

# Documentation

---

## 1. Data Understanding and Representation

### a. Data Loading and Initial Exploration

- Loaded the dataset from 'audi.csv' using Pandas.
- Checked the shape of the dataset using `print(df.shape)`.
- Displayed the first few rows of the dataset using `df.head()`.

### b. Data Cleaning and Preprocessing

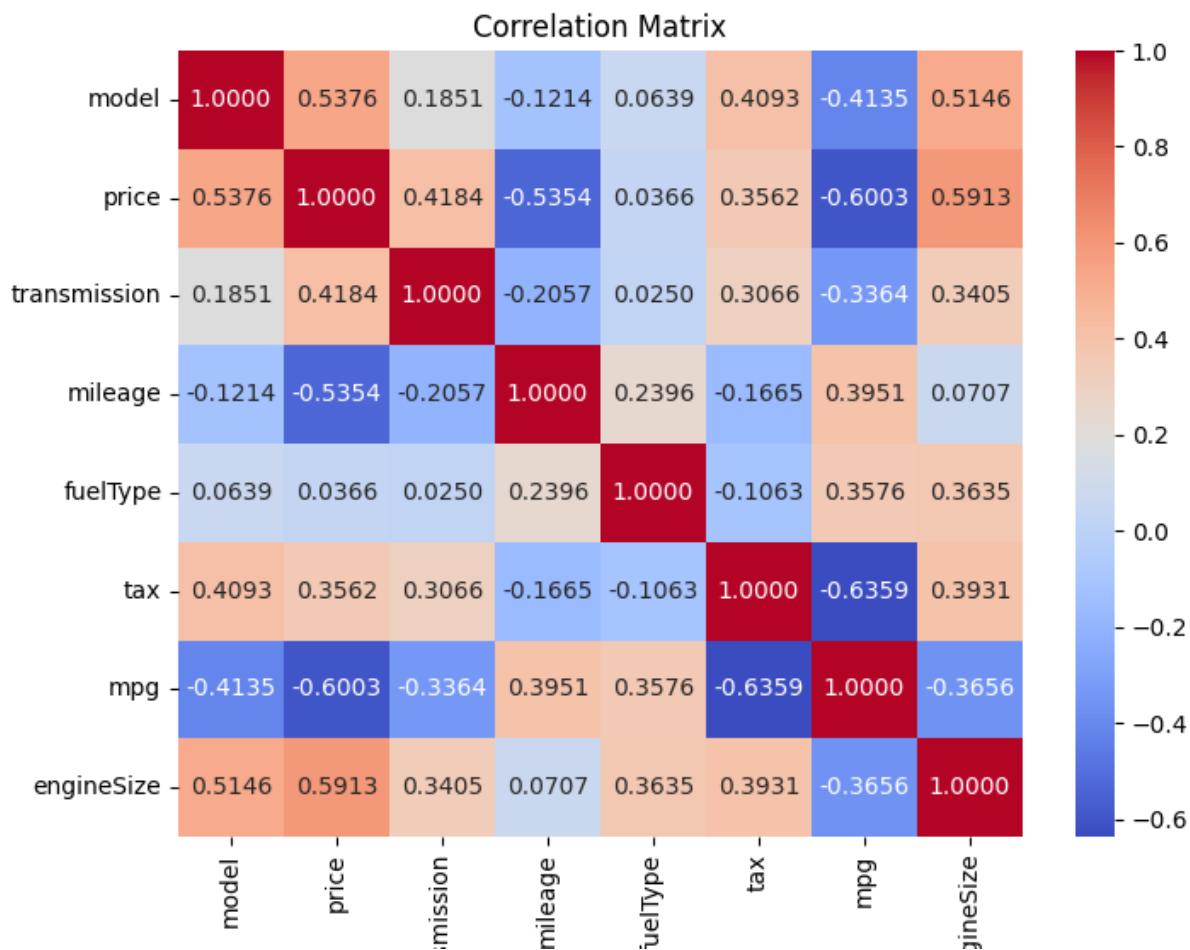
- Dropped the 'year' column as it was considered unnecessary.
- Checked for missing values using `print(df.isna().sum())` and dropped rows with missing values.
- Mapped categorical columns ('model', 'transmission', 'fuelType') to numerical values using a mapping function (`map_columns`).
- Standardized numerical columns (excluding mapped and target columns) to have zero mean and unit variance.

## 2. Implementing PCA using Covariance Matrices

### a. Covariance Matrix:

- Computed the covariance matrix, `covMatrix`, for the standardized dataset using `df.cov()`. This matrix provides insights into the relationships and variances between different features.
- Visualized the covariance matrix using a heatmap to highlight the strength and direction of the linear relationships between pairs of features. The heatmap was generated with `sns.heatmap()` function.
- The heatmap allows us to identify patterns and understand how changes in one variable relate to changes in others.

### b. Correlation Matrix:



- Computed the correlation matrix, `corrMatrix`, using `df.corr()`. This matrix provides information about the linear relationship and the strength of association between pairs of variables, normalized to a range of -1 to 1.
- Visualized the correlation matrix using a heatmap to better understand the correlation patterns between features. This heatmap was created with `sns.heatmap()` function.
- The correlation heatmap aids in identifying features that may be highly correlated, which is essential for understanding potential multicollinearity issues in the dataset.

These visualizations are crucial for an initial exploration of the dataset, helping to identify patterns, relationships, and potential issues that might affect subsequent analyses.

### 3. Eigenvalue and Eigenvector Calculation

#### a. Eigenvalues and Eigenvectors:

- Computed the eigenvalues and eigenvectors using the covariance matrix of the standardized dataset: `eigenvalues, eigenvectors = np.linalg.eig(df.cov())`.
- The eigenvalues represent the variance captured by each principal component, and the eigenvectors represent the direction of these components.
- Sorting the eigenvalues and their corresponding eigenvectors in descending order allows us to prioritize the principal components with the most variance, providing a basis for dimensionality reduction.

## b. Significance of Eigenvalues:

- Eigenvalues quantify the amount of variance captured by each principal component. Larger eigenvalues indicate that the corresponding principal component retains more information from the original data.
- Sorting eigenvalues enables the identification of the principal components that contribute the most to the dataset's overall variance.

## c. Sorting Eigenpairs:

- Sorted the eigenvalue-eigenvector pairs based on eigenvalues in descending order. This ensures that the principal components are arranged by their significance in capturing variance.

## d. Dimensionality Reduction:

- Principal components with higher eigenvalues are more informative, and retaining them allows for effective dimensionality reduction while preserving most of the dataset's variance.

These calculations are fundamental for understanding the distribution of variance in the dataset and determining the most influential components that will contribute to the principal components.

## 4. Explained Variance and Cumulative Explained Variance

### a. Explained Variance Calculation:

- Calculated the explained variance for each principal component using the formula:

$$\text{Explained Variance} = \text{EigenvalueOfComponent} / \text{SumOfAllEigenvalues}$$

- This provides insight into the proportion of total variance captured by each principal component.

### b. Cumulative Explained Variance:

- Computed the cumulative explained variance by summing the explained variance values sequentially.
- The cumulative explained variance illustrates the cumulative proportion of total variance explained as more principal components are included.

### c. Importance of Cumulative Explained Variance:

- Monitoring the cumulative explained variance is crucial for deciding the optimal number of principal components to retain.
- Identifying the point where cumulative explained variance reaches a satisfactory threshold (e.g., 90%) assists in selecting an appropriate number of components for dimensionality reduction.

### d. Threshold for Variance Retention:

- Setting a threshold (e.g., 90% variance retention) guides the determination of the minimum number of principal components needed to retain a significant amount of information from the original dataset.

### e. Visualization for Decision Making:

- Plotted the cumulative explained variance against the number of components to visually assess the trade-off between dimensionality reduction and retained information.

This step is essential for understanding how well the chosen number of principal components captures the variability in the data, enabling informed decisions about model complexity and efficiency.

## 5. Solving for Principal Components

### a. Selecting the Number of Principal Components (k=3):

- Chose a specific number of principal components based on the task requirements and the desired dimensionality reduction. In this case, (  $k = 3$  ) principal components were selected.

### b. Sorting Eigenvalues and Eigenvectors:

- Sorted the eigenvalues and corresponding eigenvectors in descending order. This step is crucial as it determines the order of importance of each principal component in capturing the variability of the original data.

### c. Extracting Selected Eigenvectors and Eigenvalues:

- Extracted the (  $k$  ) eigenvectors and eigenvalues corresponding to the chosen number of principal components. These selected components represent the transformed feature space that retains the most critical information from the original dataset.

### d. Significance of Selection:

- The careful selection of the number of principal components is pivotal in achieving an optimal balance between dimensionality reduction and information retention. Too few components may result in information loss, while too many may lead to excessive complexity.

### e. Visualization of Chosen Components:

- The chosen eigenvectors and eigenvalues provide insights into the directions and magnitudes of maximum variance in the dataset. Visualization techniques, such as biplots, can further enhance the understanding of the relationships between features and principal components.

## 6. Sequential Variance Increase

### a.Calculating Variance Captured:

- Computed the variance captured by each principal component individually. The variance reflects the amount of information each component holds about the original data.

### b.Visualizing Variance Increase:

- Created a visualization to illustrate the increase in variance as additional principal components are considered. This visual representation is valuable for understanding how each component contributes to the overall variability of the dataset.

### c.Interpretation of Variance Trends:

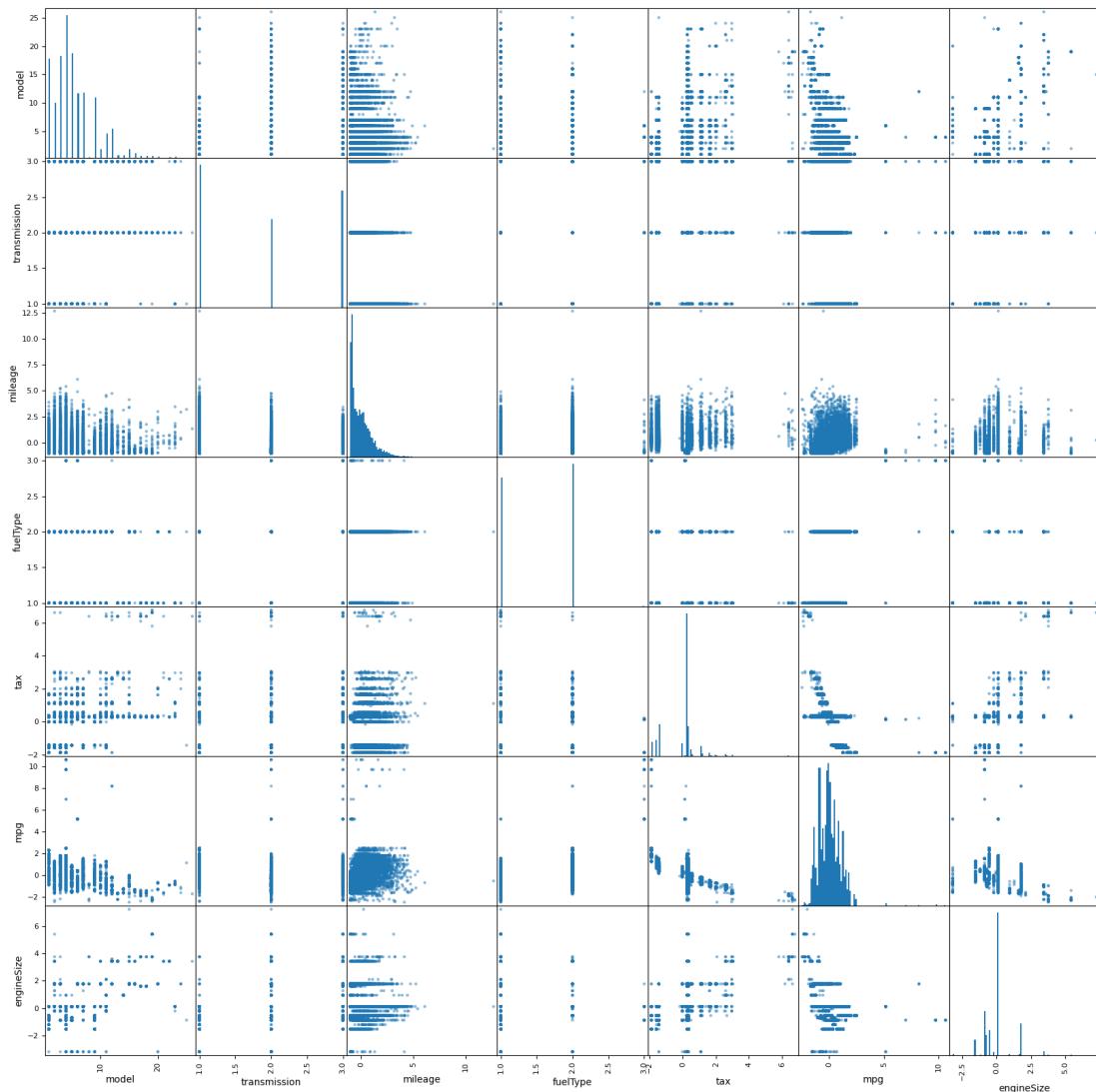
- Analyzed the trends in variance increase to identify the point of diminishing returns. This information aids in determining the optimal number of principal components needed to capture a significant amount of variability while avoiding redundancy.

#### **d. Decision Criteria:**

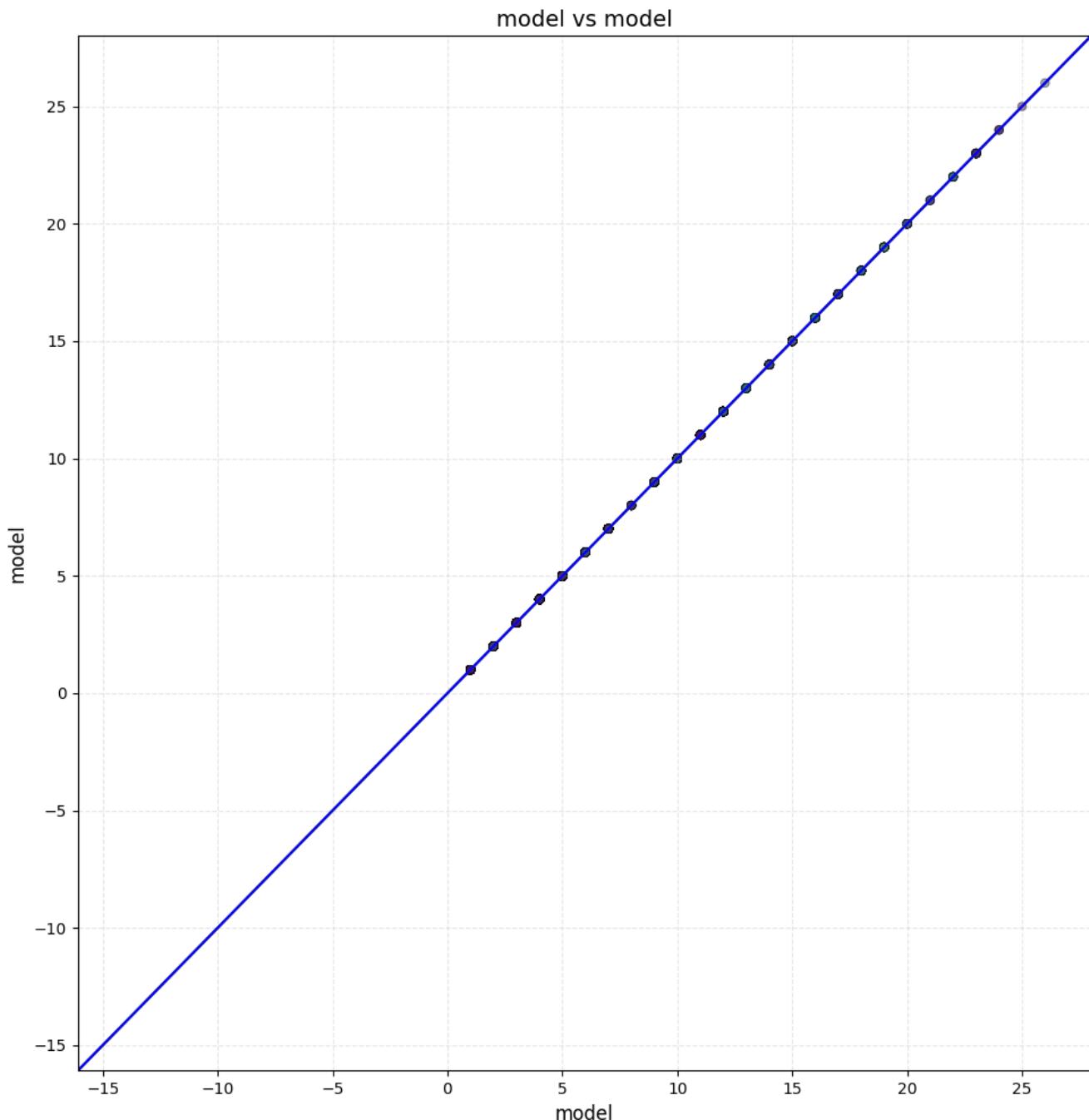
- The sequential variance increase analysis provides decision-making criteria for selecting an appropriate number of principal components. A common criterion is often to choose the number of components that collectively capture a high percentage (e.g., 90%) of the total variance.

## **7. Visualization using Pair Plots**

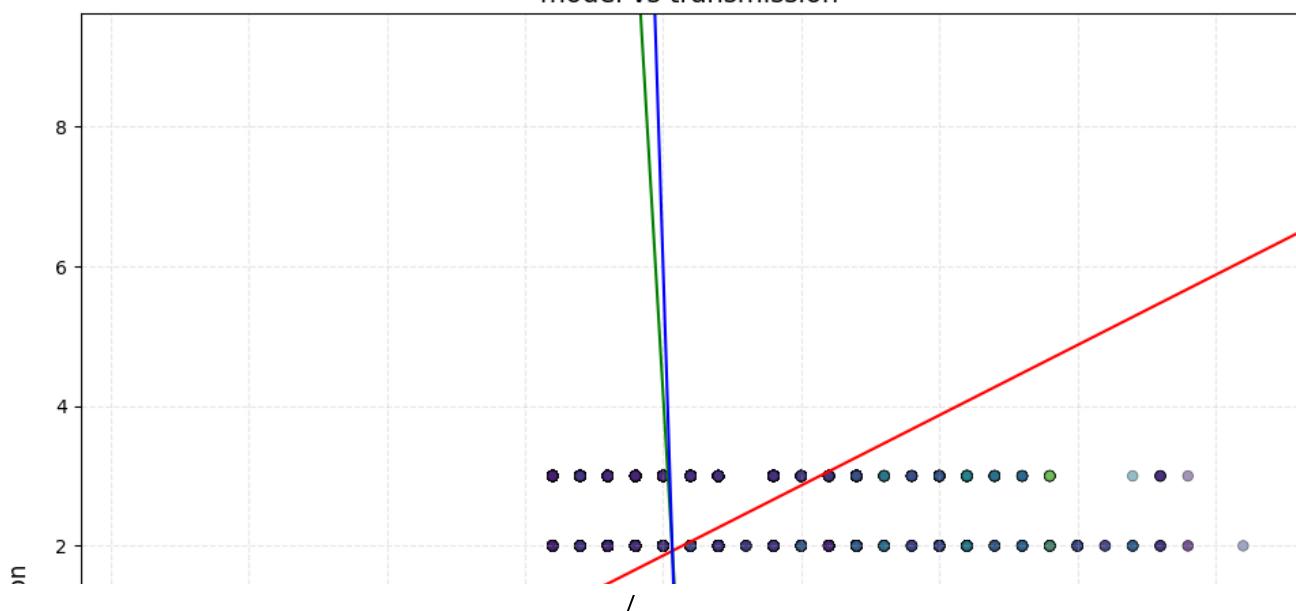
#### **a. Creating Pair Plots:**

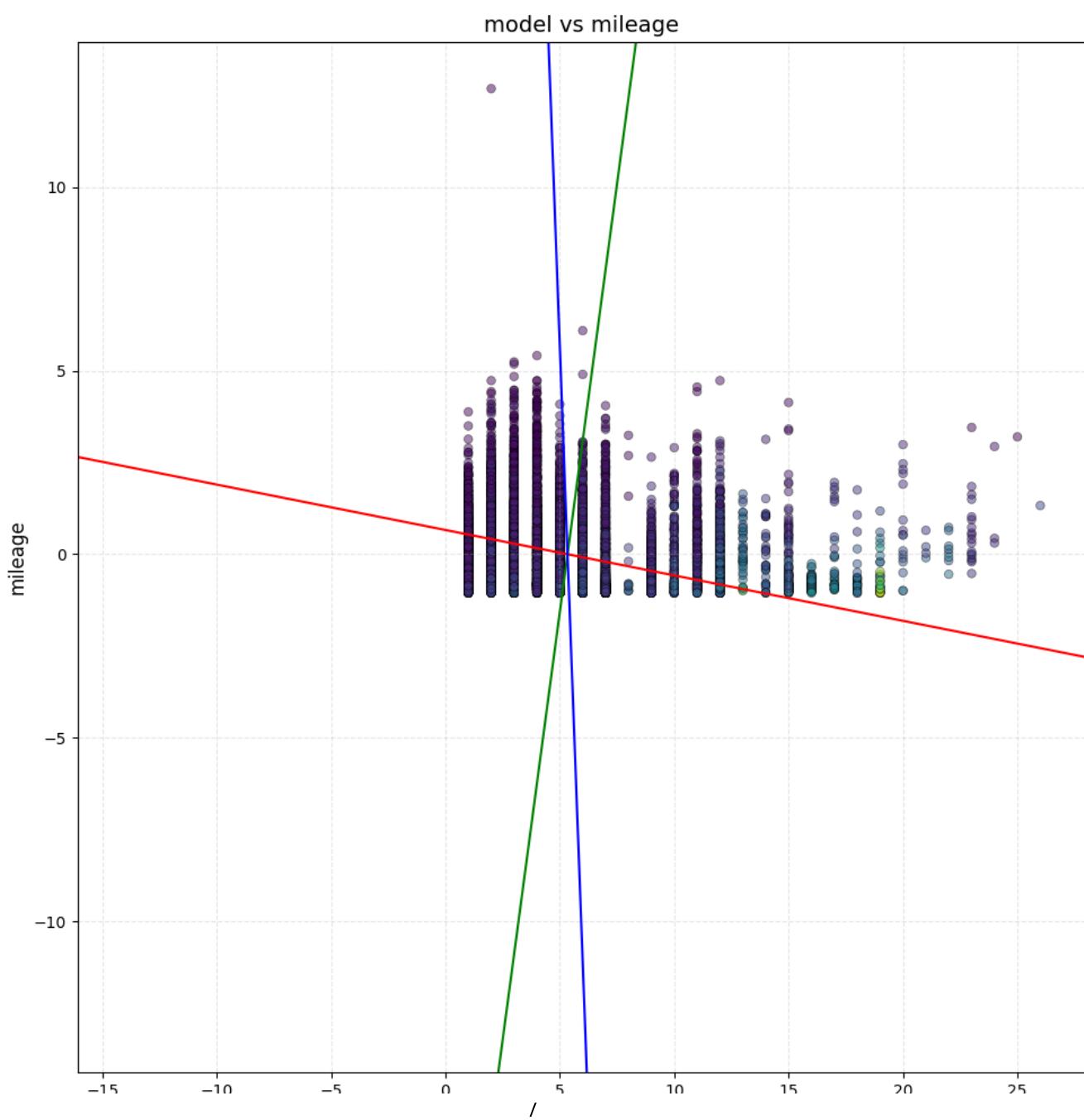
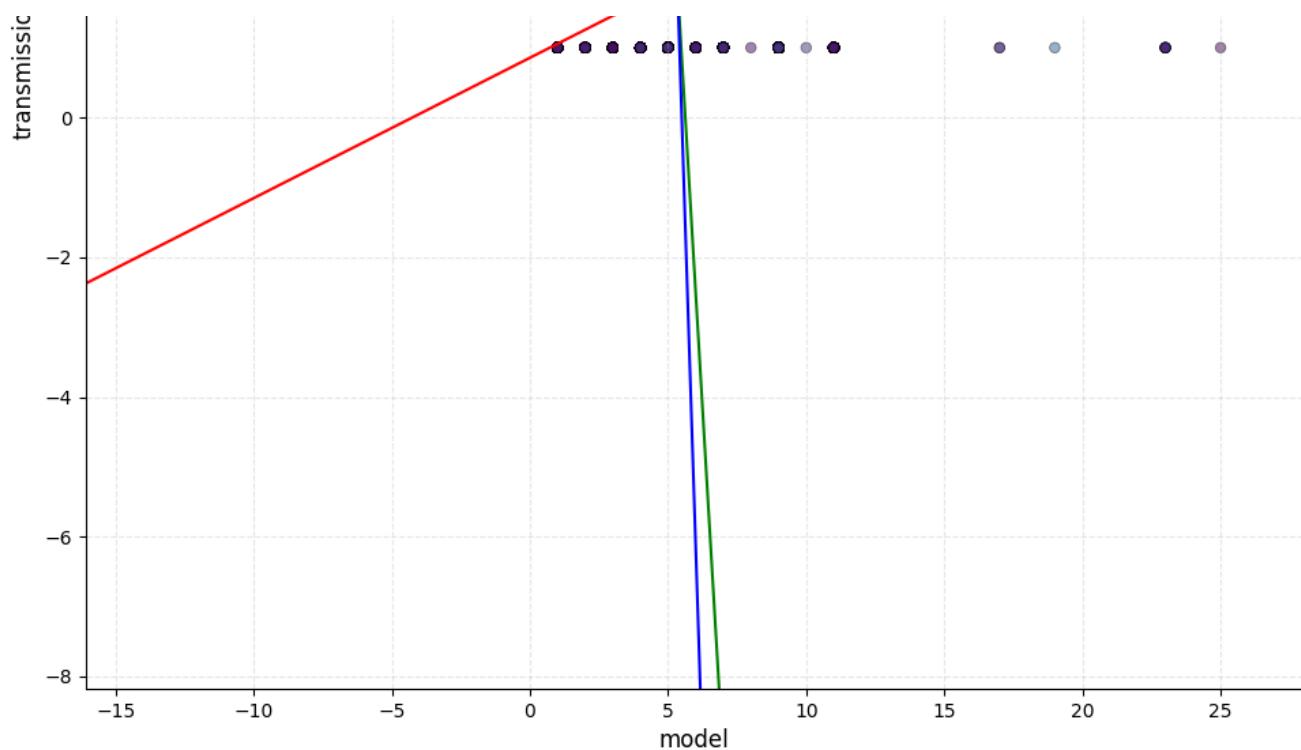


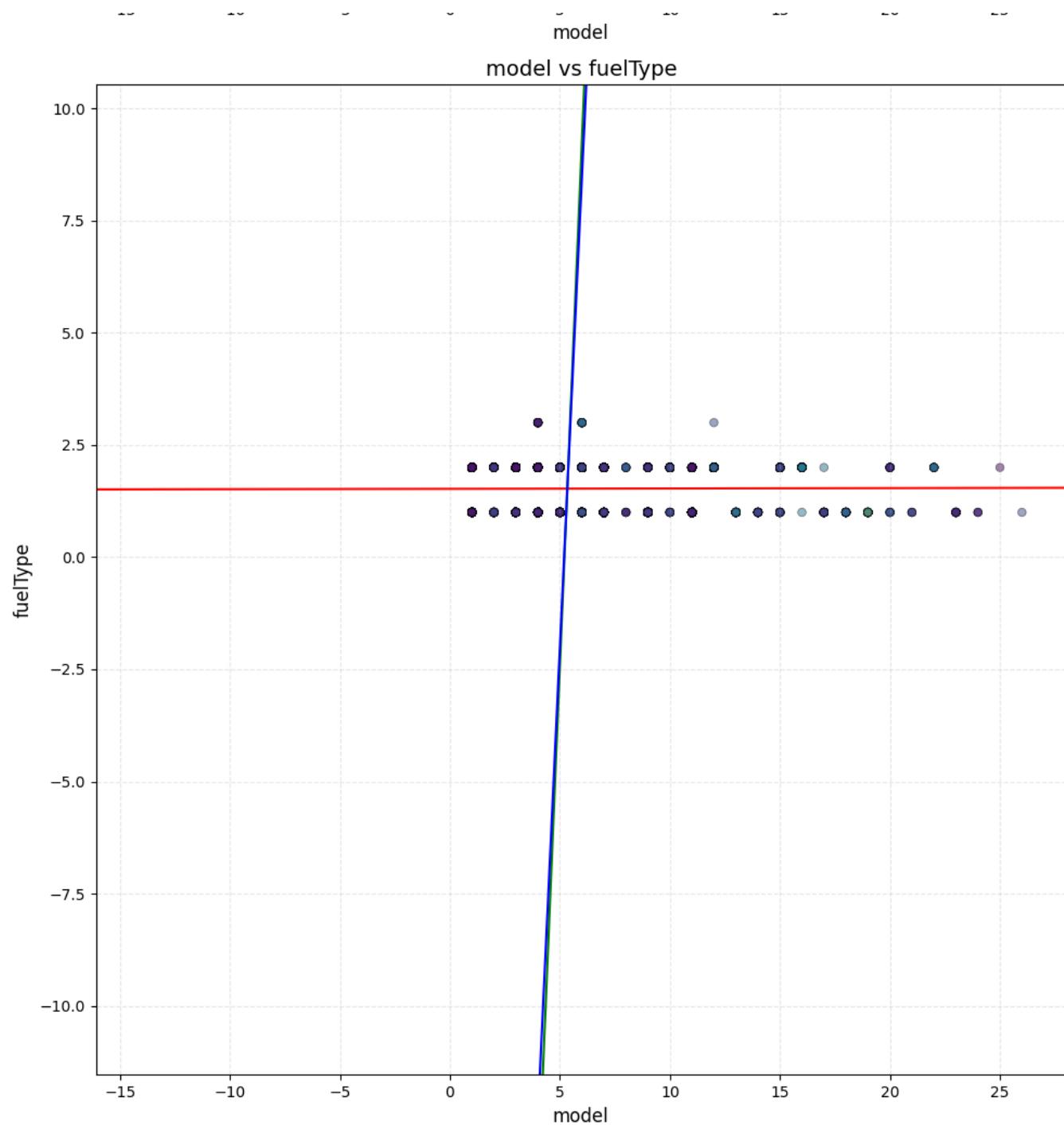
- Generated pair plots to visually inspect the relationships between different features in the dataset. These plots offer insights into potential patterns, correlations, and distributions among variables.

