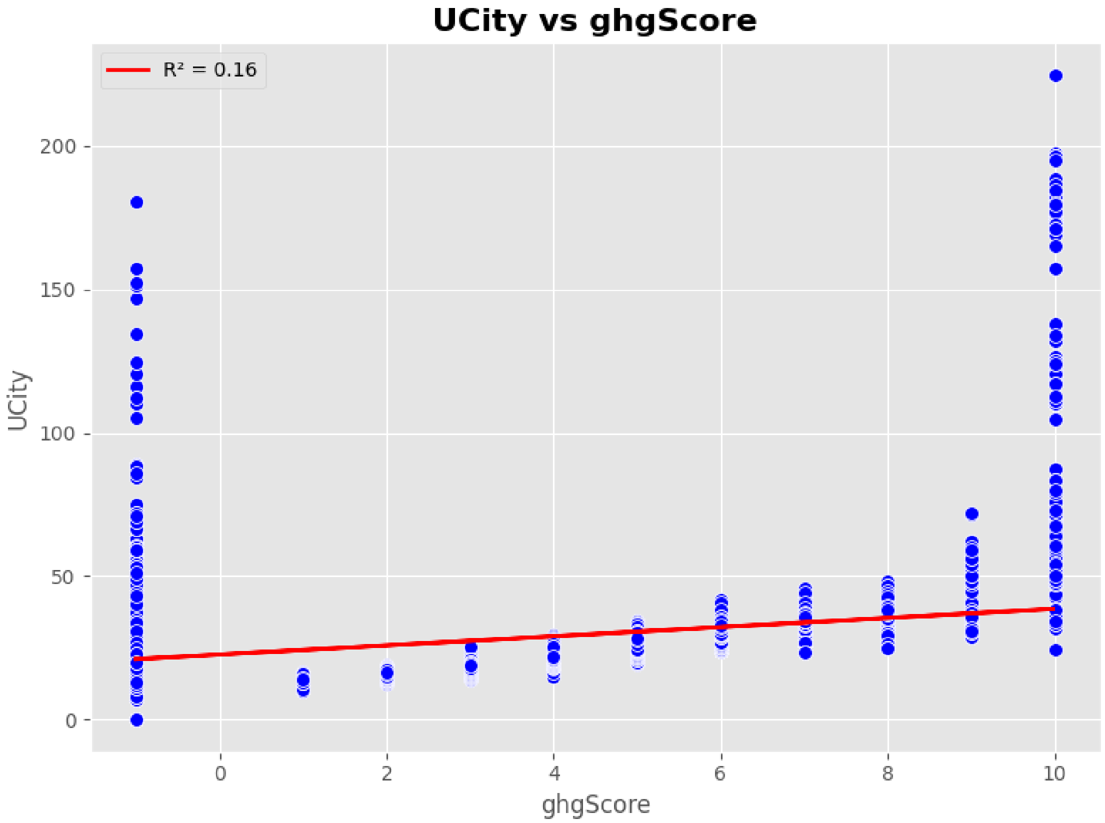
[192]:
```
sdf.plot_and_calculate_r2('cityUF', 'UCity')
```



```
[193]: sdf.plot_and_calculate_r2('feScore', 'UCity')
```



```
[194]: sdf.plot_and_calculate_r2('ghgScore', 'UCity')
```



1.3.1 Analysis of Correlation:

```
[195]: numeric_df = df.select_dtypes(include=['number'])
       corr_matrix = numeric_df.corr()
       ucity_corr = corr_matrix['UCity'].sort_values(ascending=False)
       print(ucity_corr)
```

```
UCity                  1.000000
city08                 0.997167
comb08                 0.983986
highway08              0.925728
UHighway               0.924505
youSaveSpend           0.658371
city08U                0.638491
charge240              0.623996
range                  0.601359
rangeCity              0.593646
comb08U                0.587552
rangeHwy               0.560807
highwayE               0.520141
highway08U             0.516126
```

```
combE              0.504884
cityE              0.490125
charge240b         0.411088
feScore            0.397834
ghgScore           0.397378
hpv                0.256158
hlv                0.229395
id                 0.186948
year               0.171466
pv4                0.156416
lv4                0.122900
phevCity           0.109815
phevComb           0.109099
phevHwy            0.107345
cityUF             0.098911
combinedUF         0.097910
highwayUF          0.096445
cityA08U           0.090935
rangeCityA         0.089419
rangeHwyA          0.088421
combA08U           0.085857
UCityA             0.082741
highwayA08U        0.077327
cityA08            0.074458
combA08            0.066999
UHighwayA          0.061250
co2                0.058152
highwayA08         0.055506
engId              0.039227
lv2                0.018764
pv2                0.017354
cityCD             0.005063
combinedCD         0.002521
highwayCD         -0.000834
ghgScoreA         -0.009474
co2A              -0.032402
barrelsA08        -0.066128
fuelCostA08       -0.081368
co2TailpipeAGpm   -0.082682
fuelCost08        -0.657363
cylinders         -0.679927
barrels08         -0.712469
displ             -0.713249
co2TailpipeGpm    -0.725386
charge120               NaN
Name: UCity, dtype: float64
```