**b. Projection of Principal Components:**

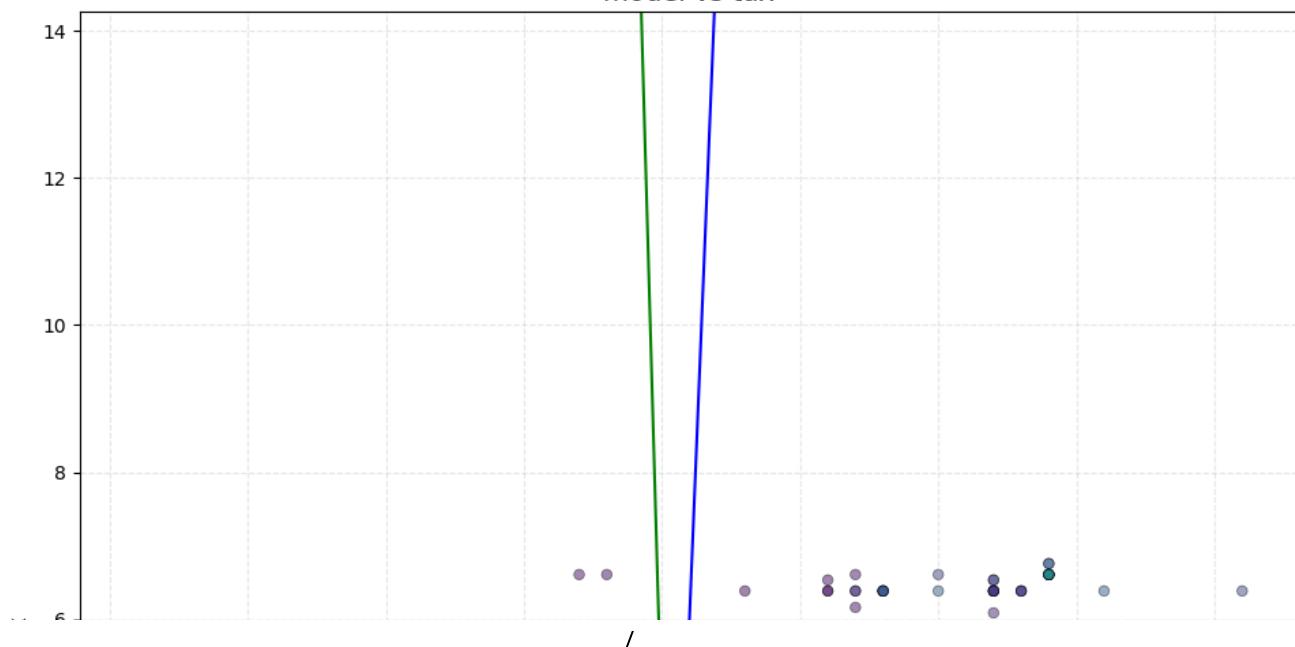
model vs transmission

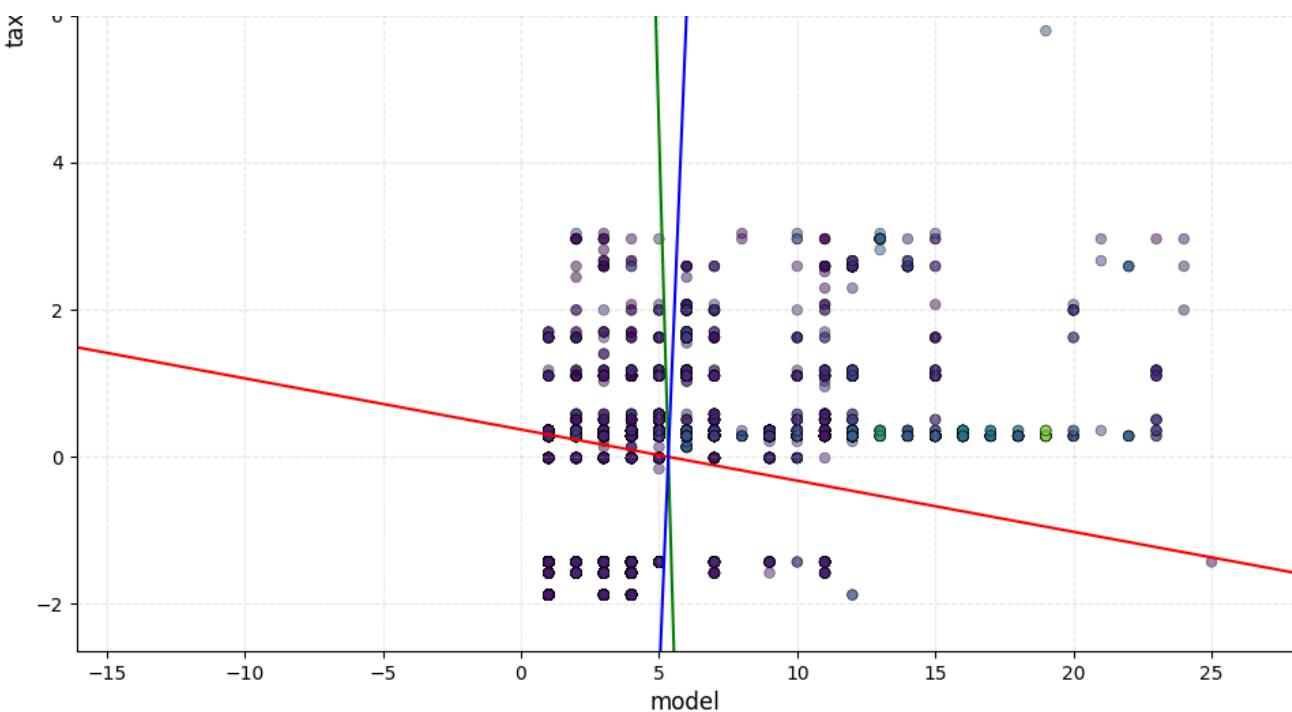




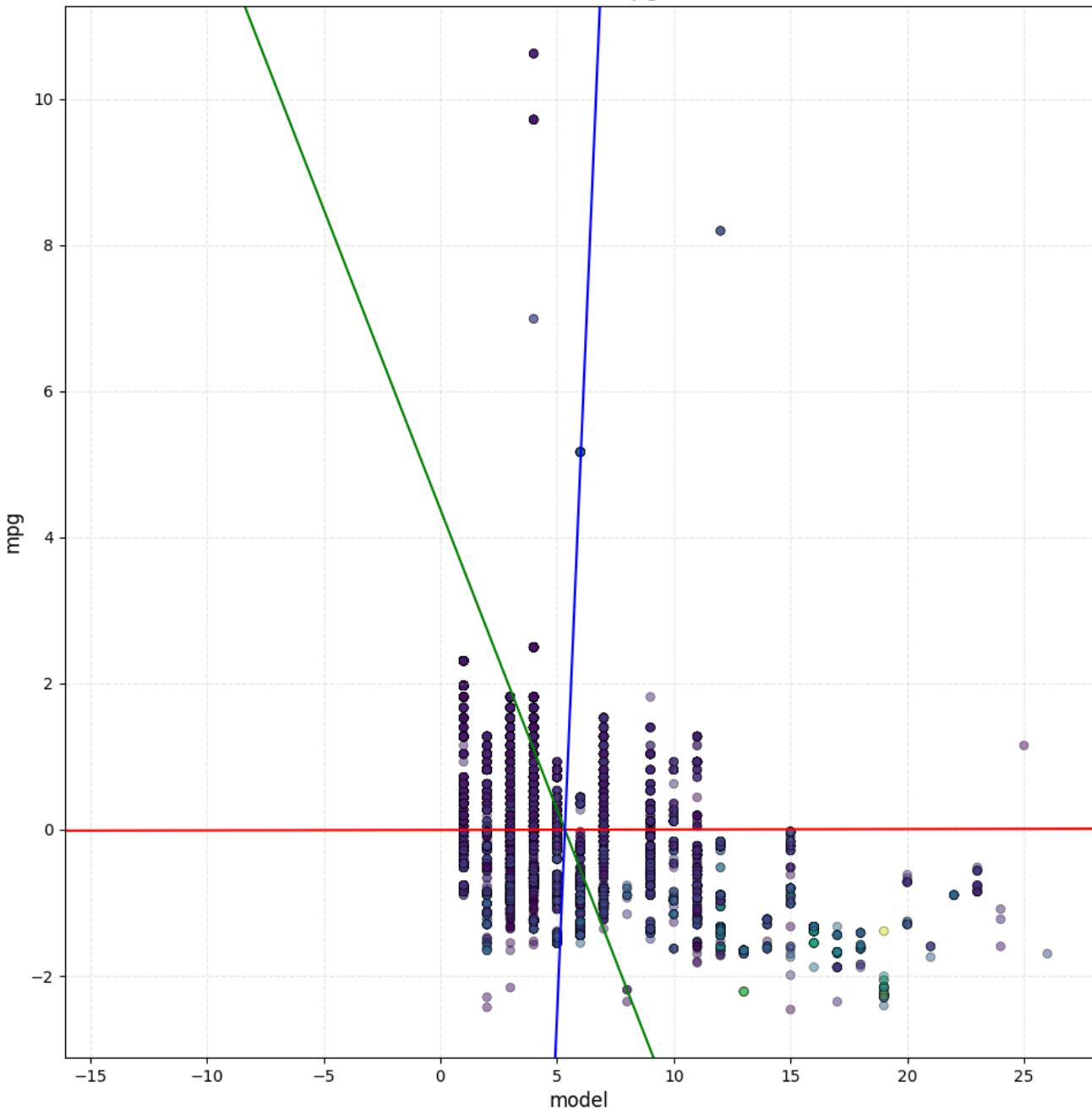


model vs tax

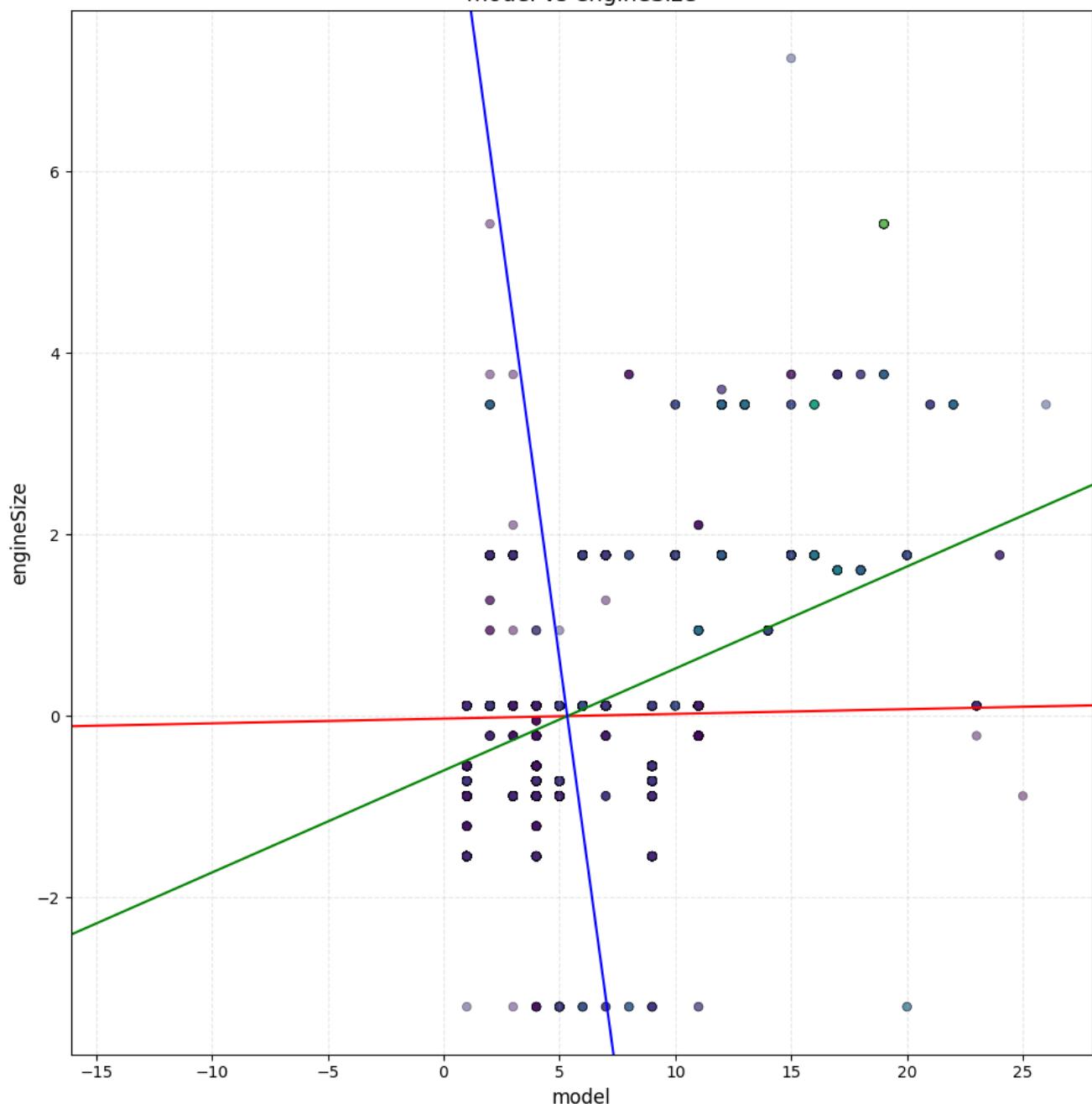




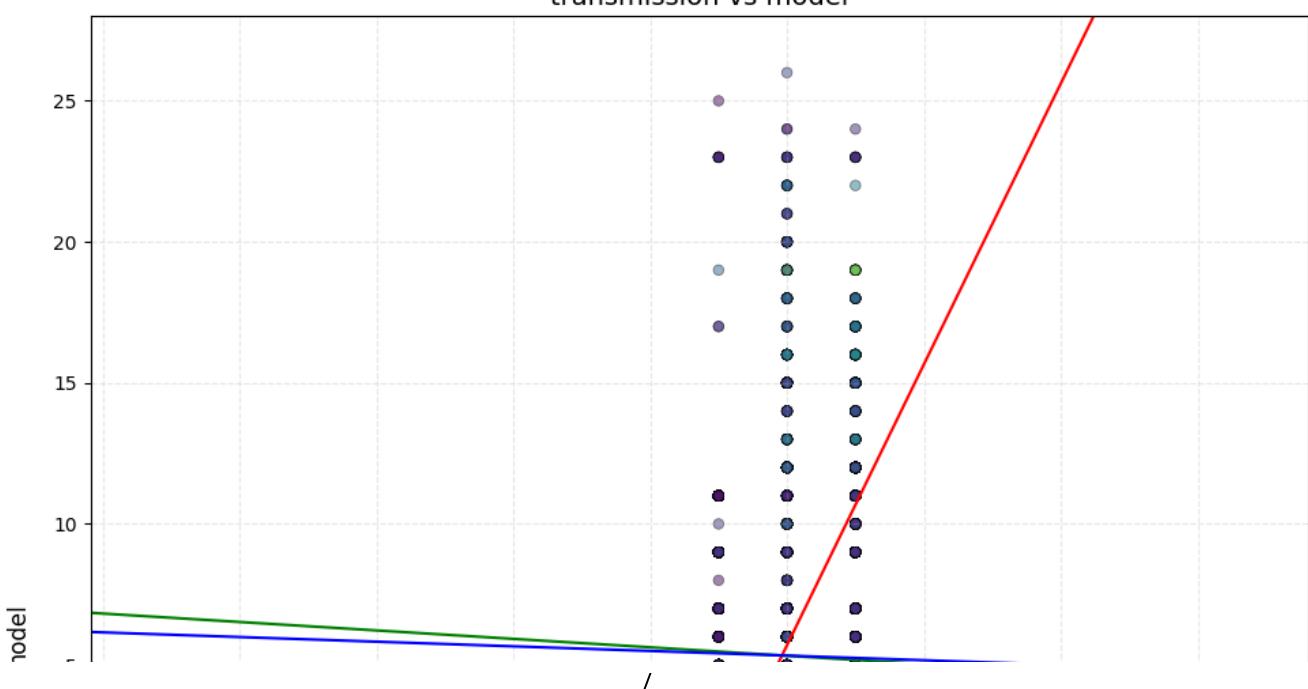
model vs mpg

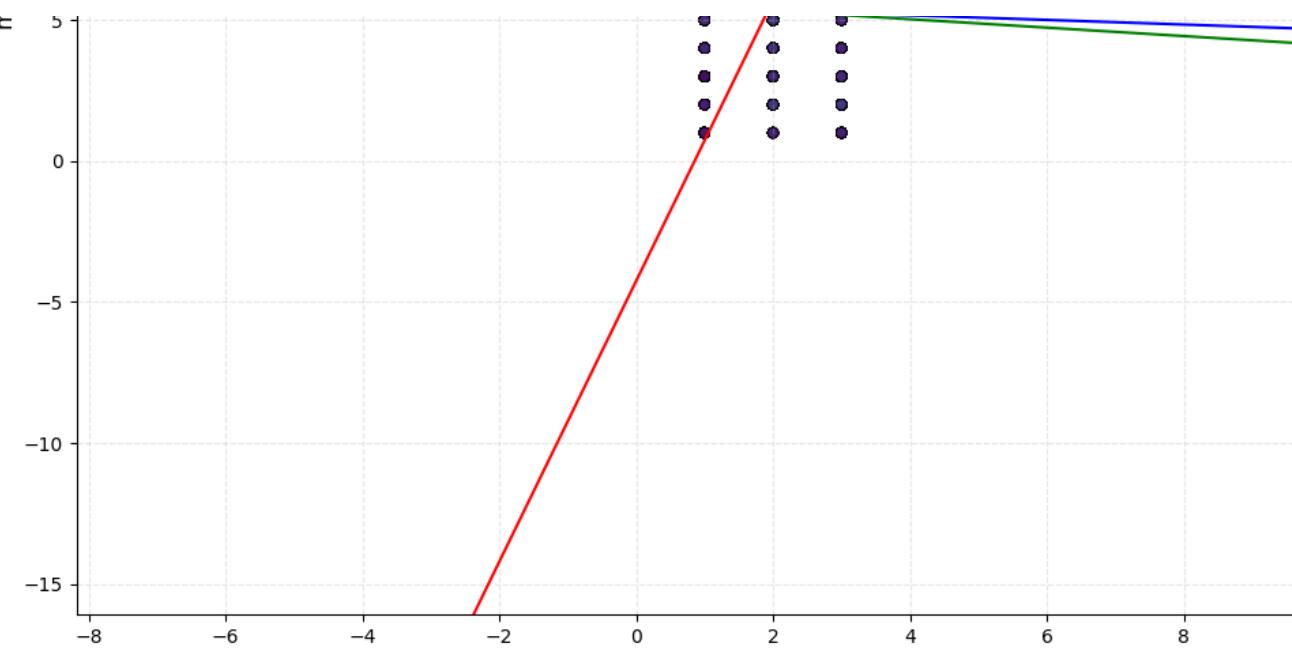


model vs engineSize

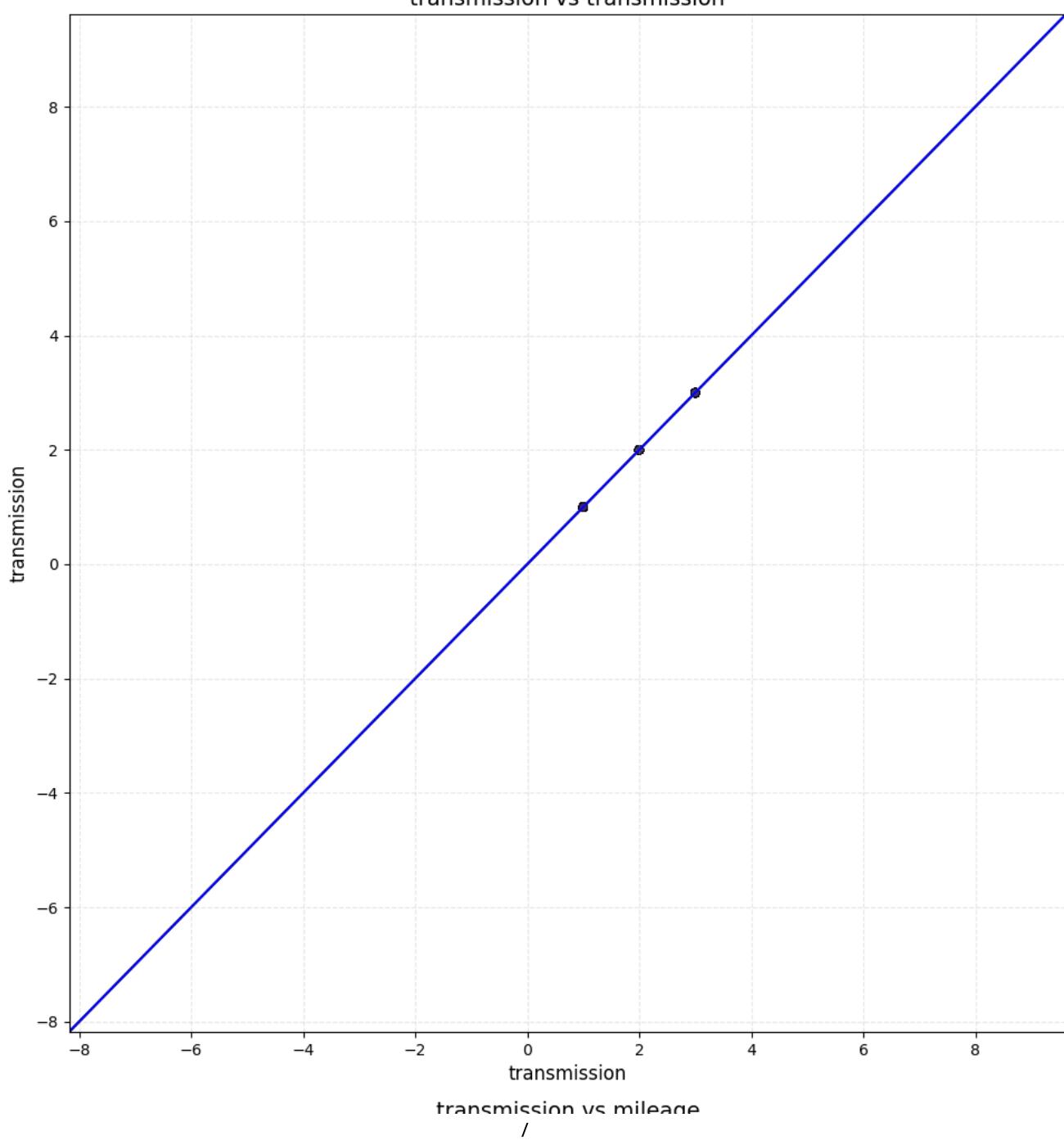


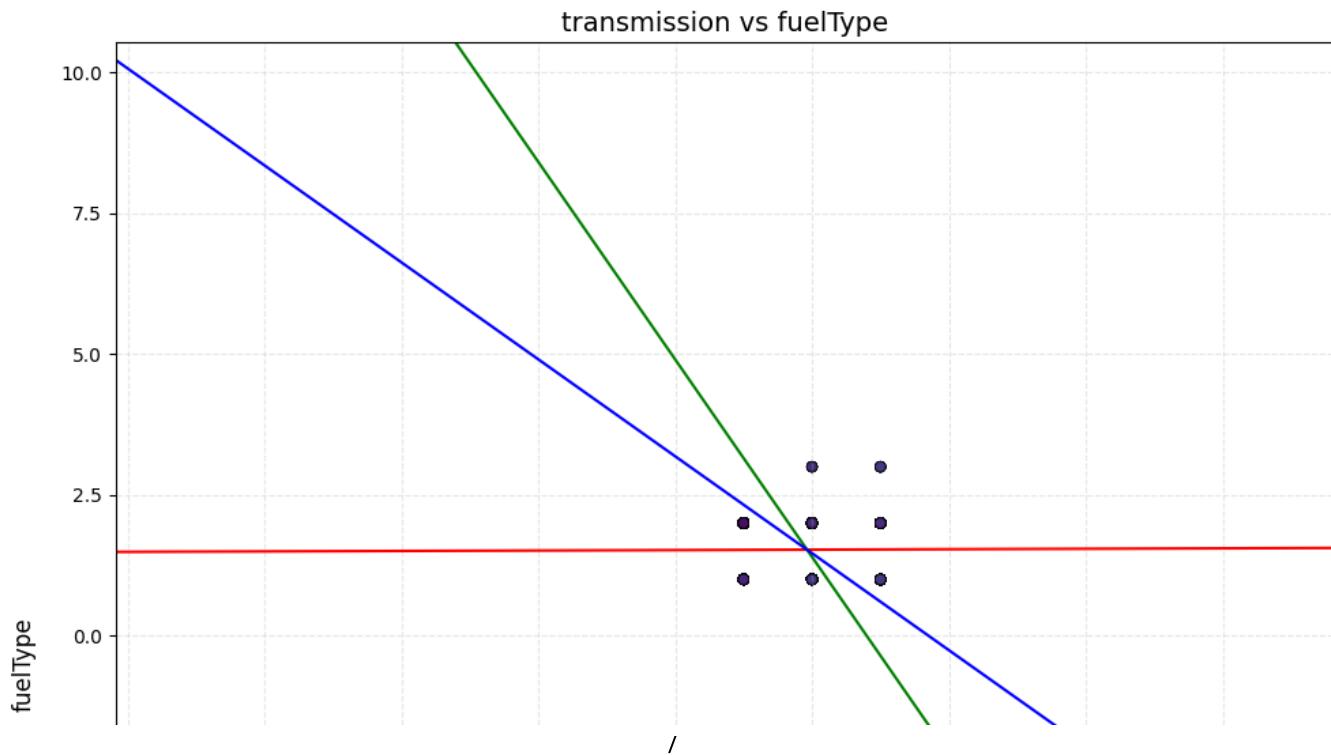
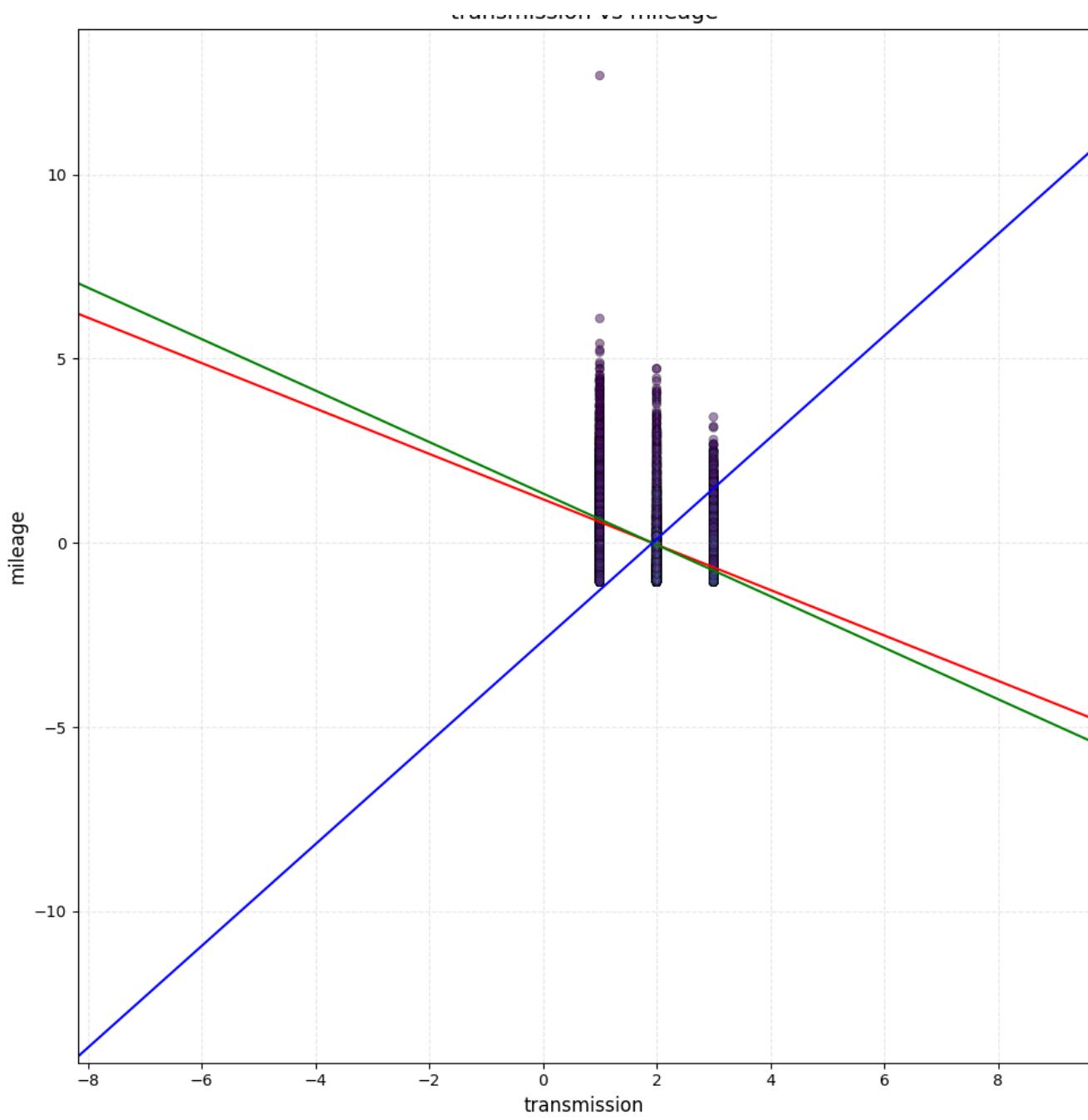
transmission vs model

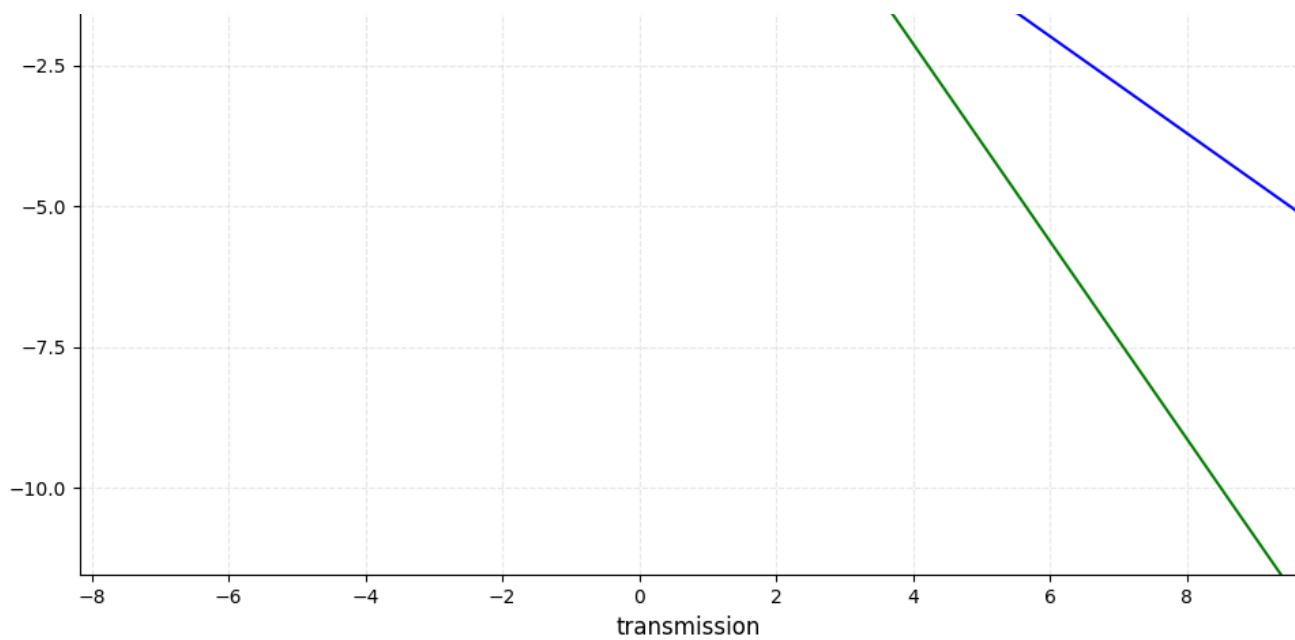




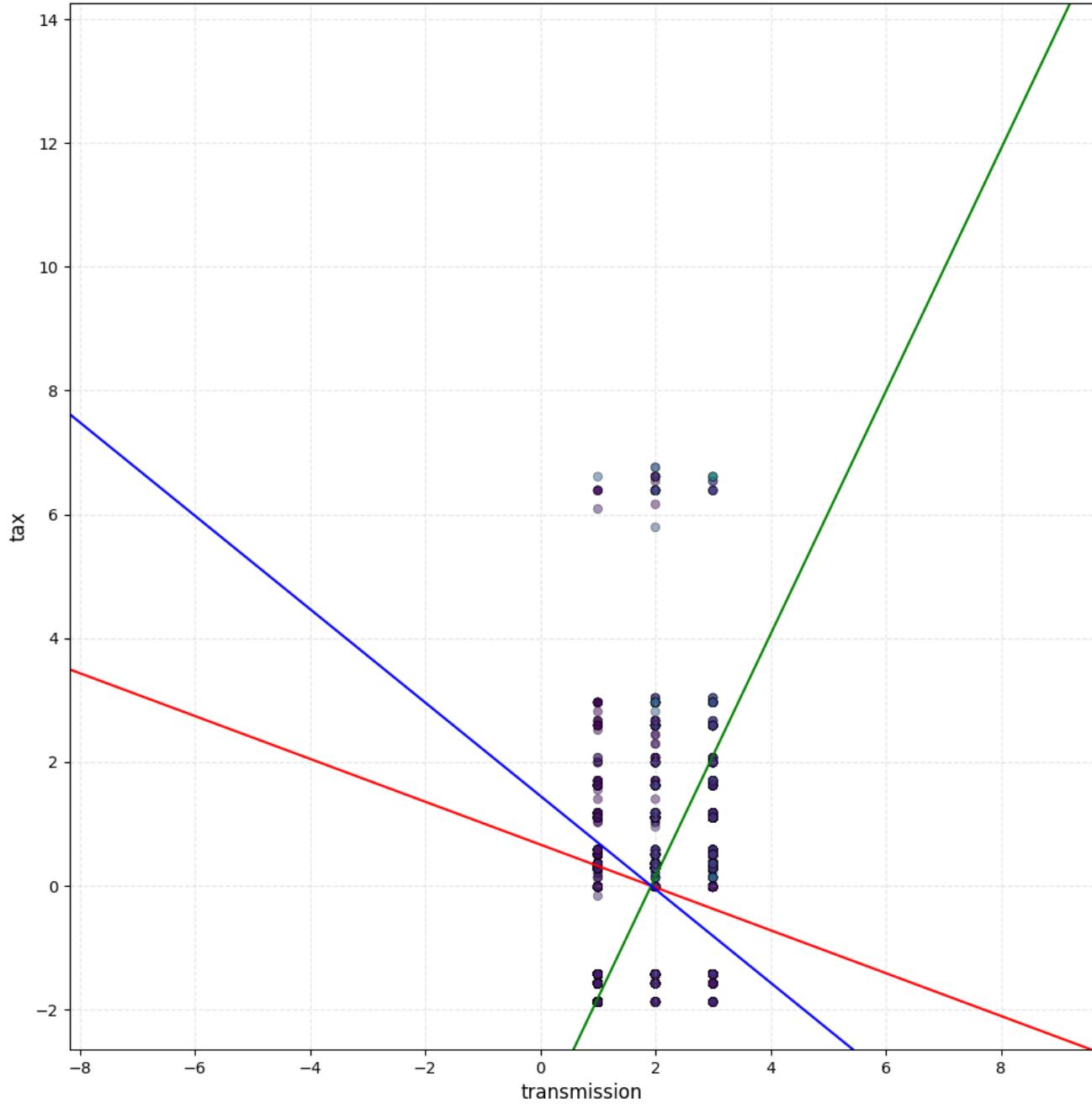
transmission vs transmission



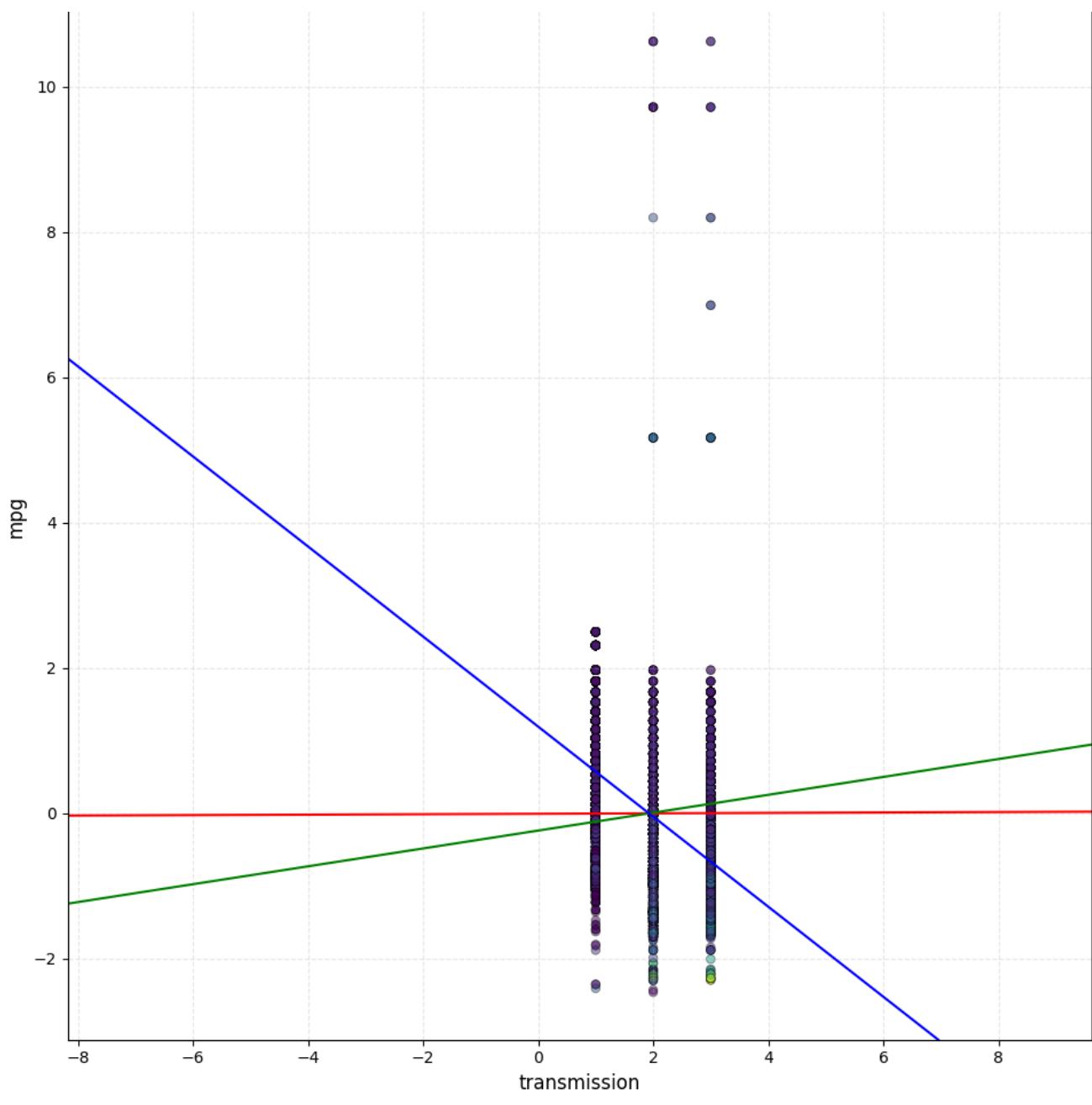




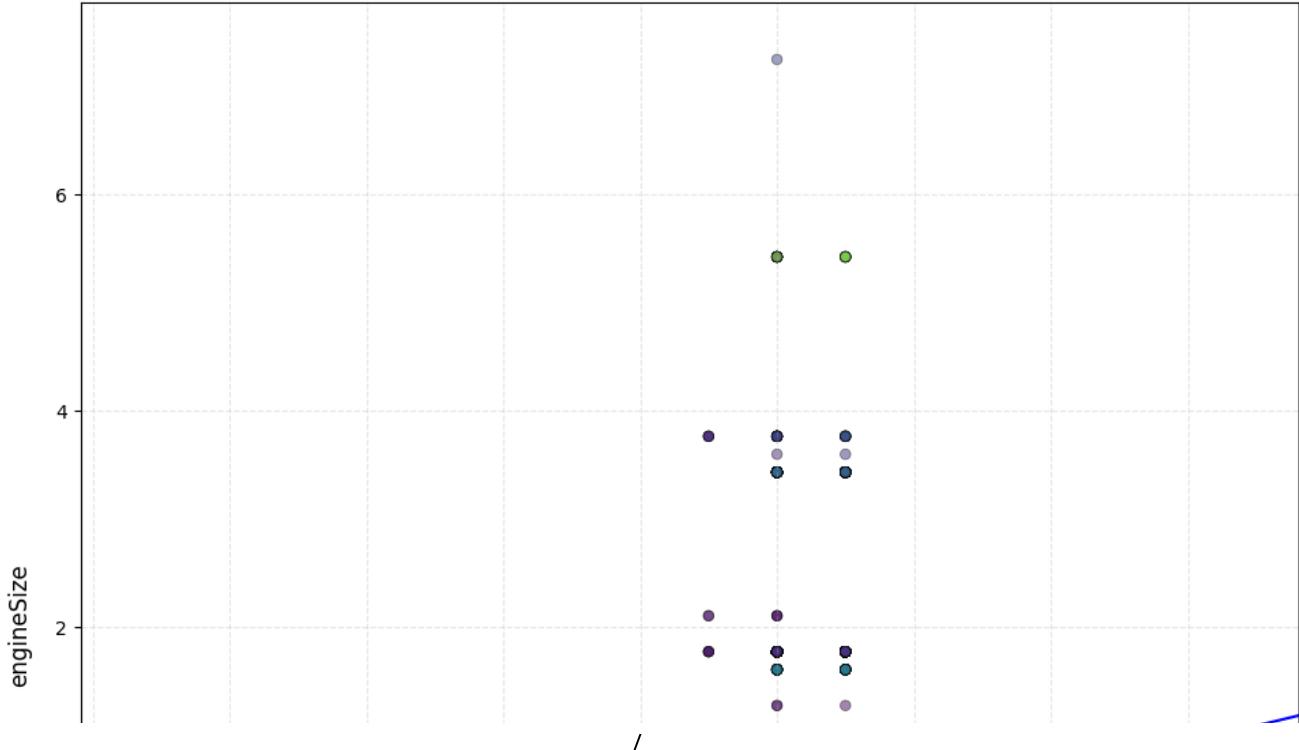
transmission vs tax

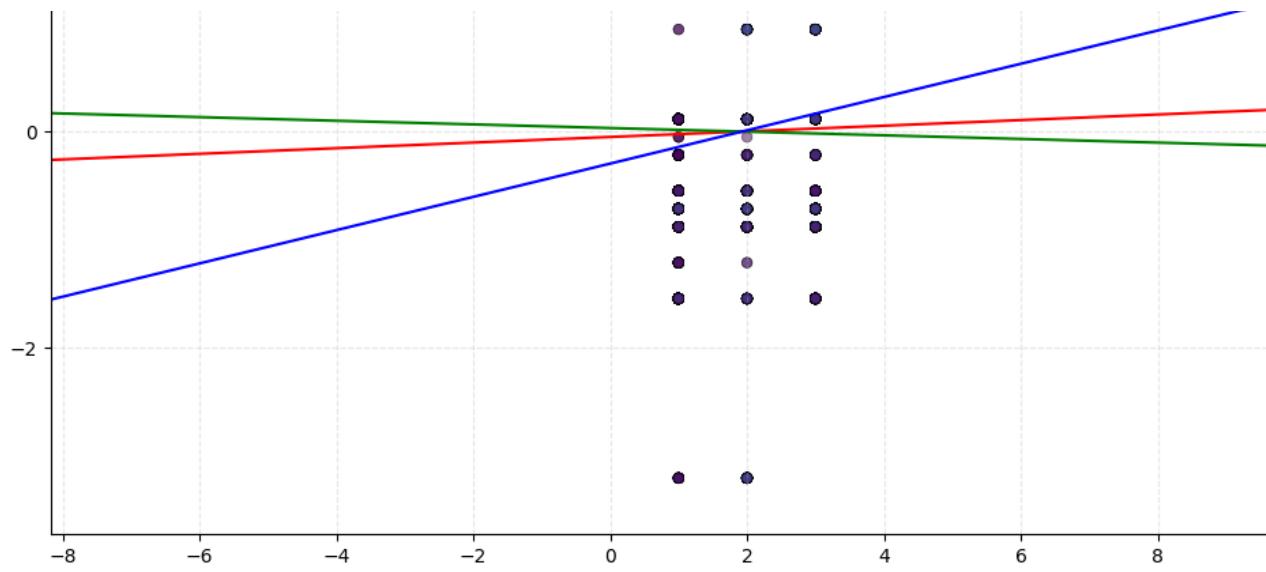


transmission vs mpg

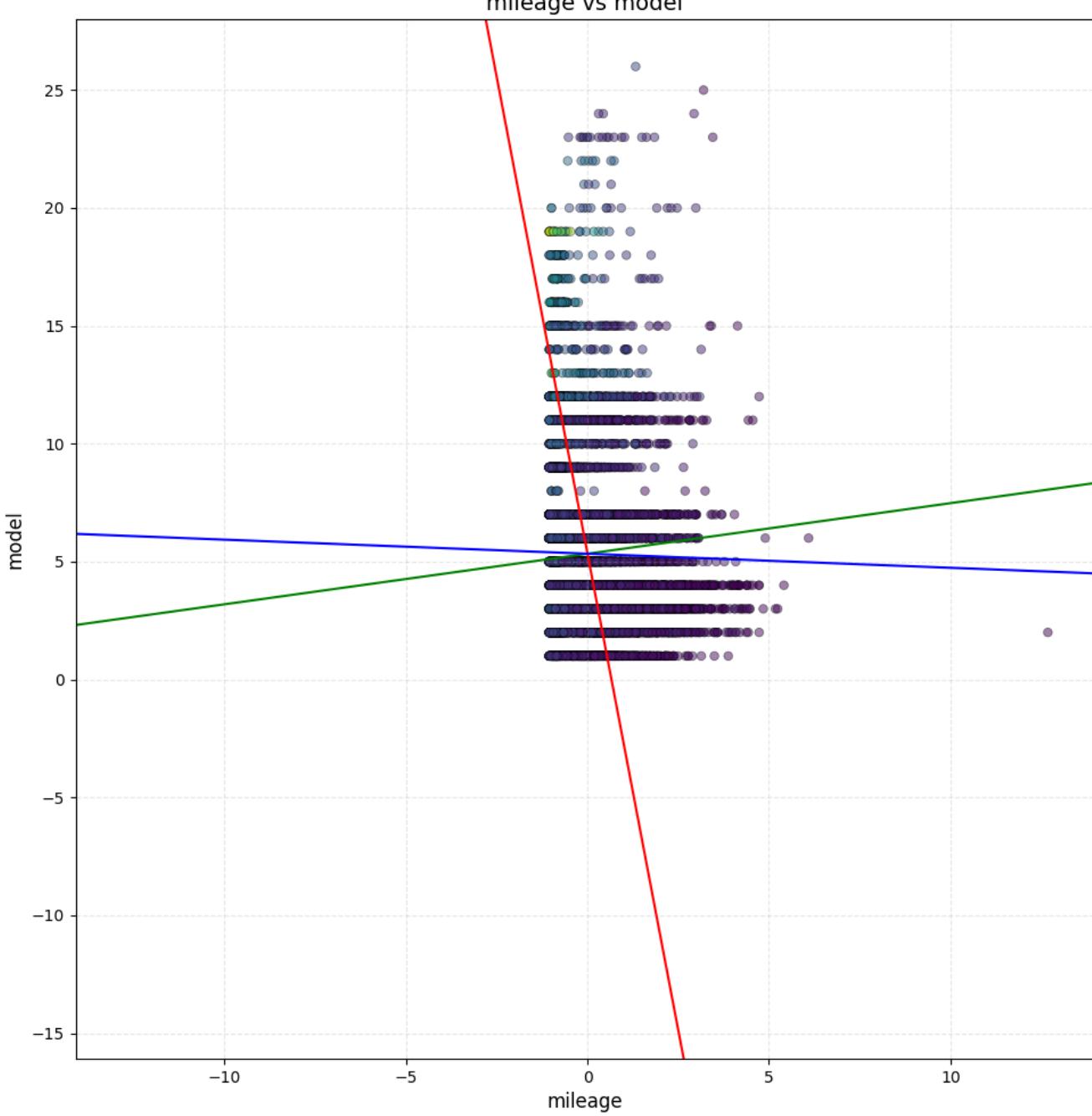


transmission vs engineSize



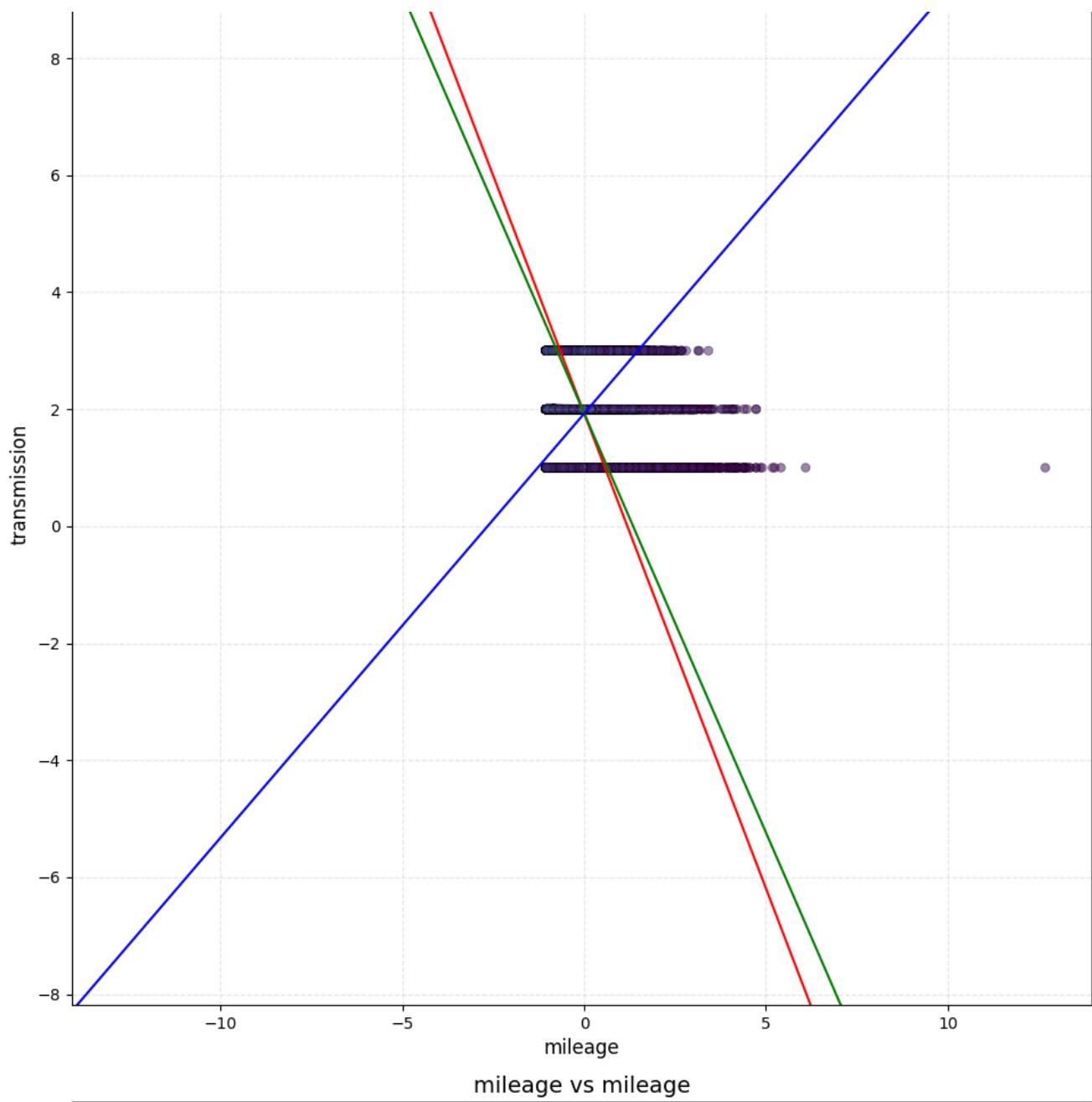


mileage vs model

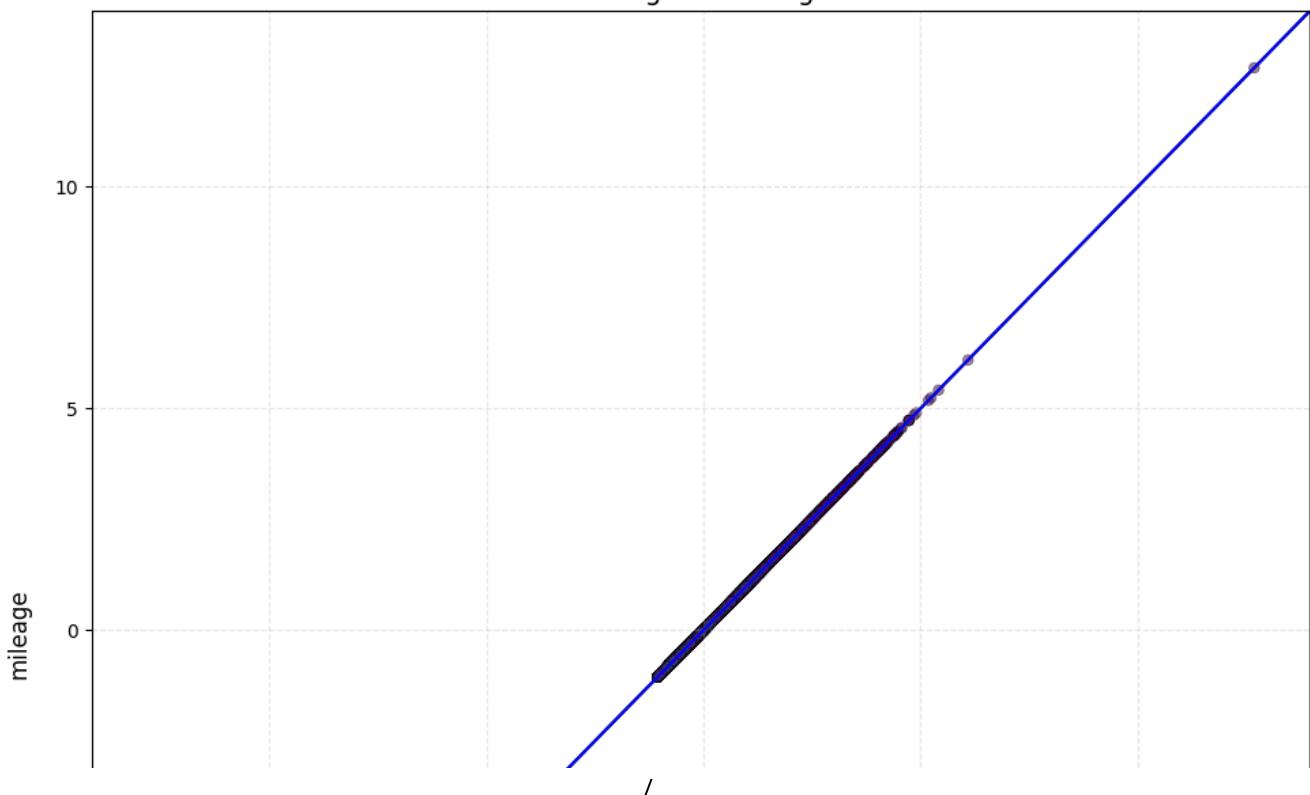


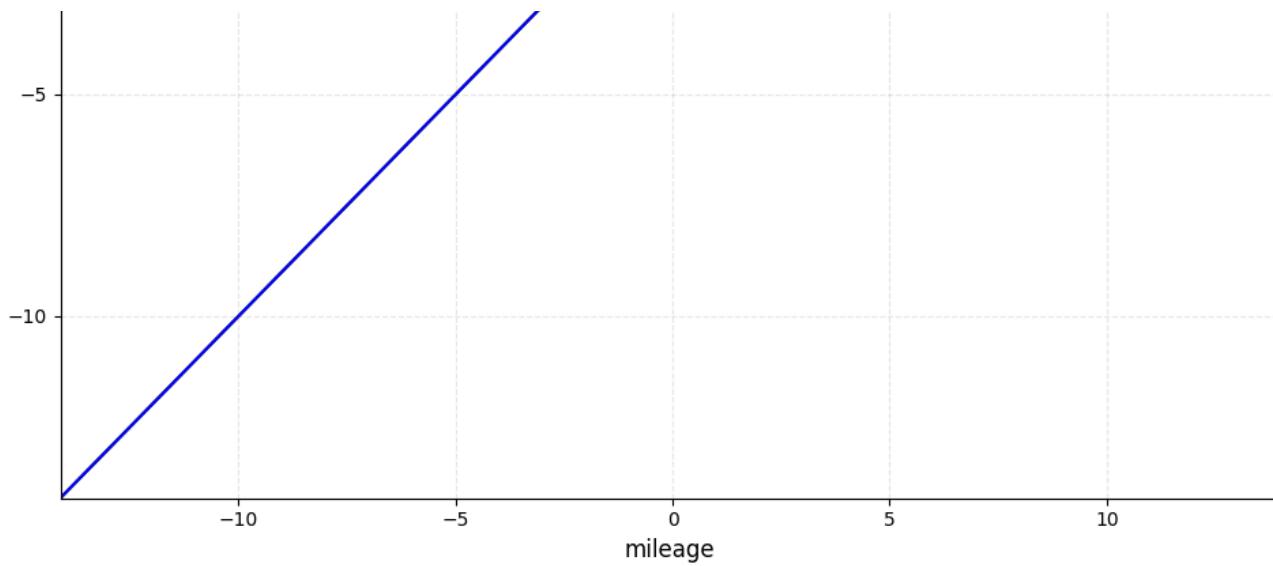
mileage vs transmission



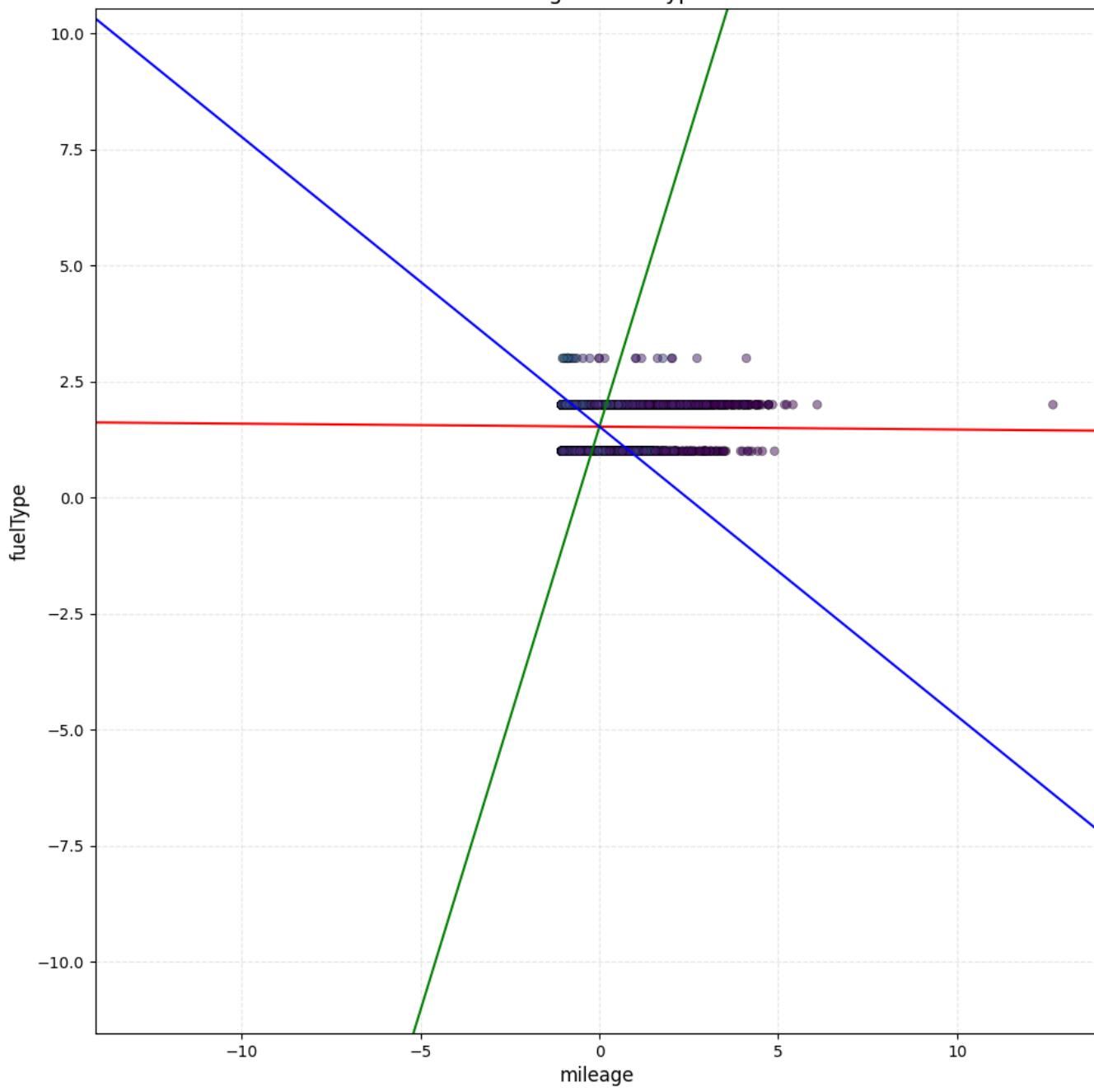


mileage vs mileage

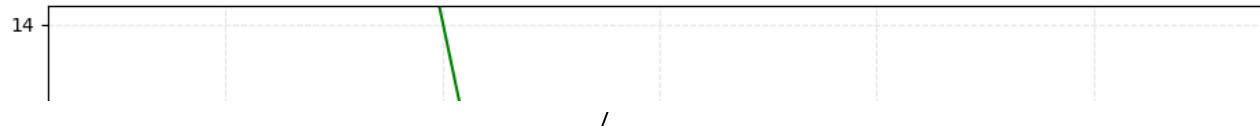


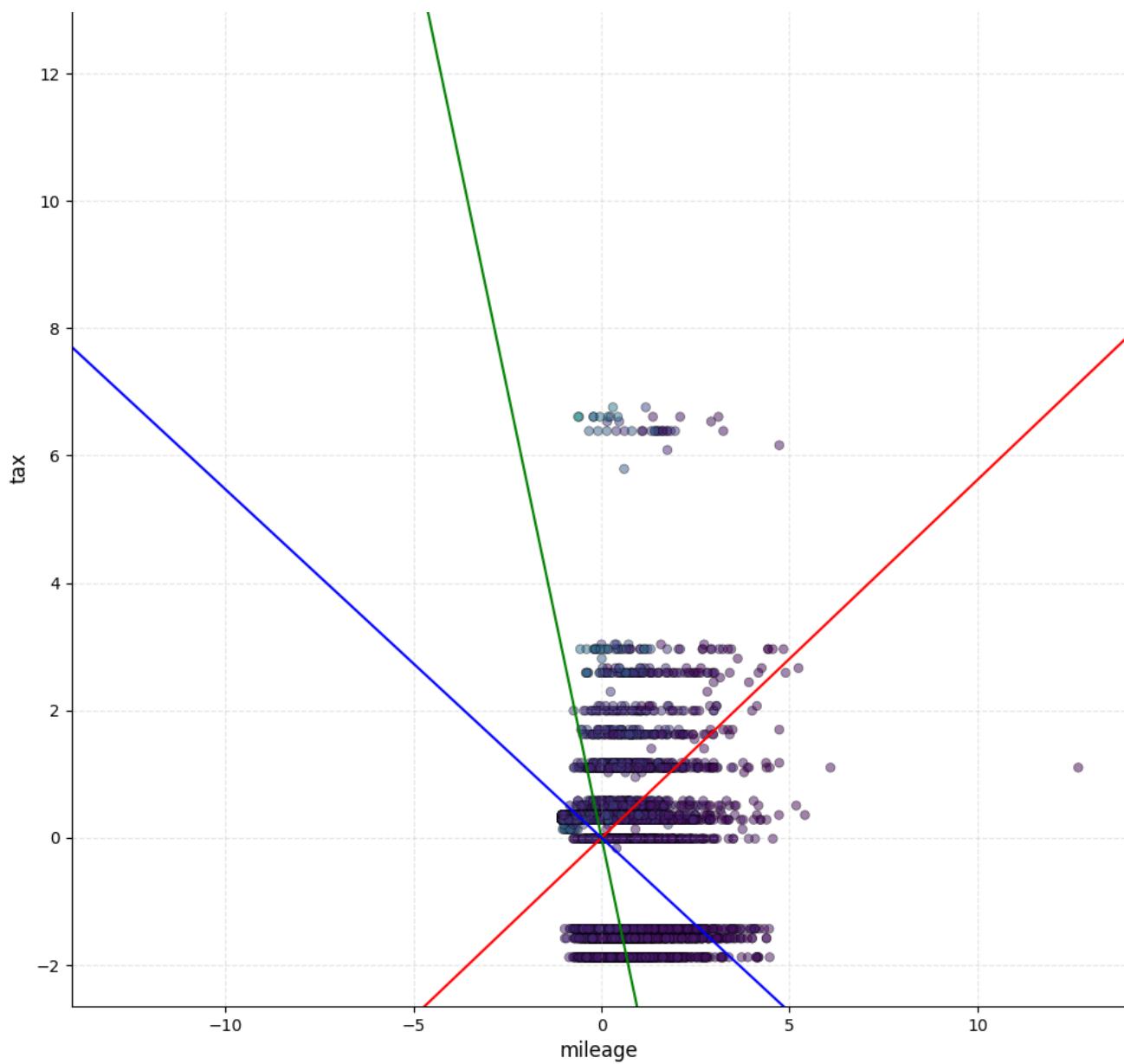


mileage vs fuelType

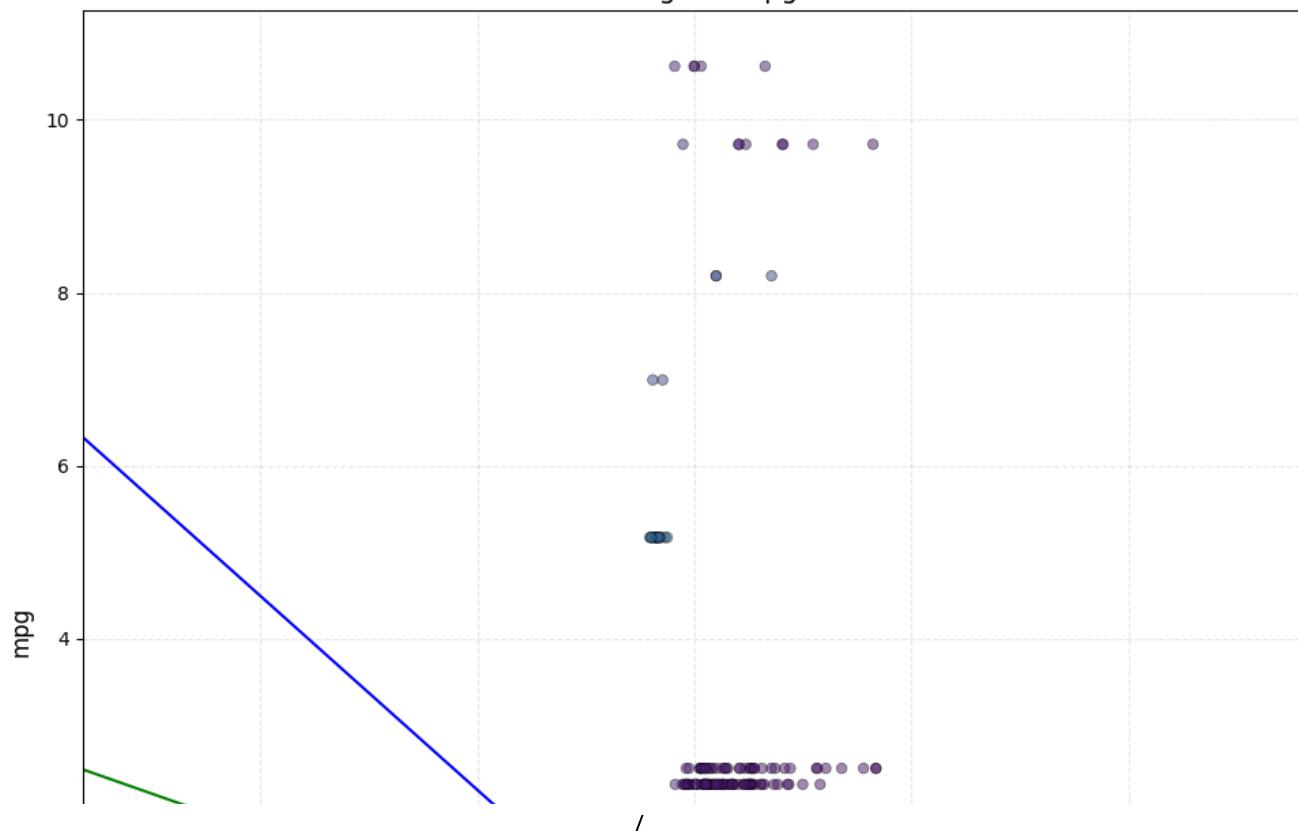


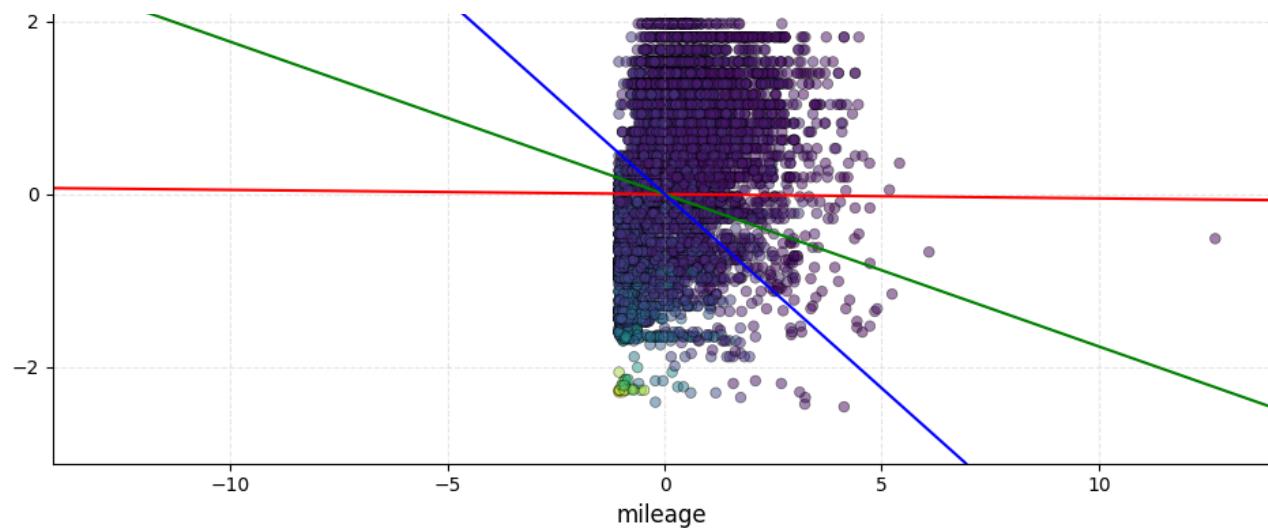
mileage vs tax



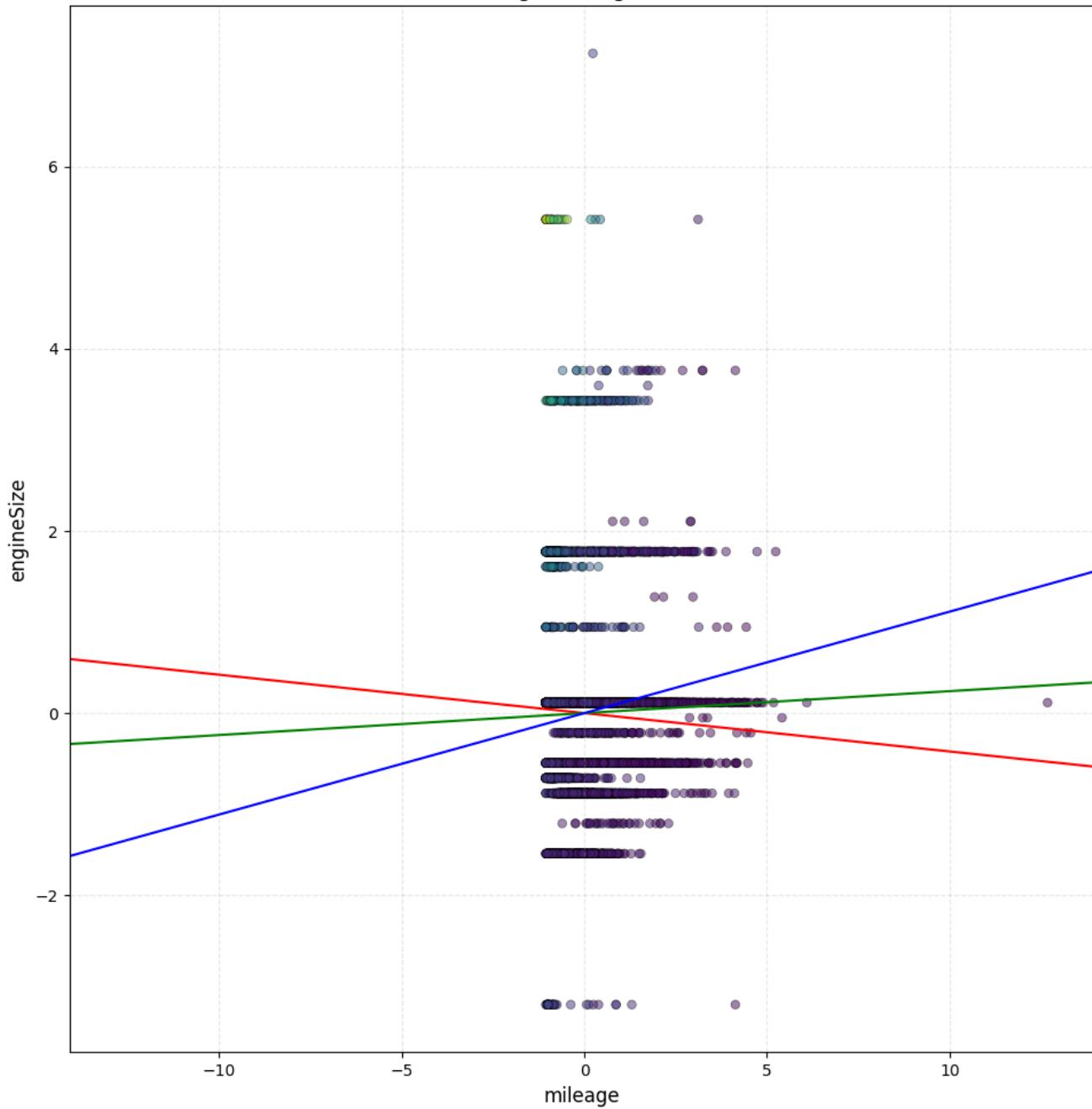


mileage vs mpg

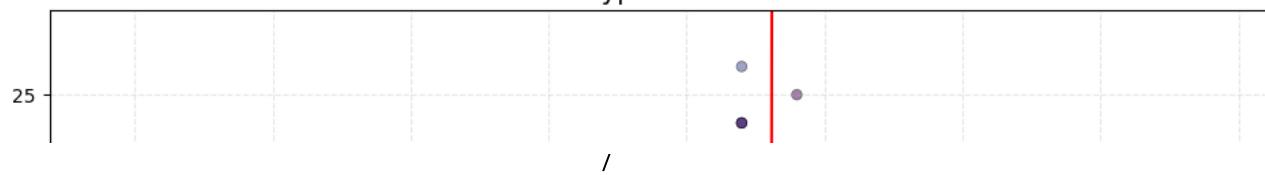


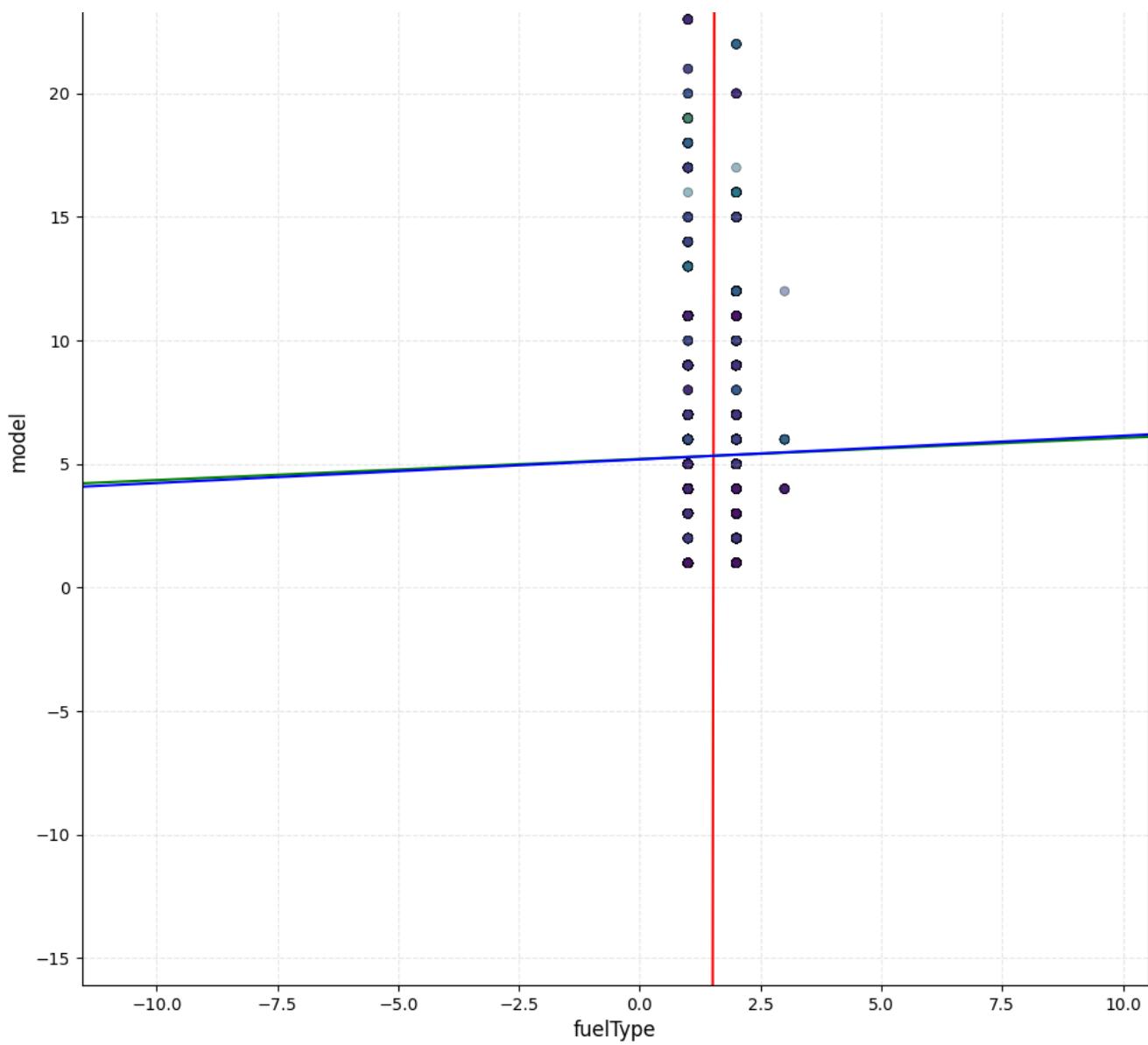


mileage vs engineSize

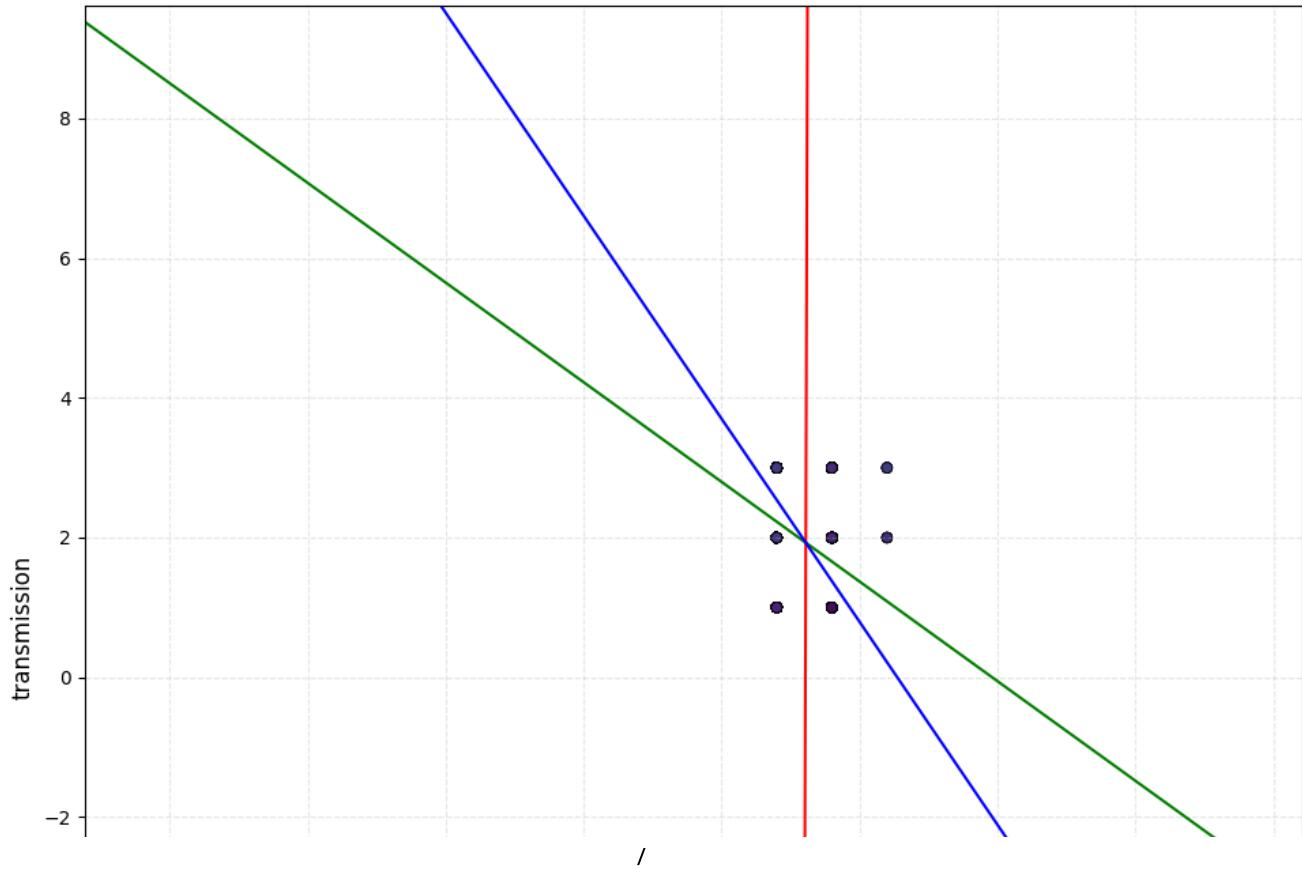


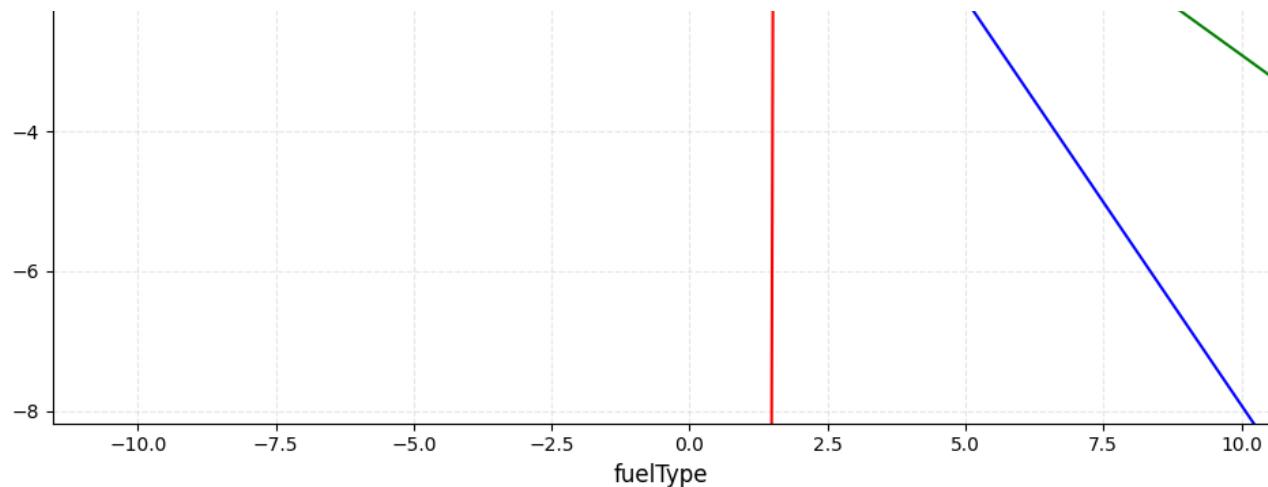
fuelType vs model



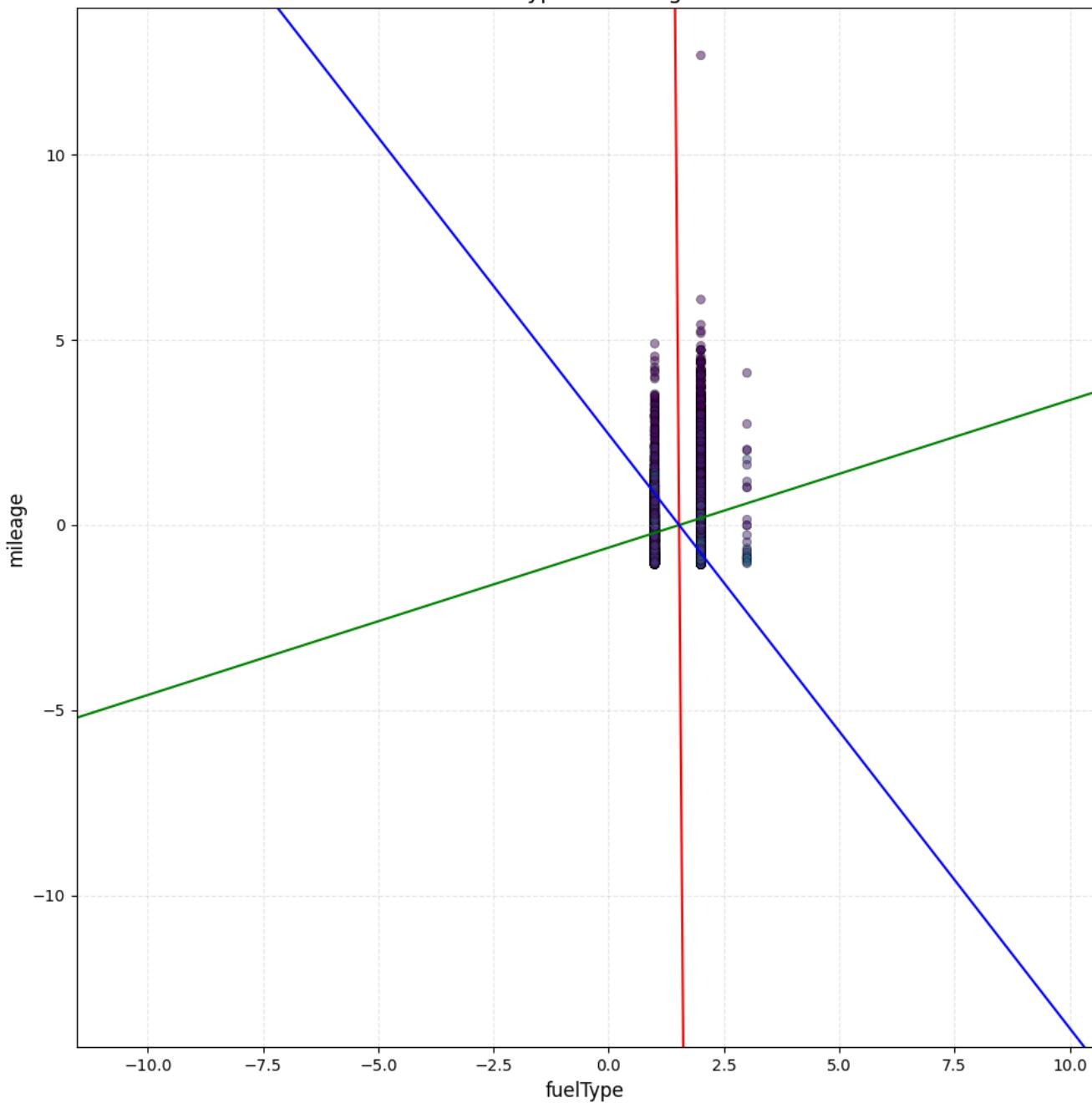


fuelType vs transmission

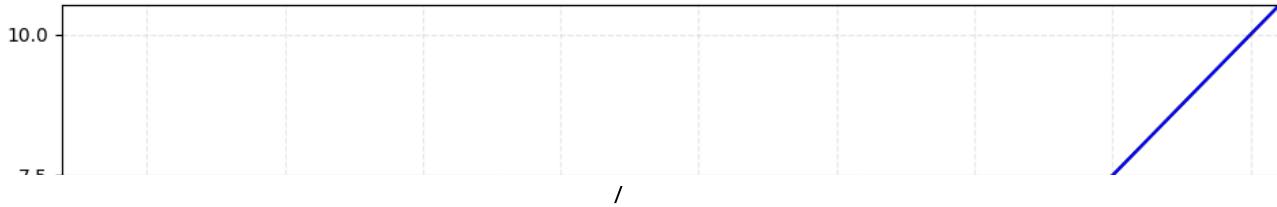


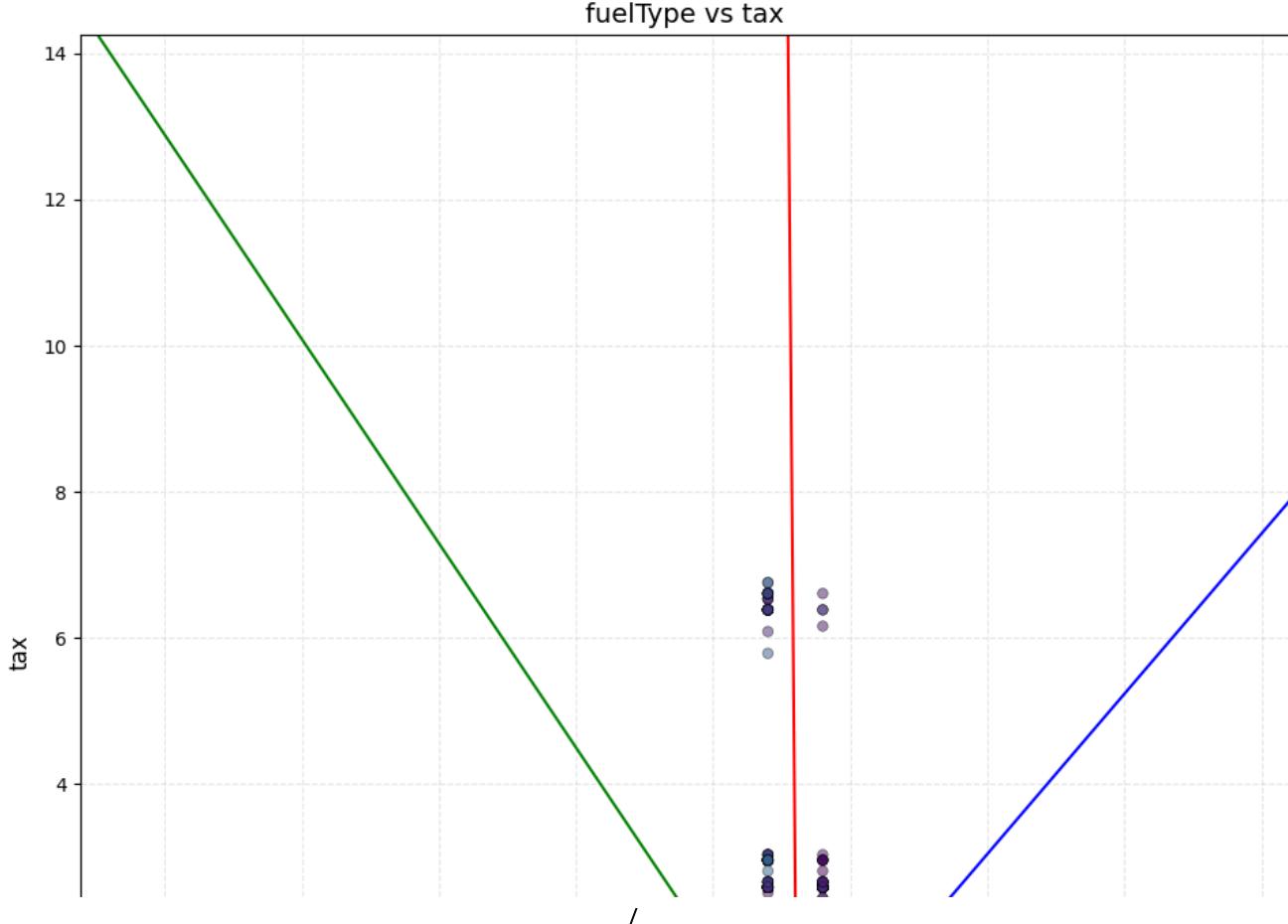
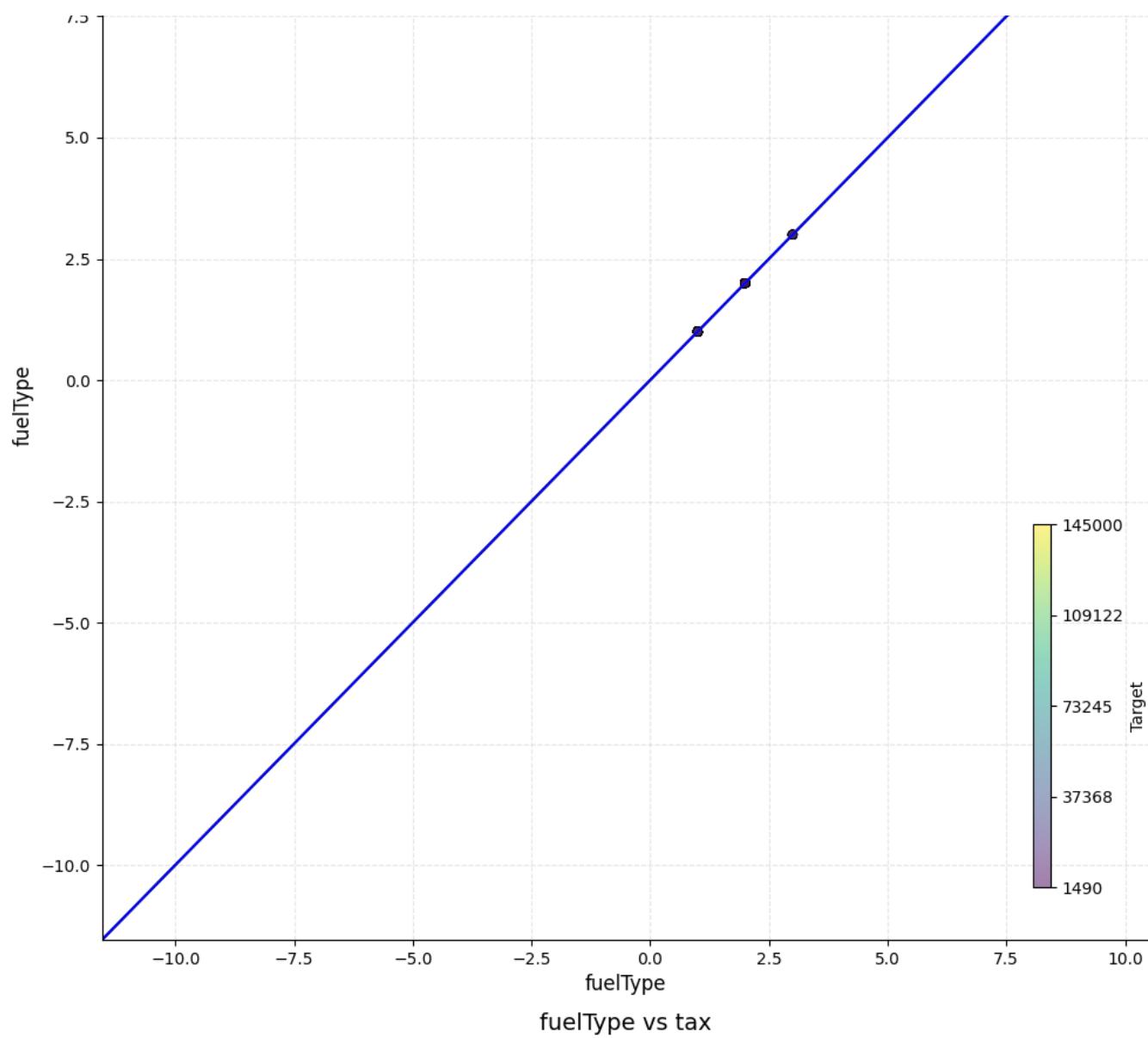


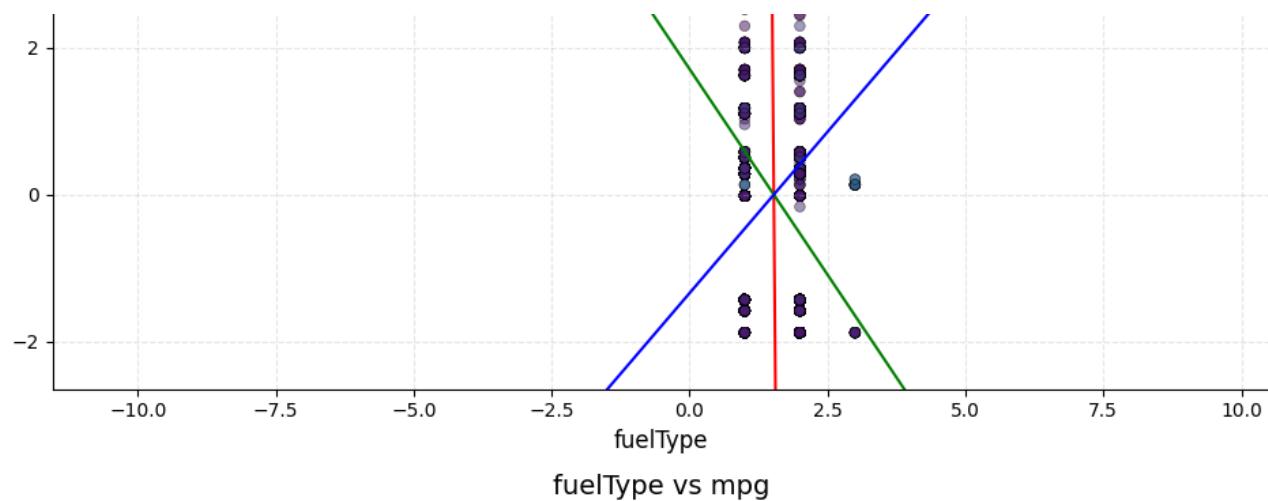
fuelType vs mileage



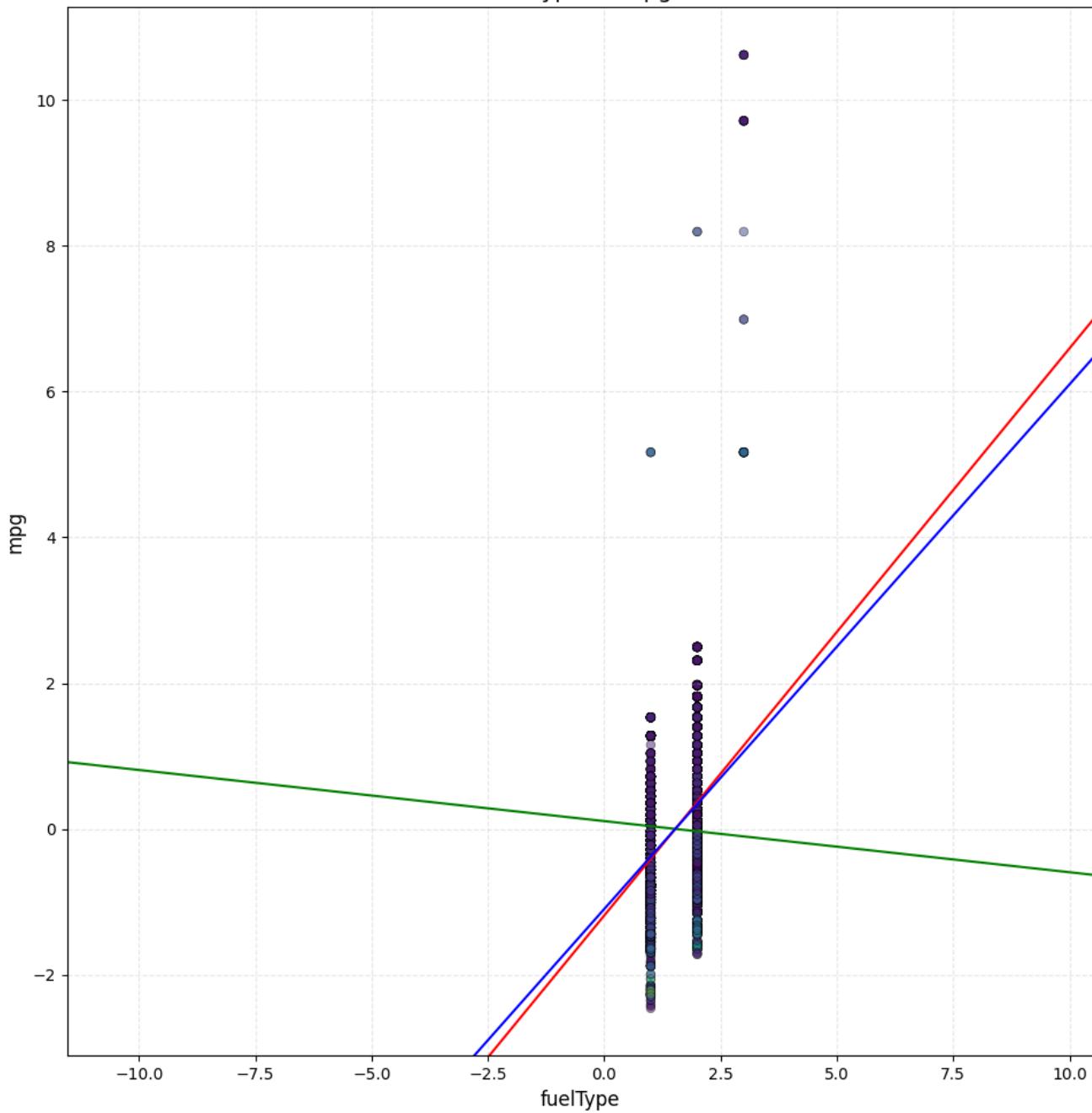
fuelType vs fuelType



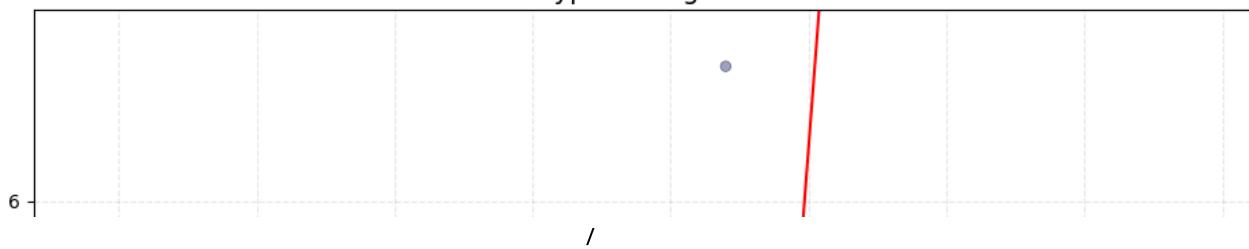




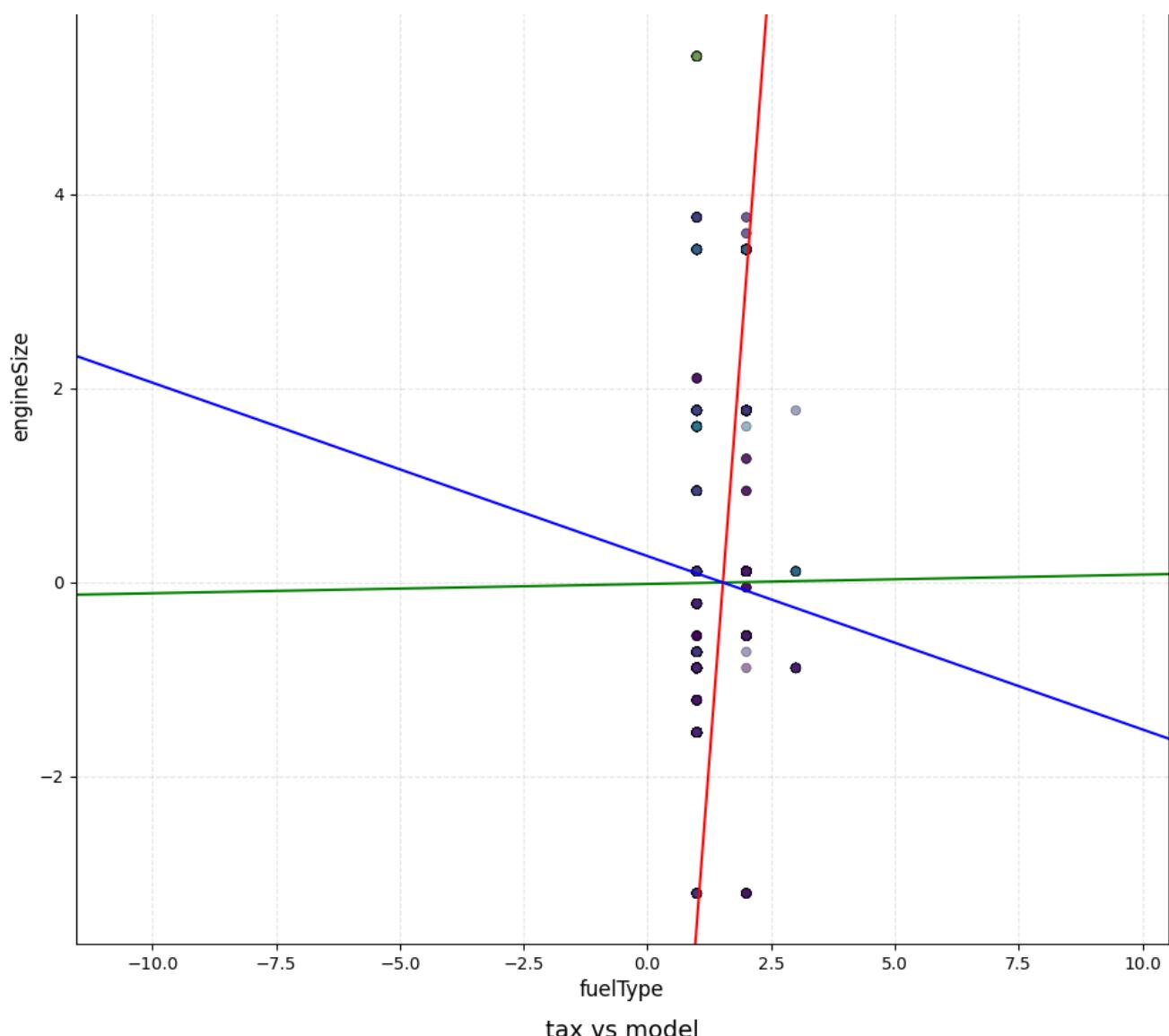
fuelType vs mpg



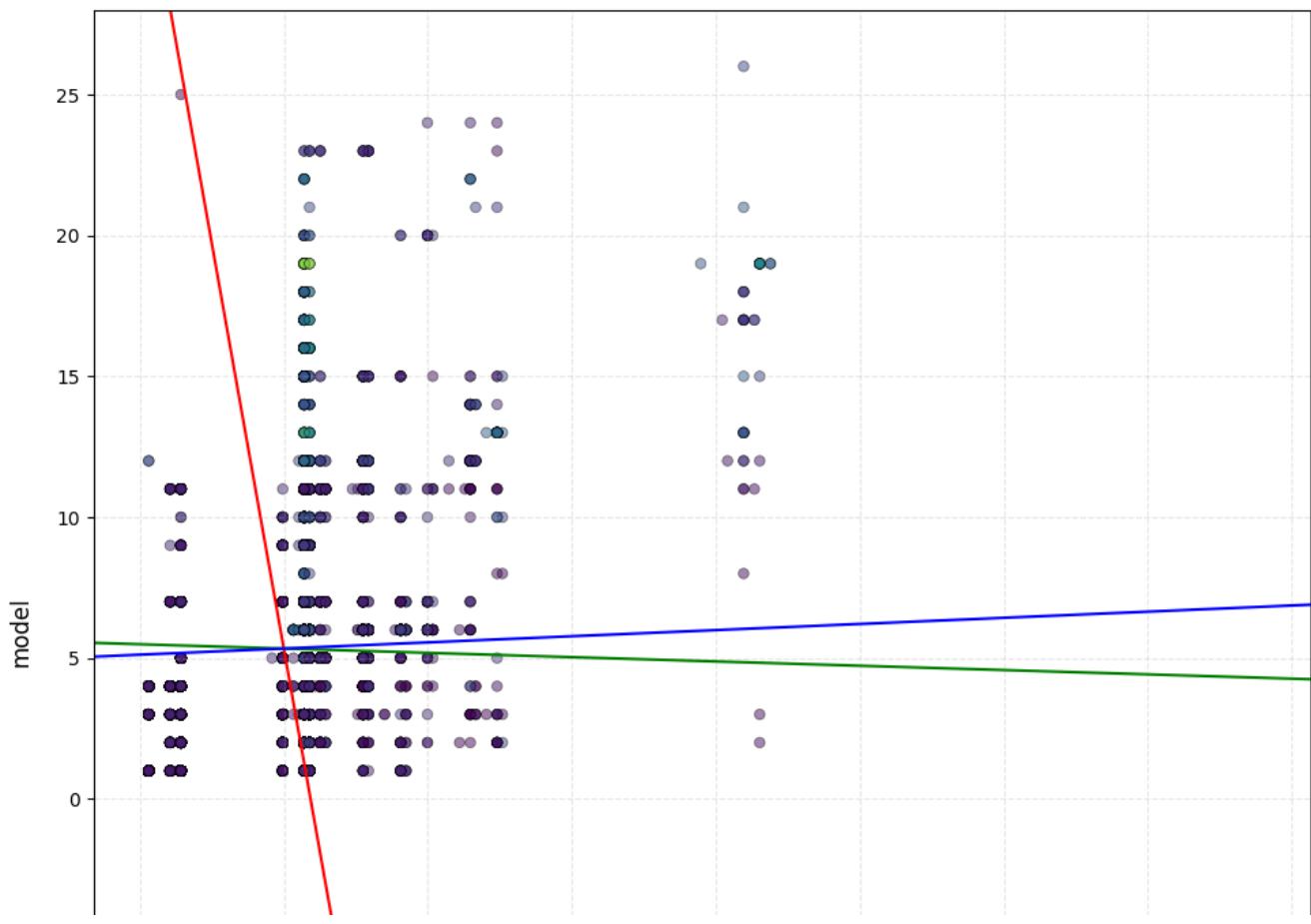
fuelType vs engineSize

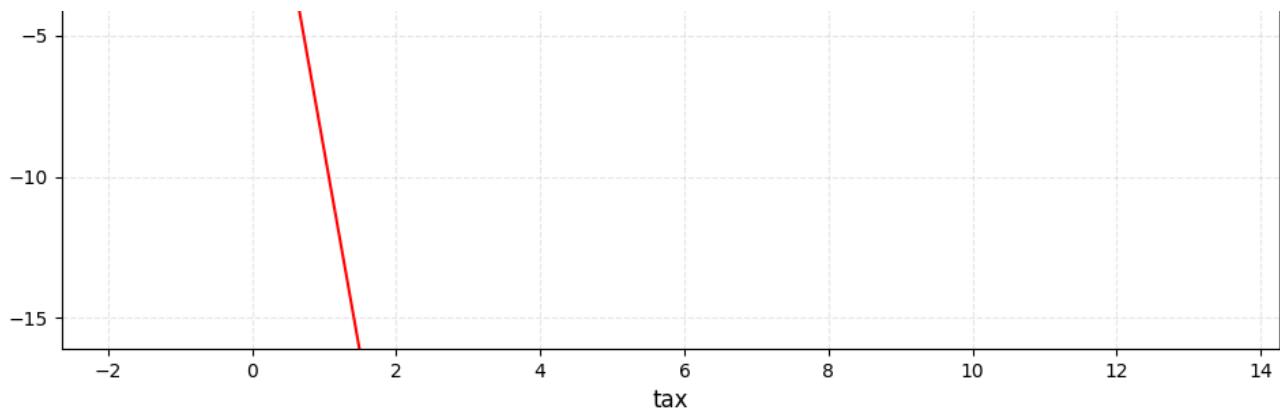


/

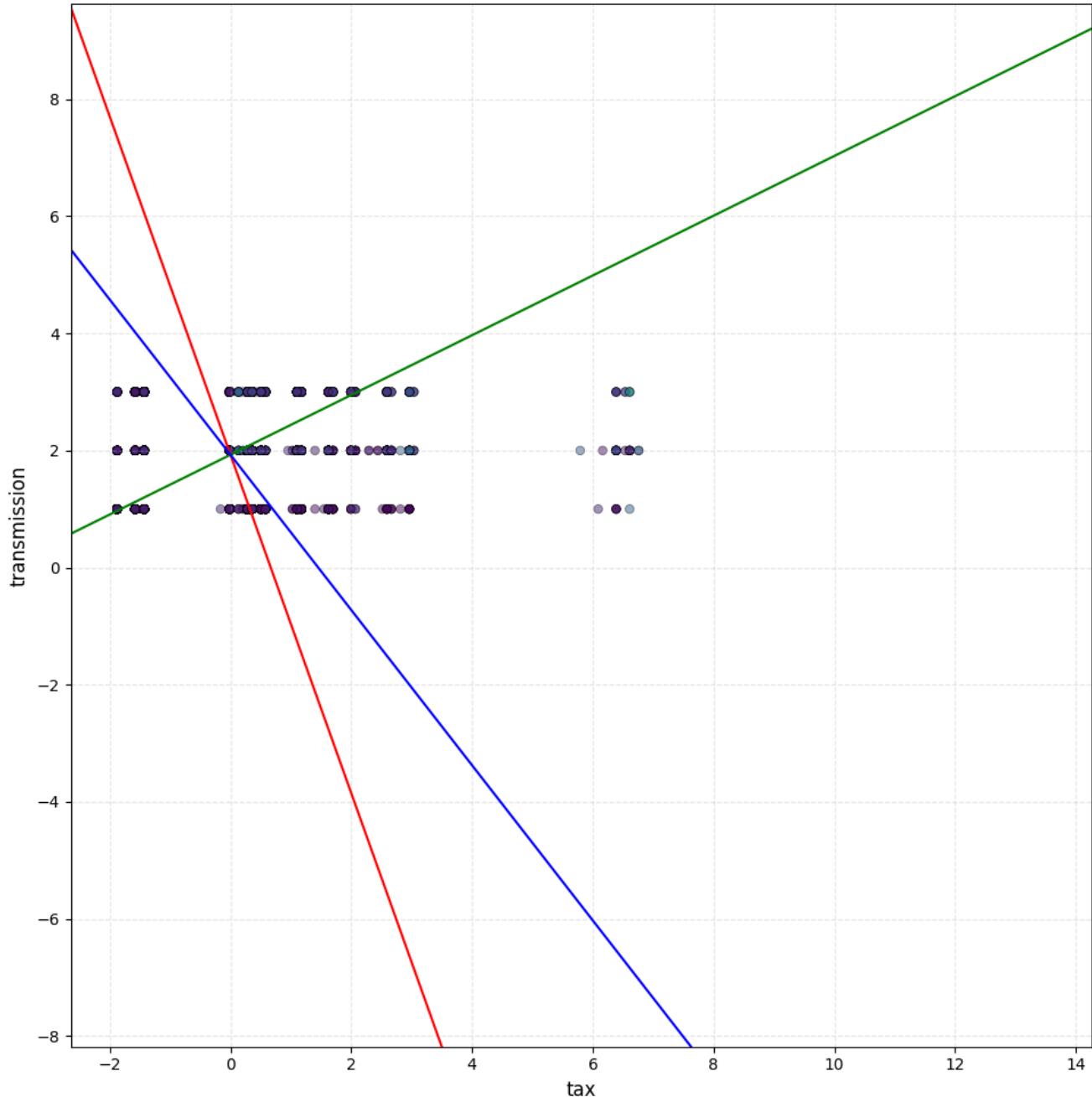


tax vs model

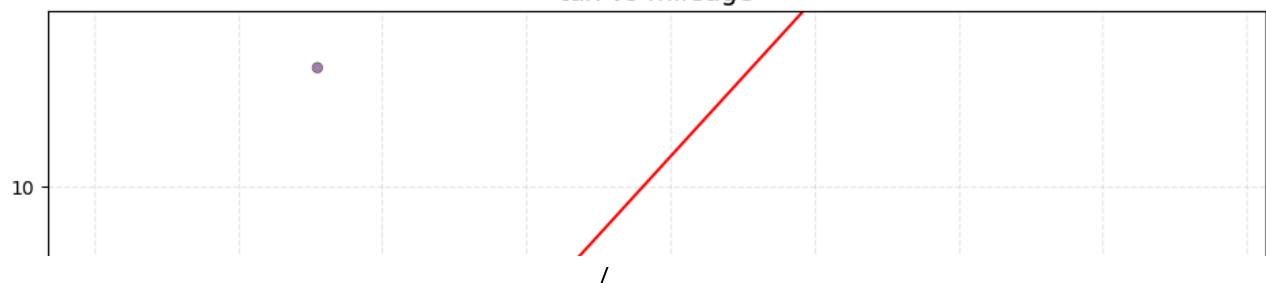


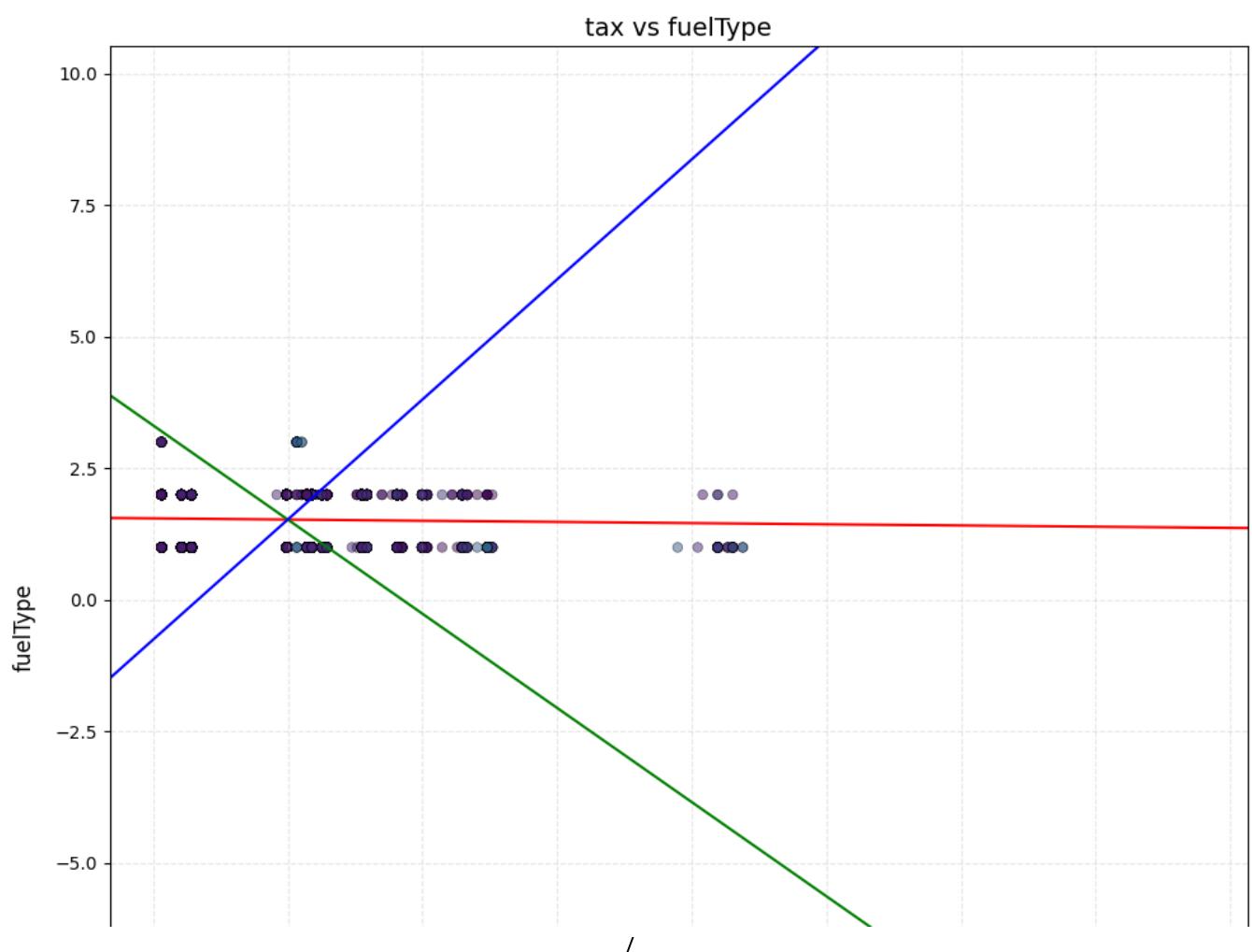
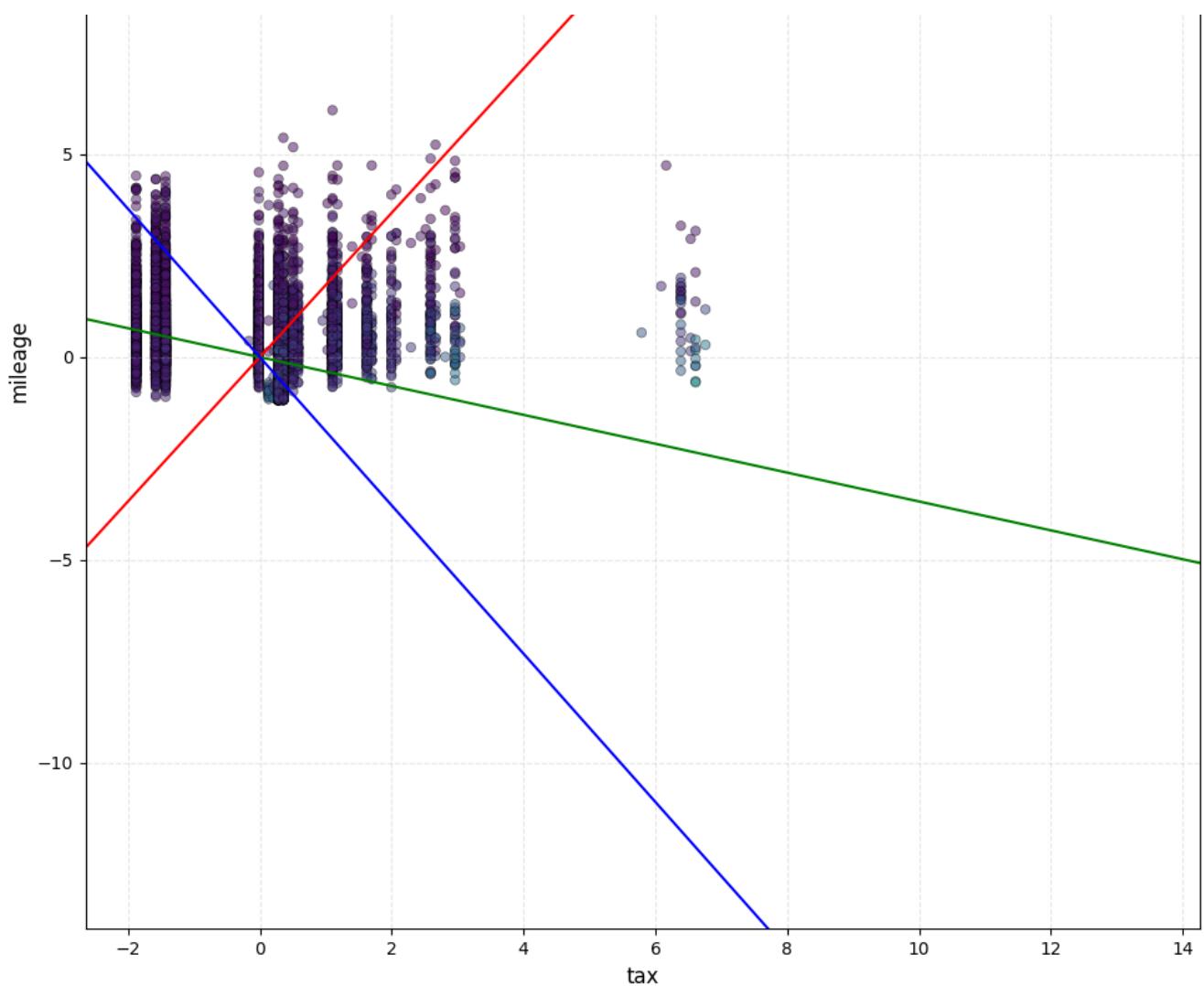


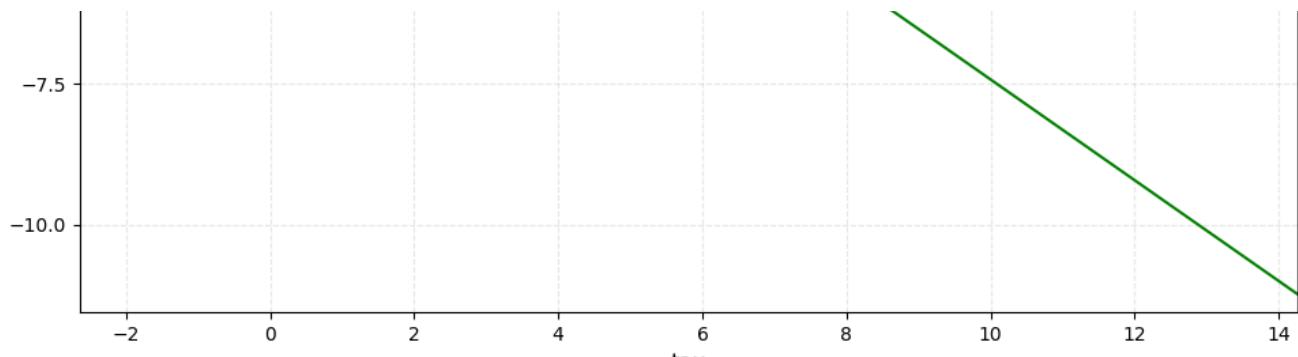
tax vs transmission



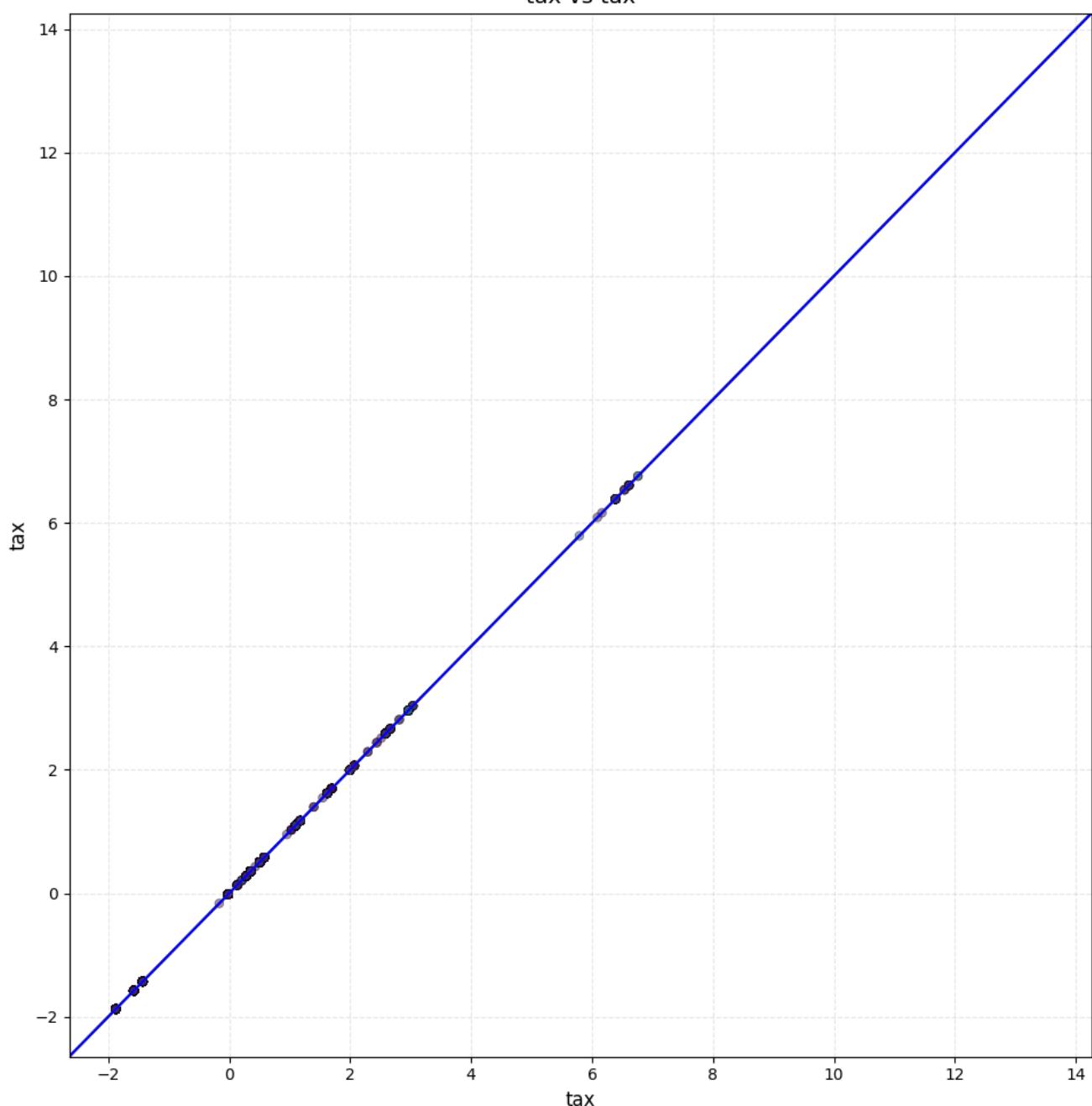
tax vs mileage







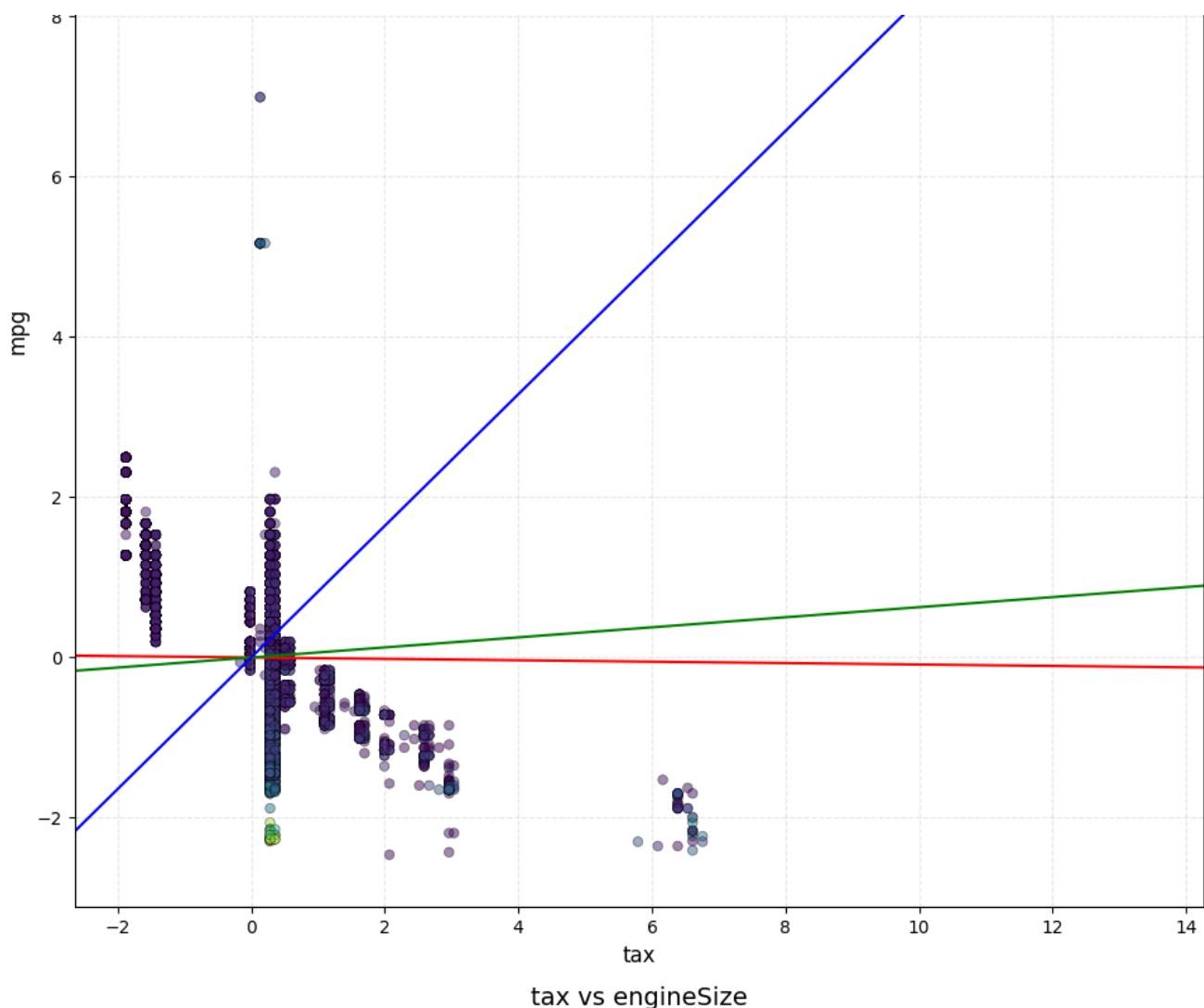
tax vs tax



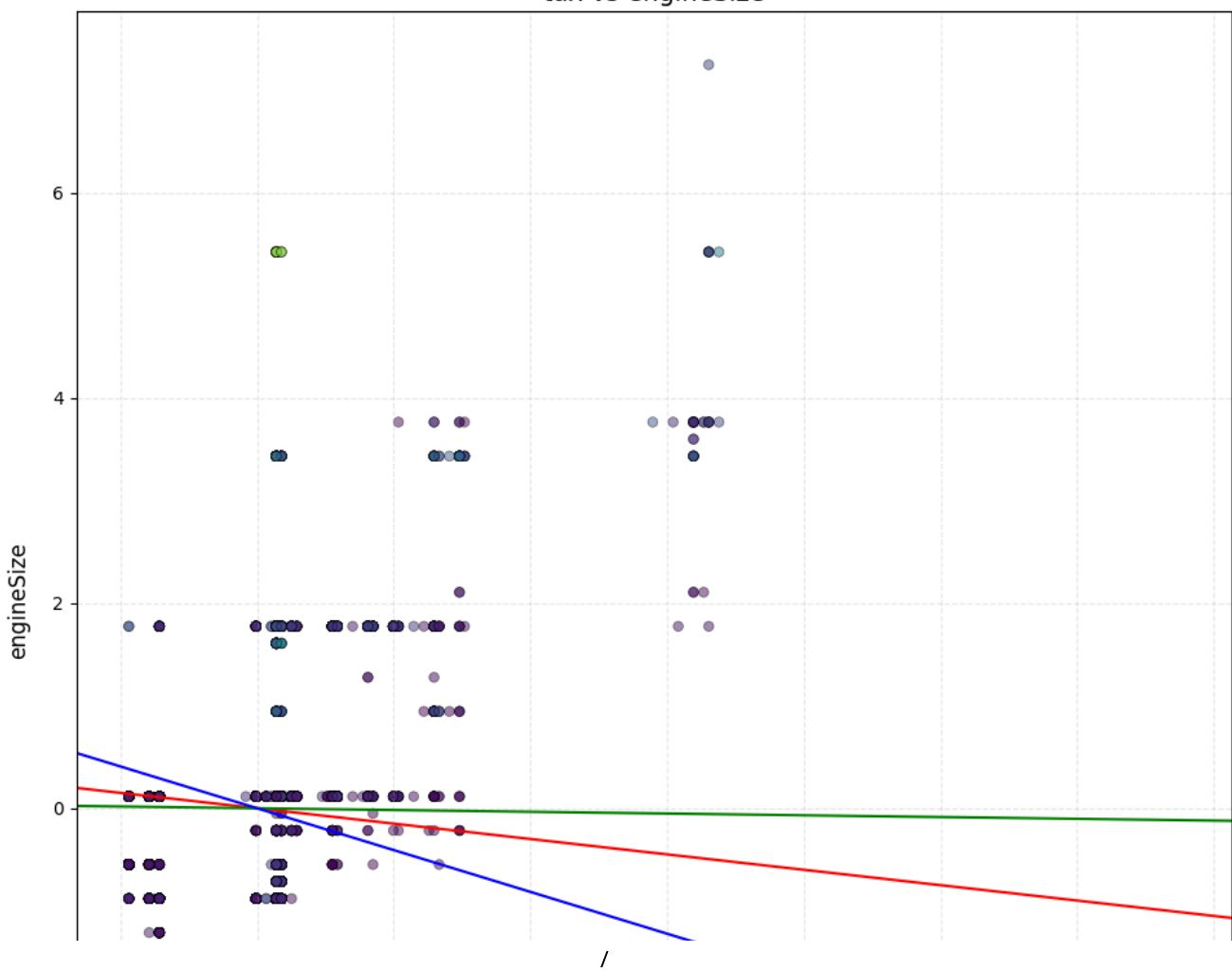
tax vs tax

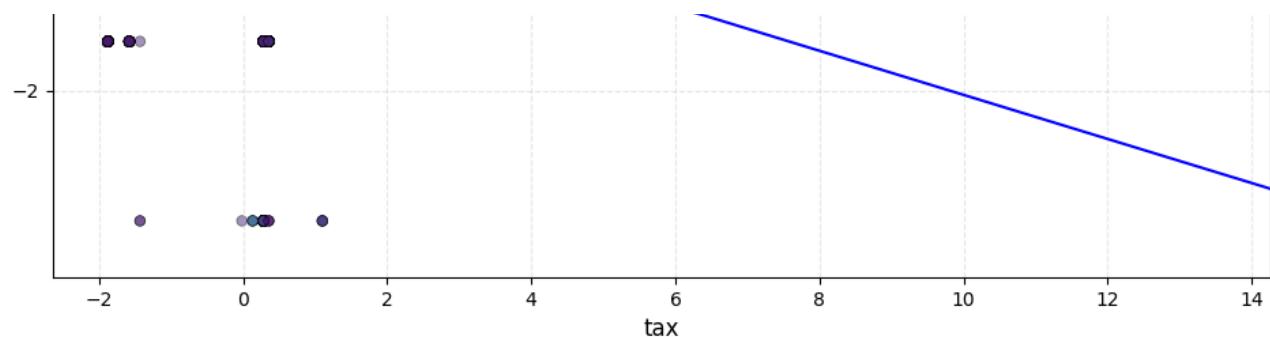
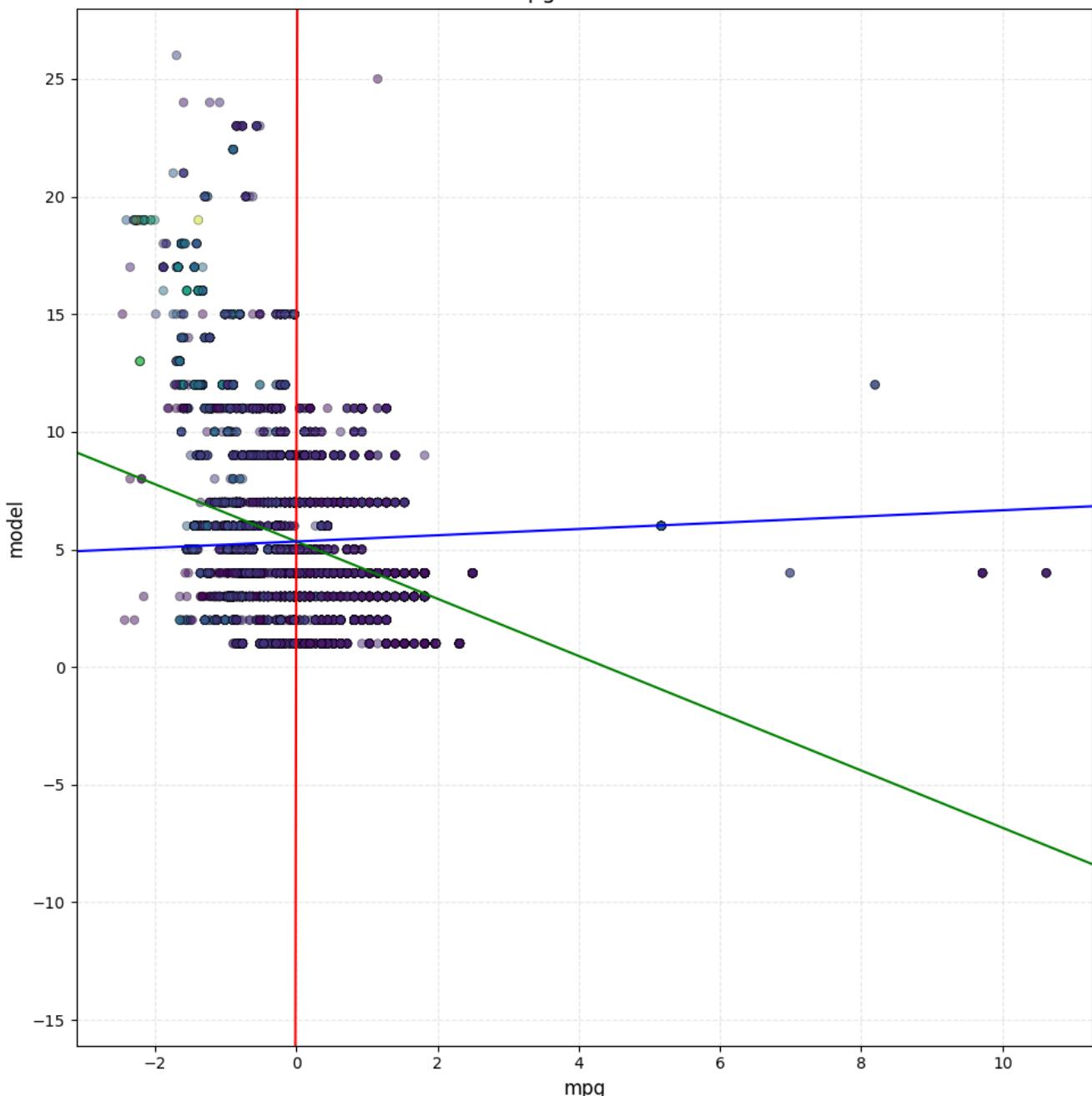


/

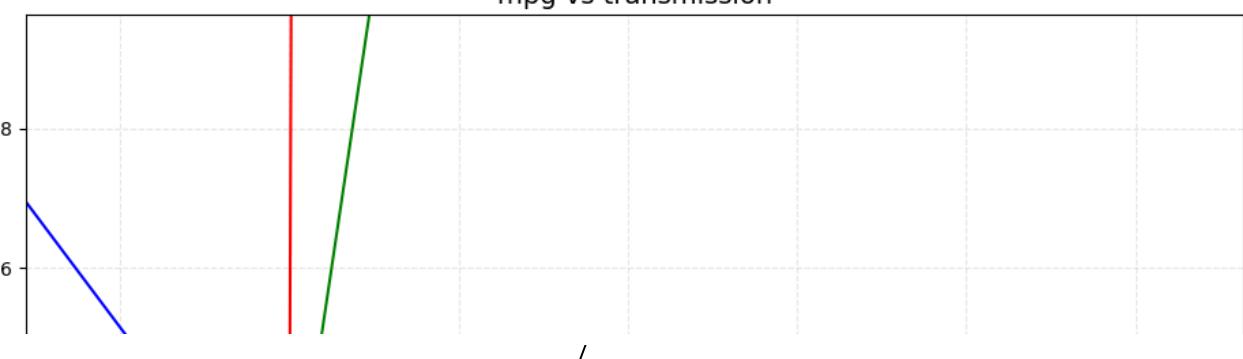


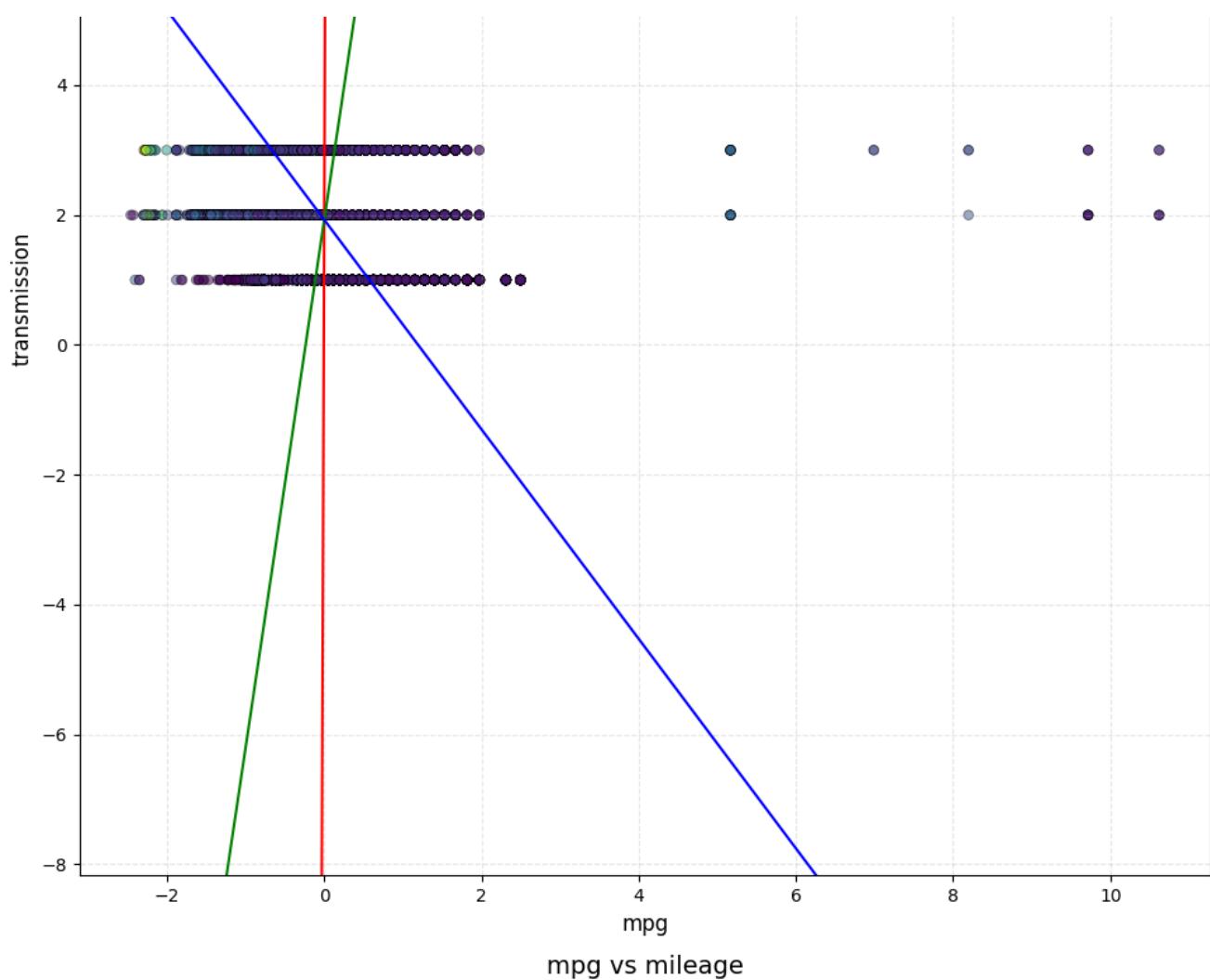
tax vs engineSize



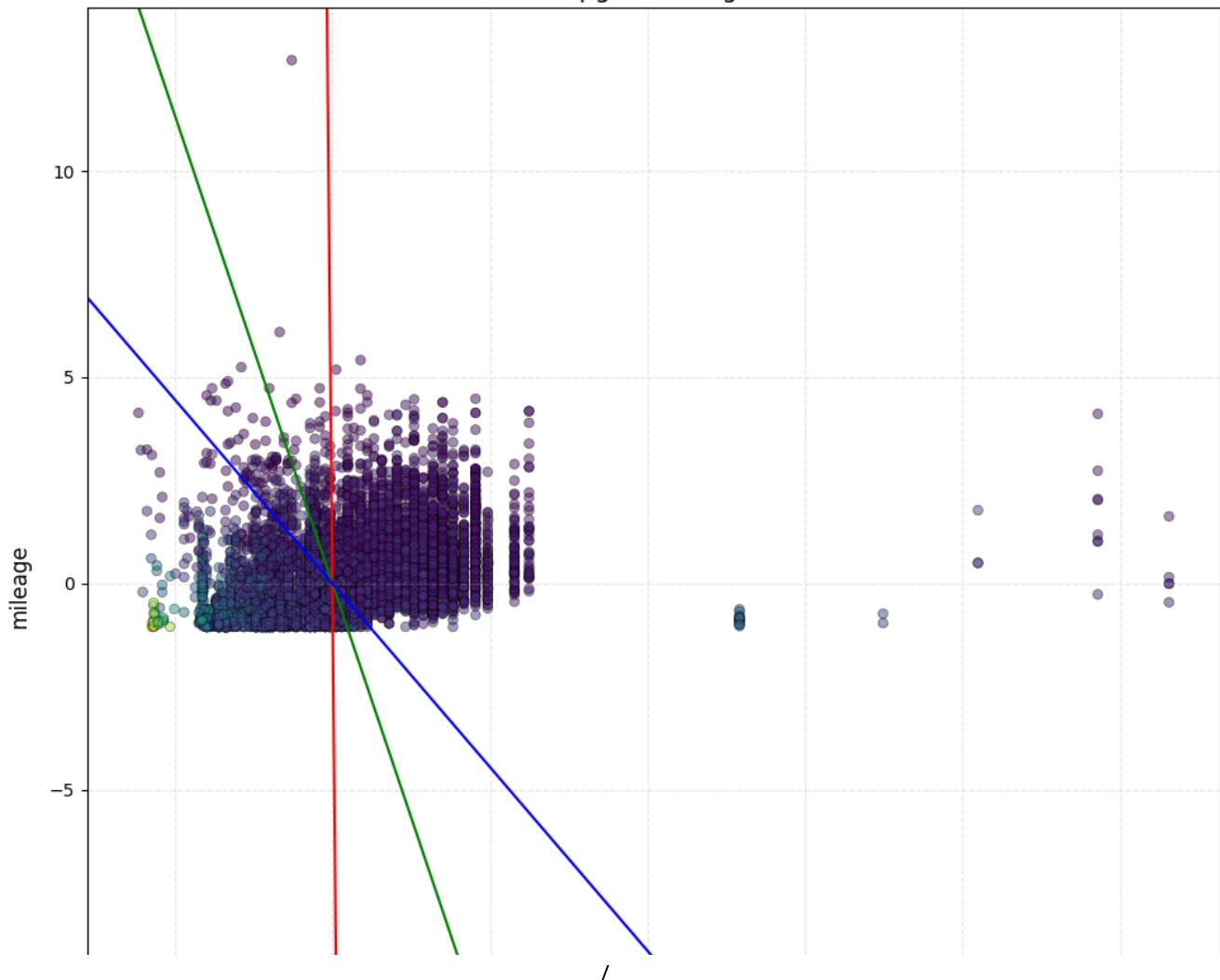
tax  
mpg vs model

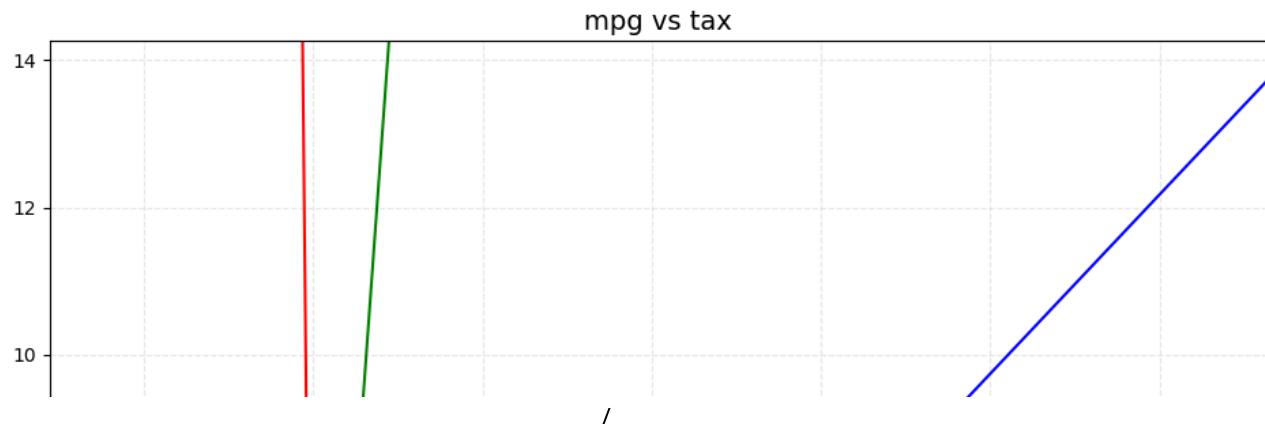
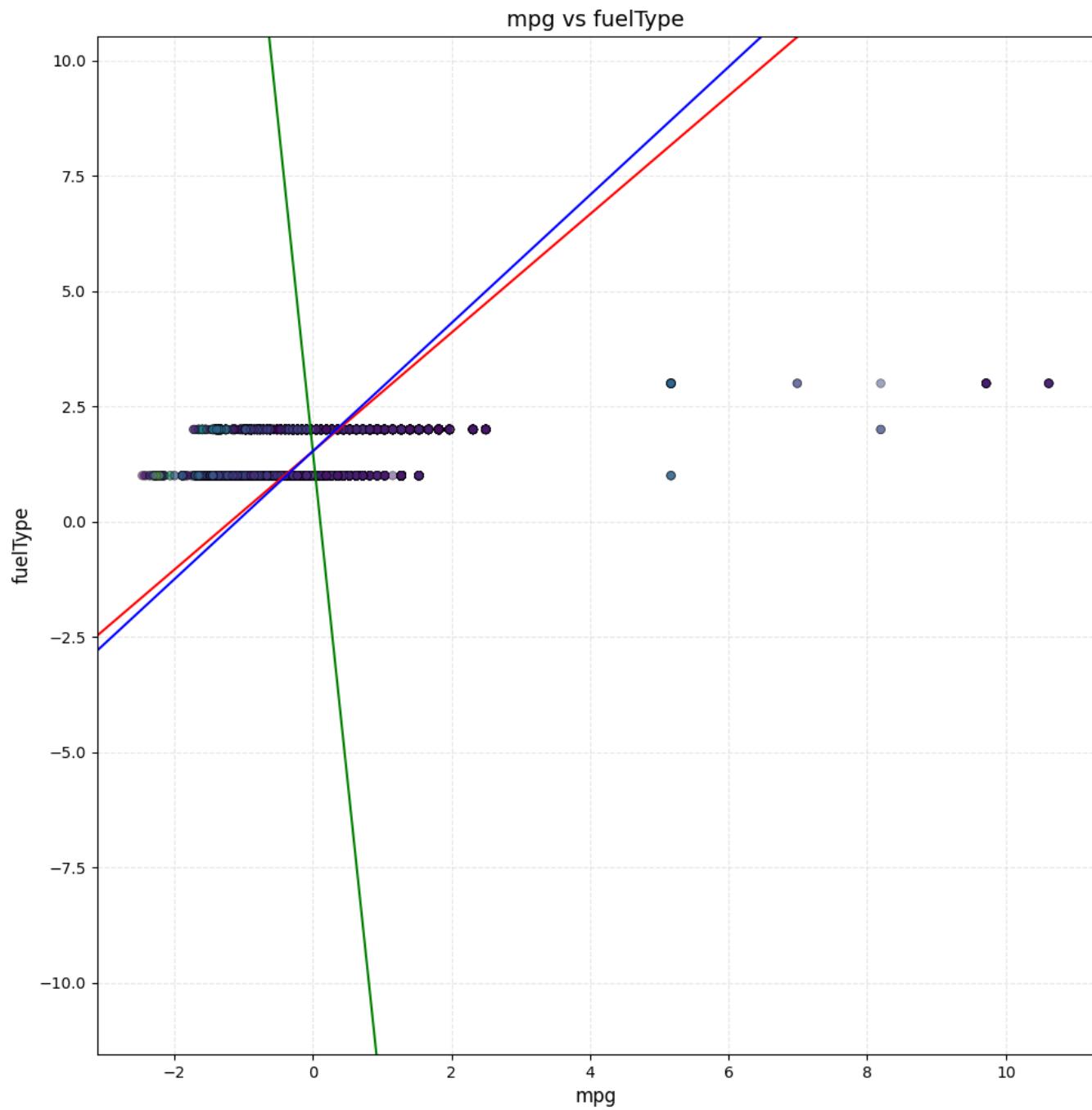
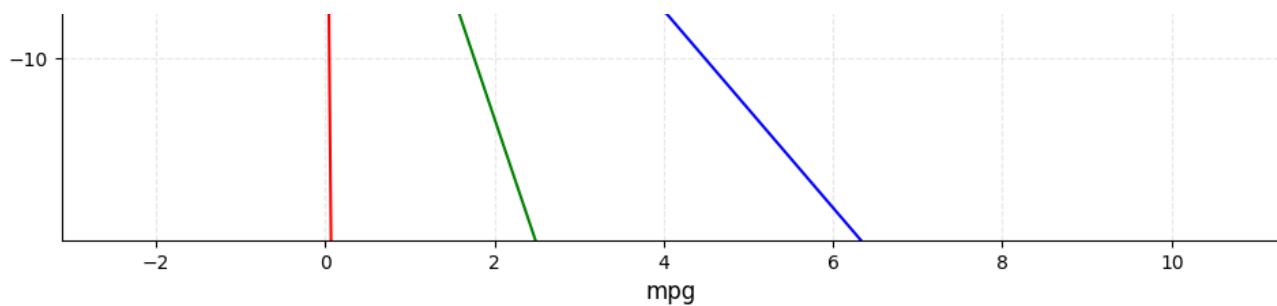
mpg vs transmission

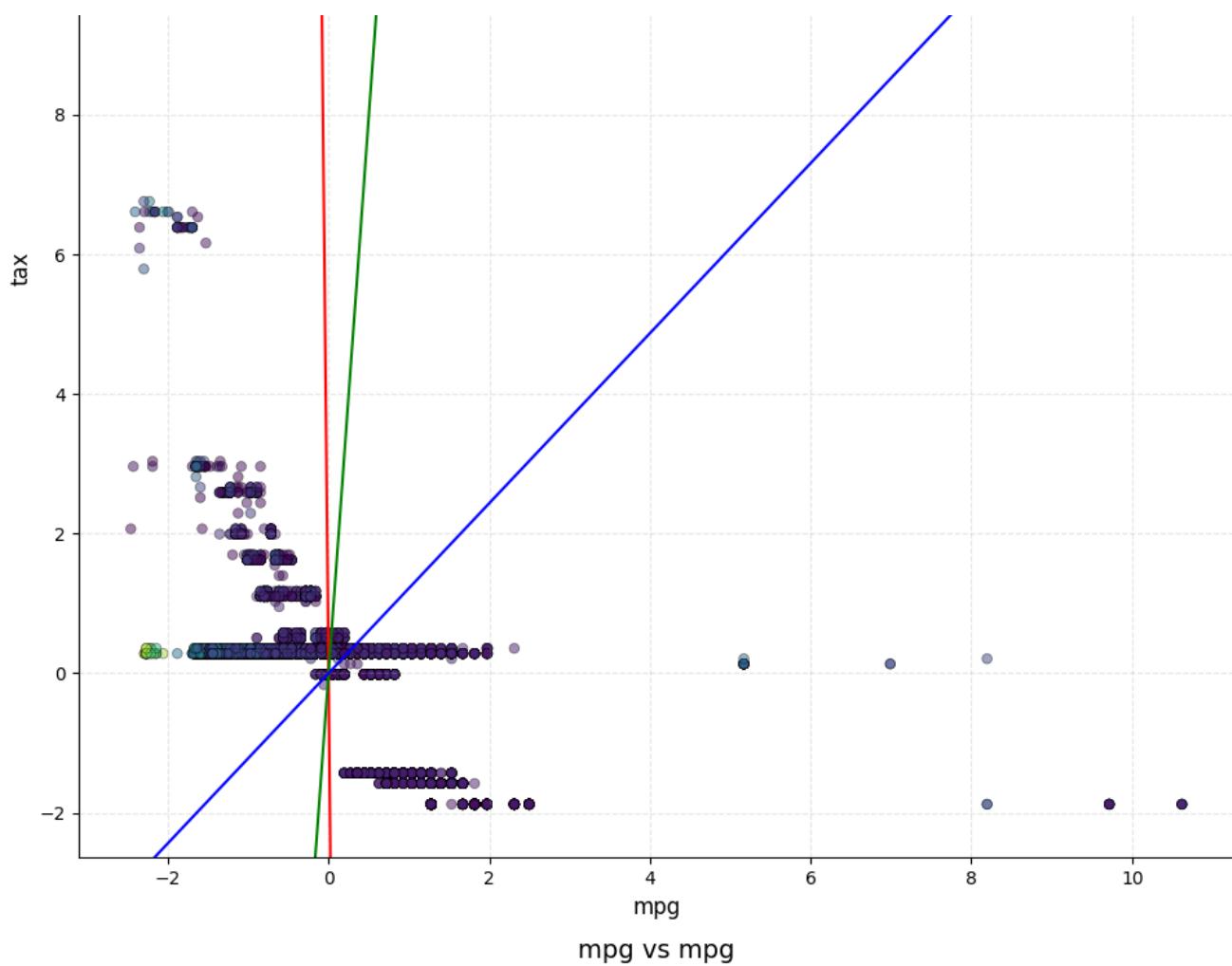




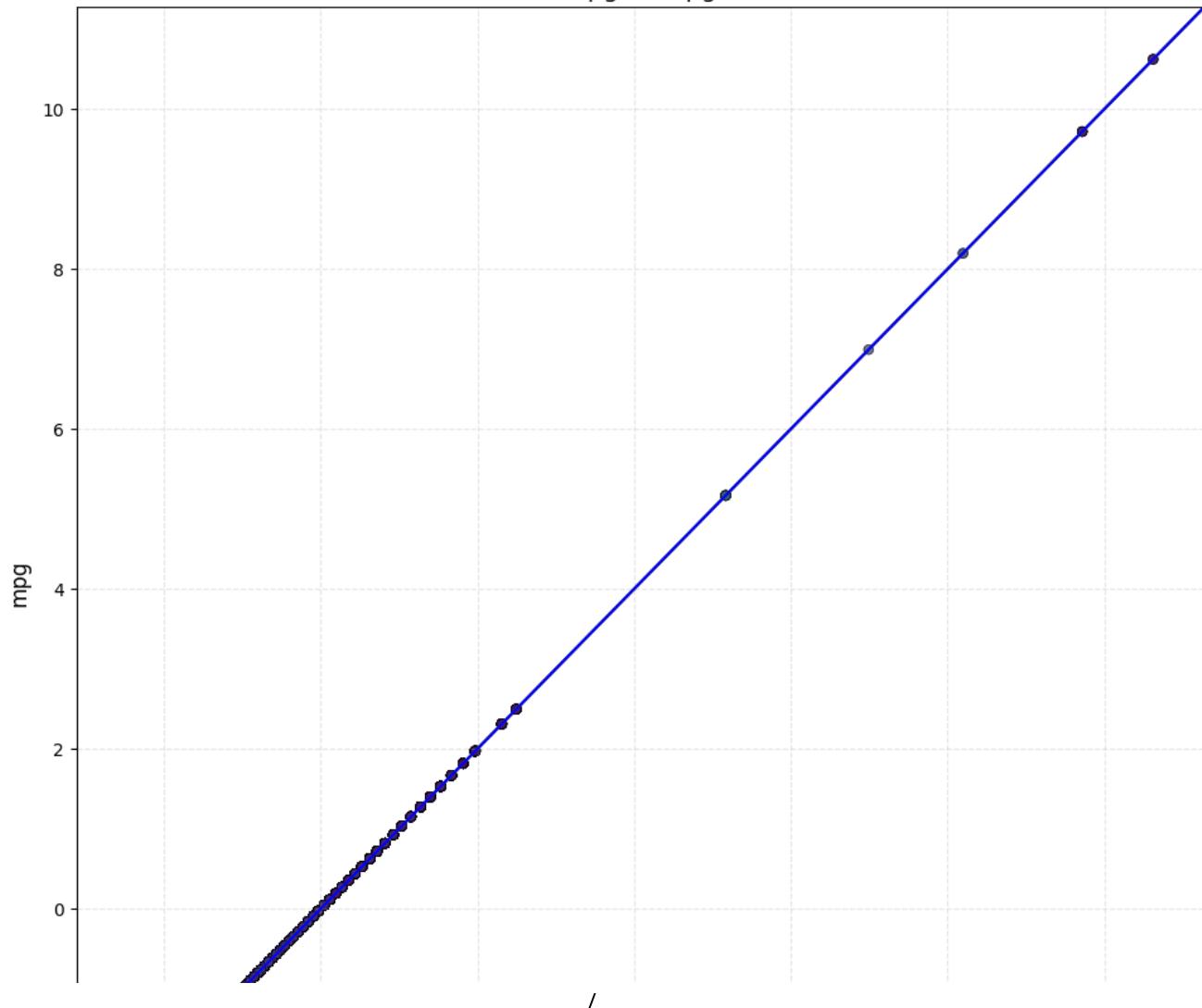
mpg vs mileage

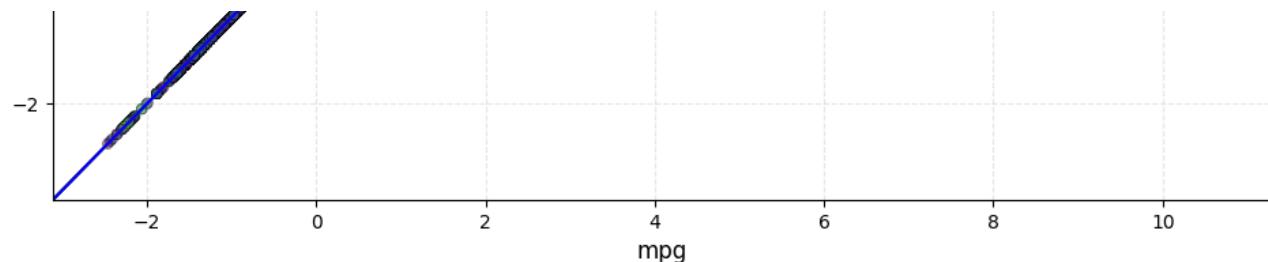




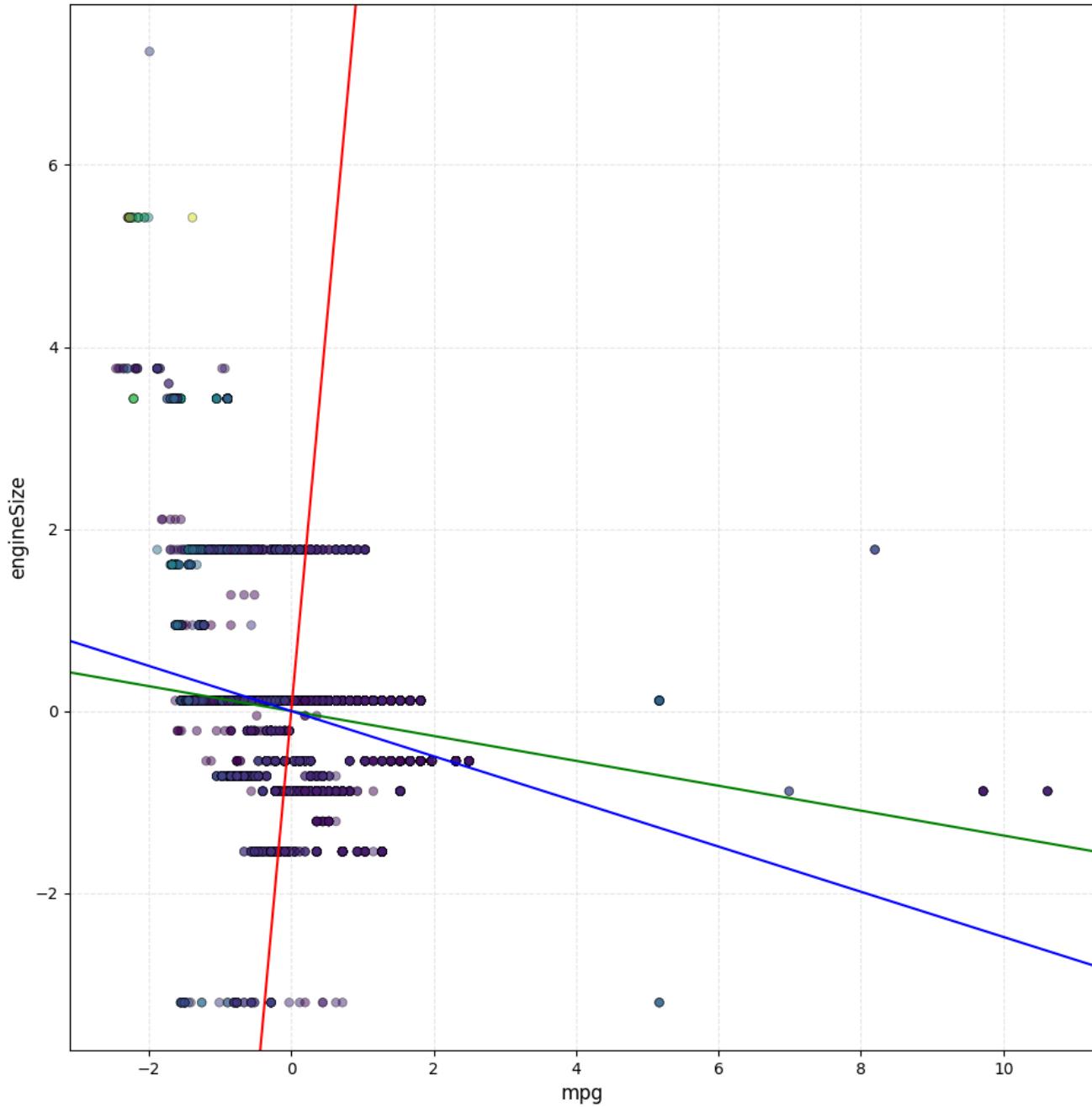


mpg vs mpg

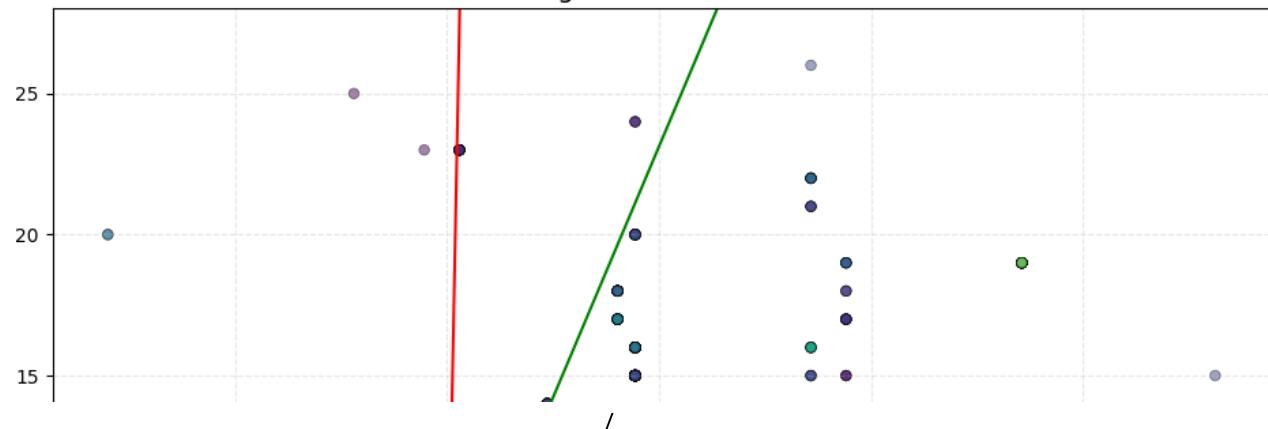




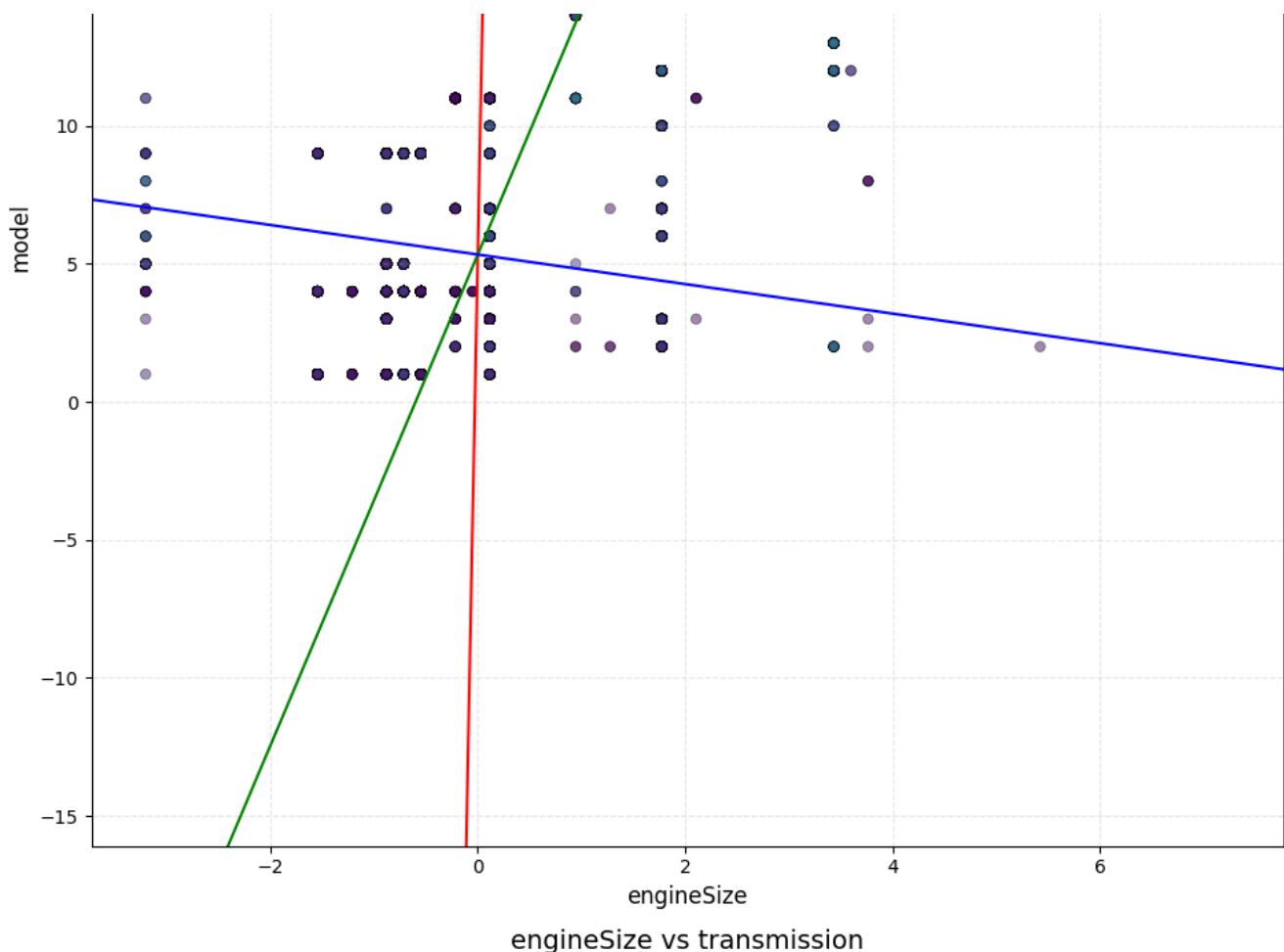
mpg vs engineSize



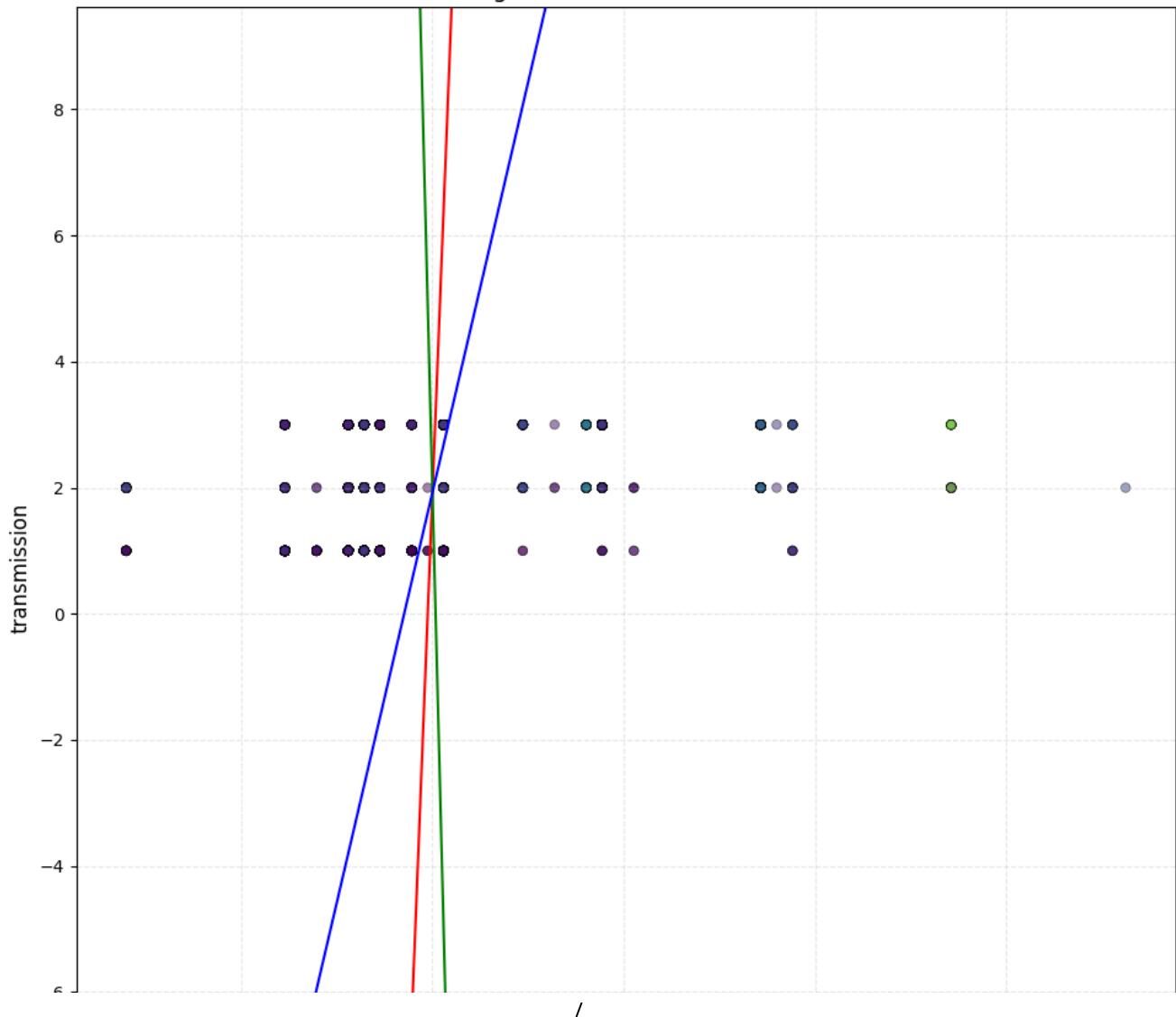
engineSize vs mpg

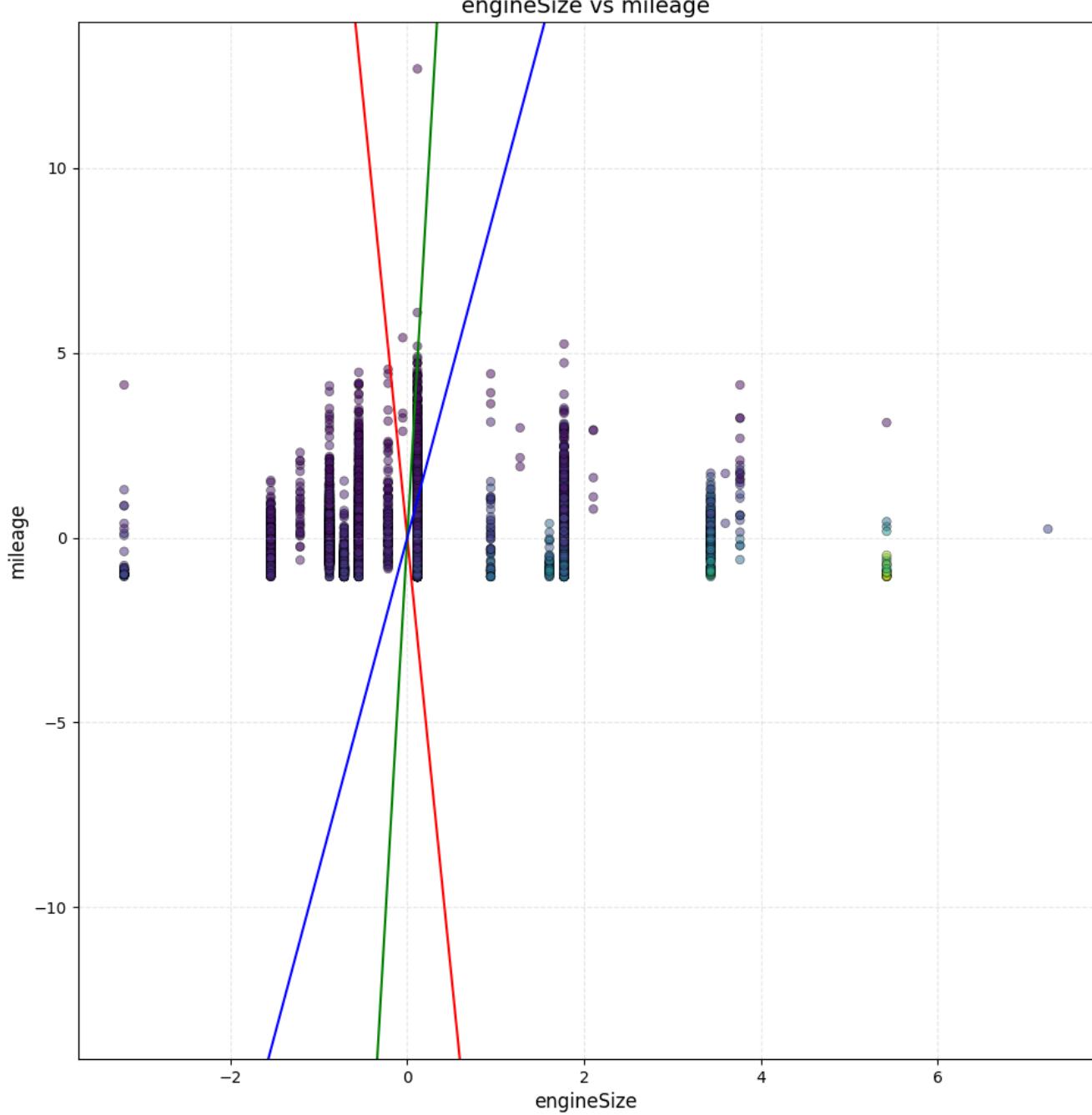
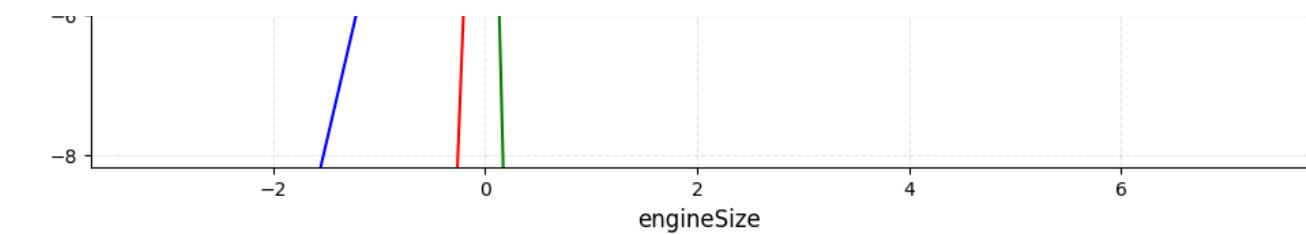


/

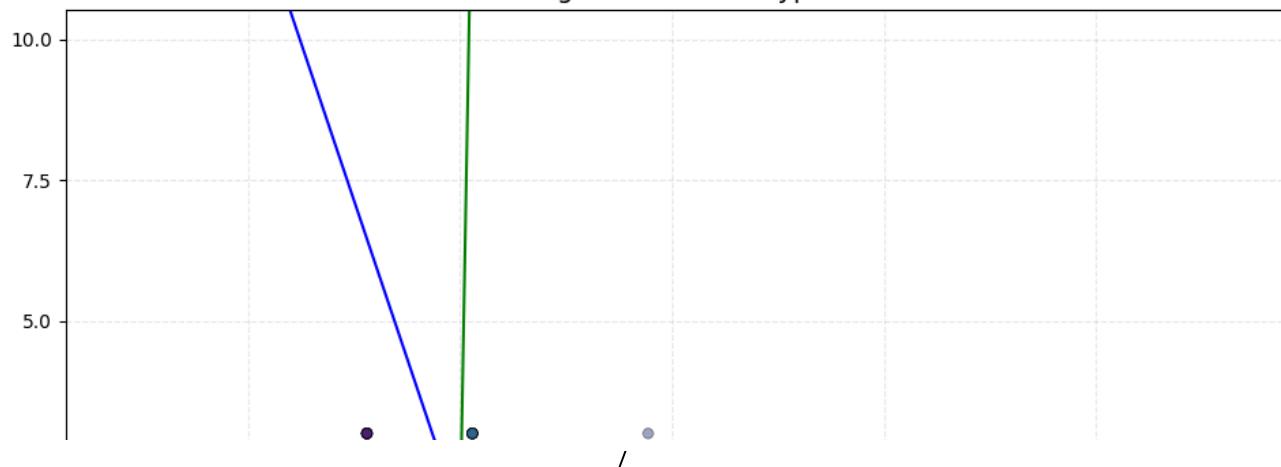


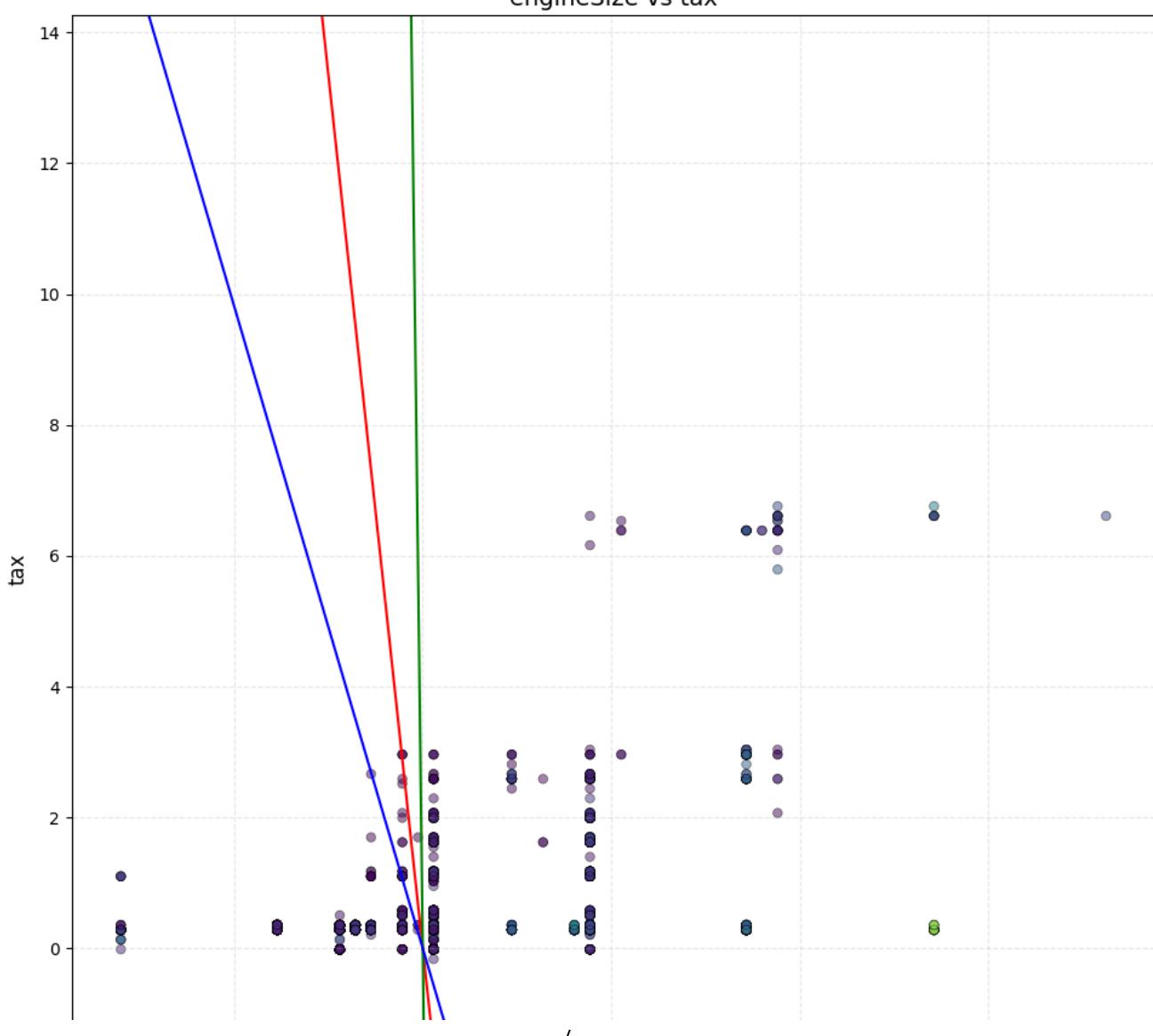
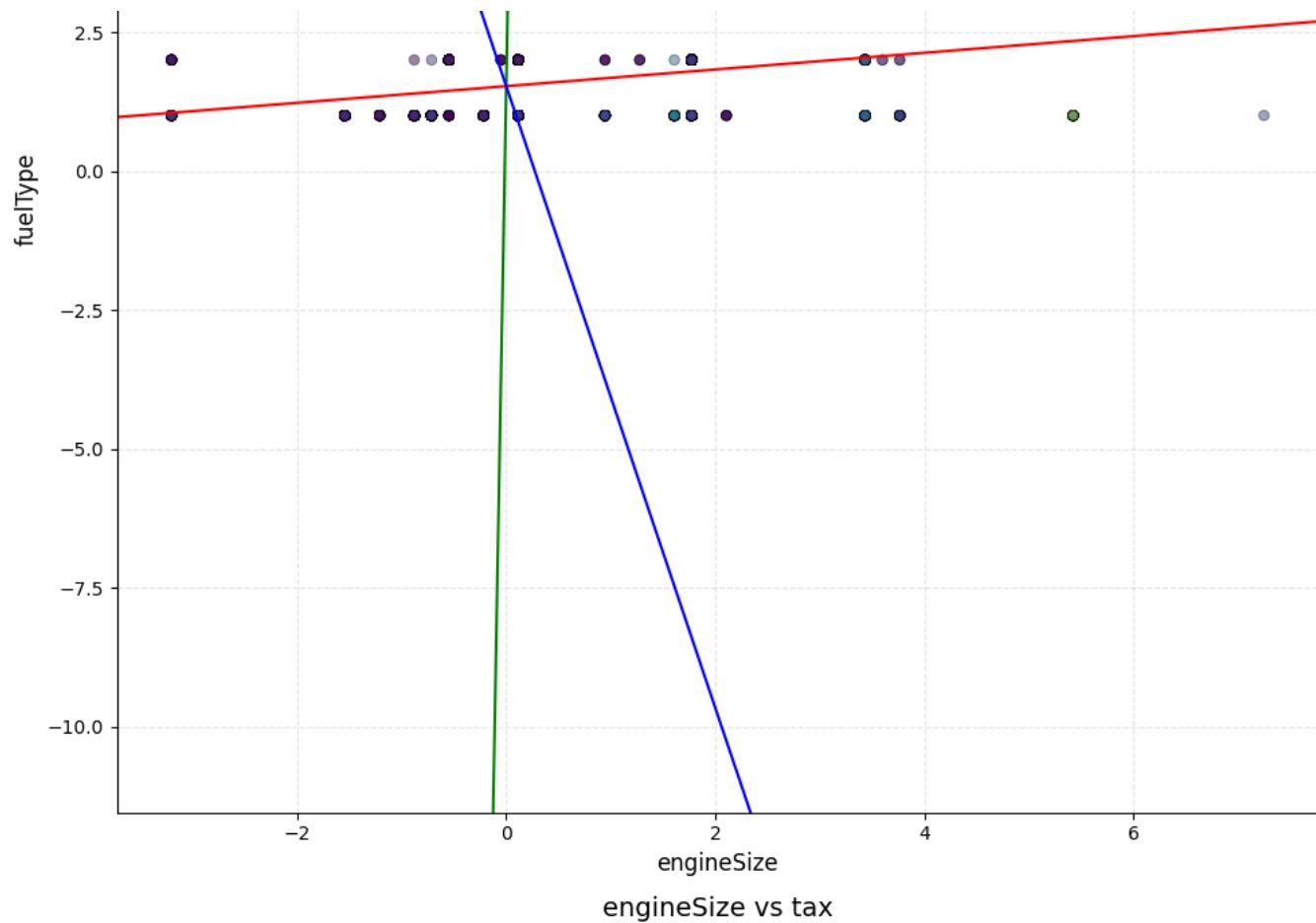
engineSize vs transmission

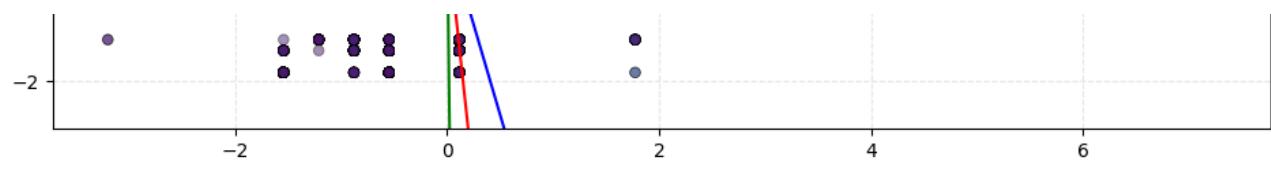




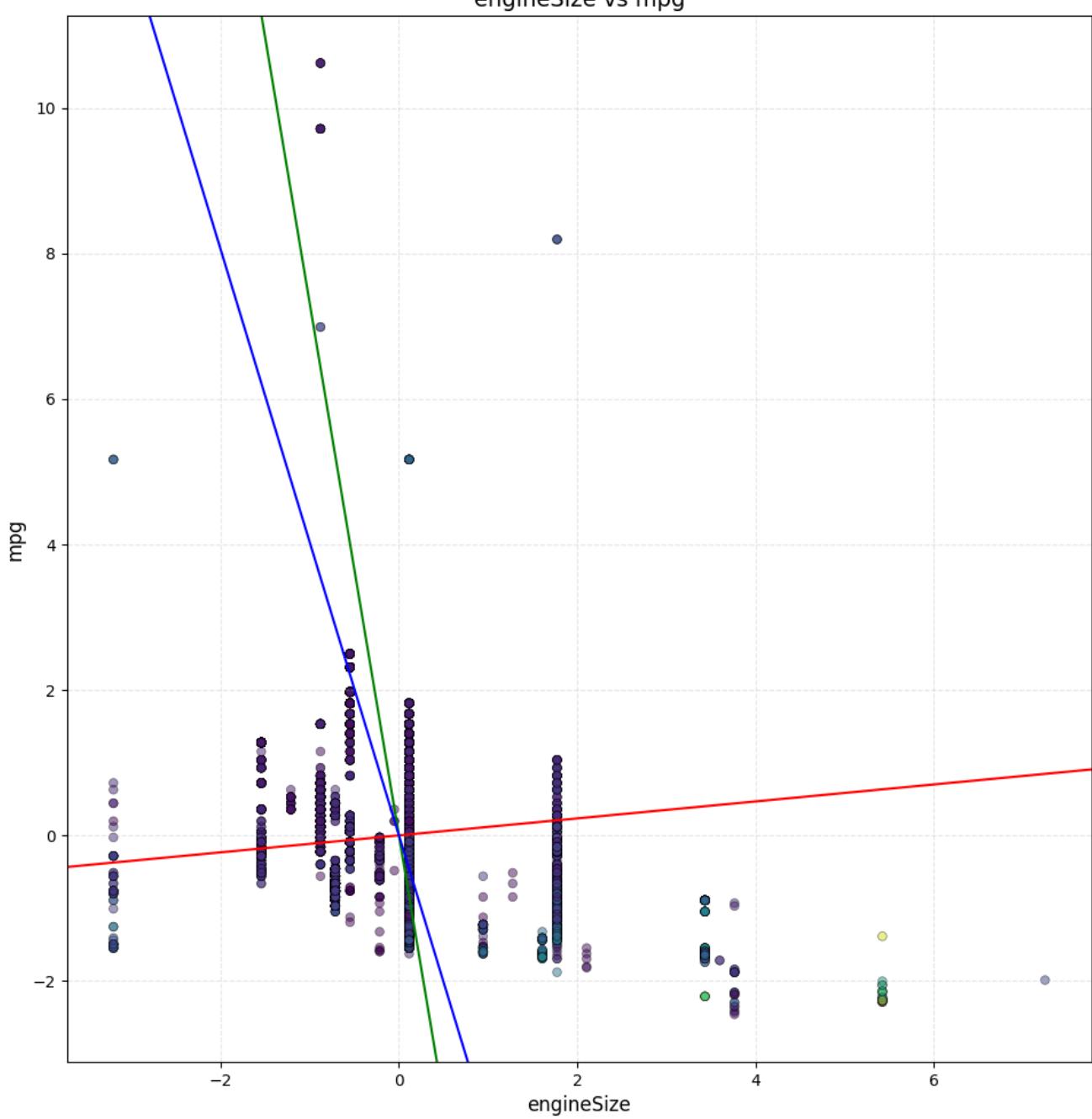
engineSize vs fuelType



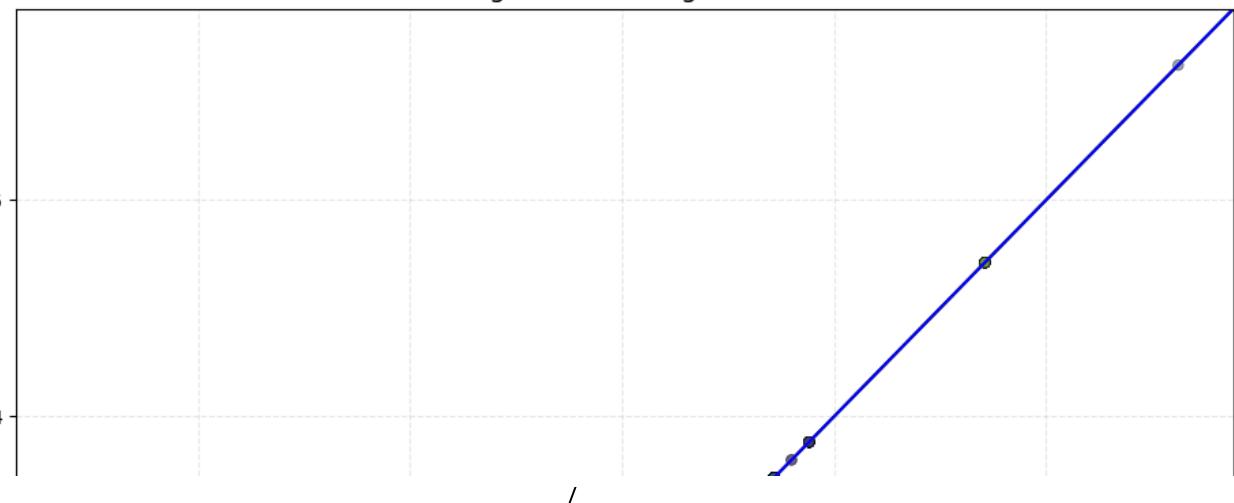




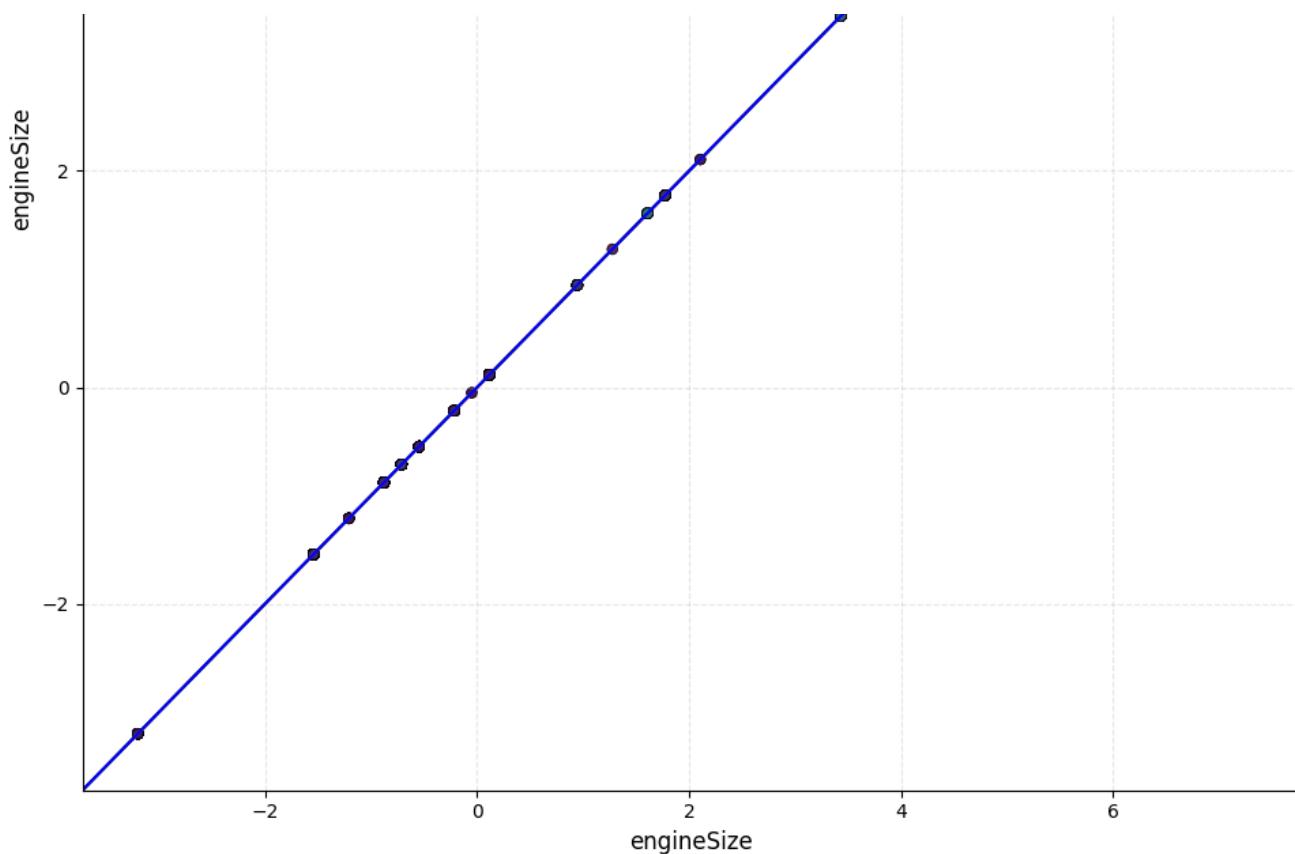
engineSize vs mpg



engineSize vs mpg

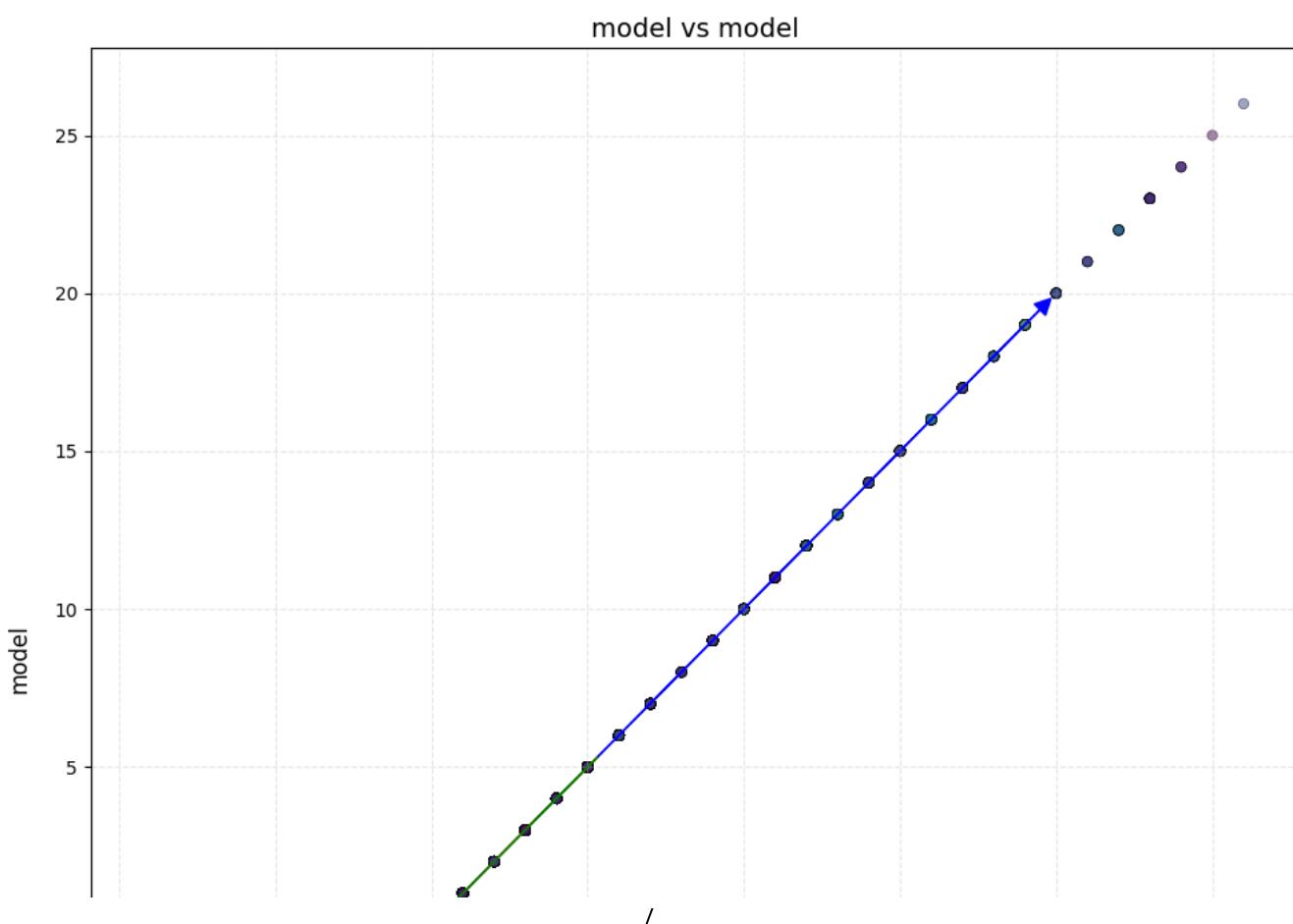


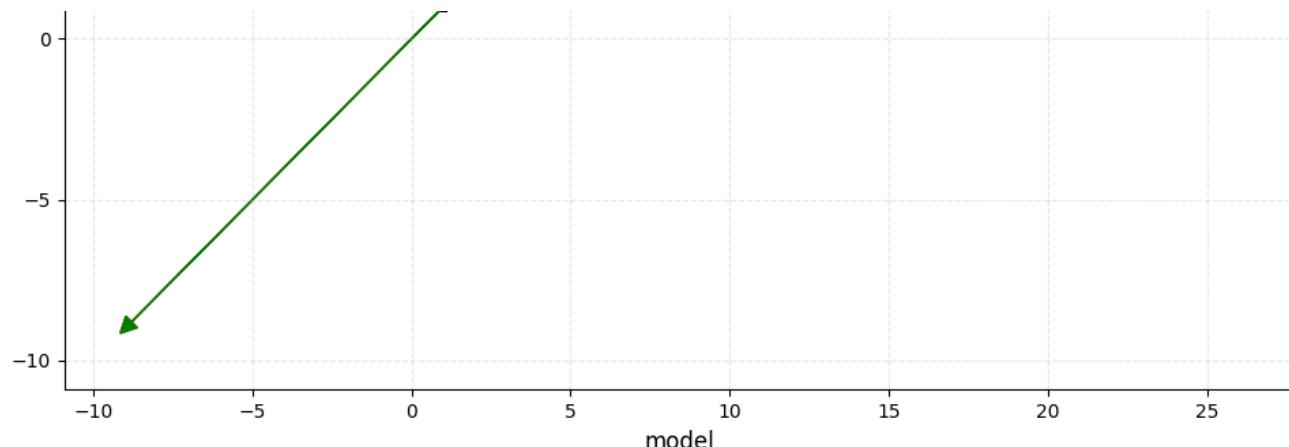
I