[ ]:

```
[
    'displ',
    'cylinders',
    'fuelType',
    'VClass',
    'drive',
    'highway08',
    'highway08U',
    'city08',
    'city08U',
    'comb08',
    'comb08U',
    'fuelCost08',
    'ghgScore',
    'feScore',
    'range',
    'rangeCity',
    'rangeHwy',
    'co2TailpipeGpm',
    'co2TailpipeAGpm',
    'make',
    'model',
    'charge120',
    'charge240',
    'phevCity',
    'UHighway'
]
```

- The analysis above reproduces the EDA from the provided sample with more than 40 features explored and understood.
- The code is organized, documented, commented, and modular so that it can be reused and repurposed for multiple analyses.
- A class was implemented for the analysis, and several methods added to assist with the EDA.
- Graphs were prepared and separated into their respective sections (Dependent, Independent) for easier readability
- Several new insights were revealed in some of the diagrams above, such as the larger correlation matrix showing some dependencies within the variables, tailpipeGpm and highway, and the distribution of trany. These insights can be useful as we develop the model.
- One interest observation was the feScore and ghgScore which showed an interesting increasing trend and pattern in the data.

1.4 Task 2: The UCity variable is treated as dependent/target variable in the specimen. Discuss your approach to build a predictive model. Is it going to be a classification model or regression model. Why?

- Since the UCity represents the city miles per gallon for a vehicle, we know that this is a continuous numerical value, not a categorical label, therefore we will likely use a regression model to predict this value.

- We can see from the analysis above that some of the independent variables have a linear relationship with UCity.
- This is great since we will want to uphold the assumption of having a linear relationship.
- Our first step is going to be feature selection in which we determine which features the model will need.
- The model will require features that make sense to keep (linear, IID, etc…)
- Next, we will ensure that there is no multicollinearity within those features. We can see that some of the features do have correlations in some of the plots above.
- In addition, we will need to ensure homoscedasticity and normality of residuals.
- With the features determined, next we will clean the data up by removing NANs, missing values, and a few others. We will also need to encode our categorical data to represent it numerically.
- Once cleaned up, we will likely need to scale the datasets values to ensure identical distribution.
- After that, we will select our model of interest. We can start with a simple linear regression model, then move on to some others such as ridge regression and afew others we will learn in our course.
- We will then split the data into training and testing sets, and train the model using the training data to ensure that it is generalizing well.
- We will follow that up by testing the model on the testing set and evaluating its metrics. Normally we would use Accuracy, Recall, Precision, F1, and AUC ROC, however since this is a regression we can evaluate the model using MSE, RMSE, and R2 to check how well the model predicts UCity on unseen data.
-

1.5 Task 3: Discuss which variables you will not consider as inputs to the model. Why? (20 points)

- There will be a few variables that we will not use by default such as the ID numbers, the createdOn and manufacturedOn dates„ the manufacturing codes, model, engineId, end_desc, guzzler, youSaveSpend, and a few others. These variables, which in some cases may have some strange correlations with our dependent variable, do not in reality have an impact on the model and will not allow use to generalize on new data. Therefore we will ignore those.
- To avoid multicollinearity, we will remove any highly correlated independent variables. For example, city08 and city08U are highly correlated.
- We will also discard features with too many missing or NAN values such as evMotor and a few others.
- We will also want to remove any that violate our assumptions of linearity, IID, homoscedasticity, normality of residuals, and lack of multicollinearity.
- Will will also check into categories like make and model, and any others that have high cardinality to remove those as well.

1.6 Task 4: How will you evaluate your model to avoid over-fitting/under-fitting. (20 points)

- We can use metrics to evaluate the model such as MSE, RMSE, and R2. This will allow us to determine if the model is generalizing or not.
- We will need to make sure that when you split the data into training and testing, that the

testing data is used to evaluate the models performance.

- The use of train test split and both cross validation or nested cross validation (as needed) will be implemented to ensure generalizability.
- We can also use learning curves, as in plots of the training error and validation error as the size of the training data increases to determine if it is overfitting or underfitting.
- Finally, the use of lasso (L1) and ridge (L2) regularization to add a penalty to large coefficients, will force the model to prioritize the simpler solutions that are less likely to overfit the training data, giving us a better understanding of the performance.

[ ]: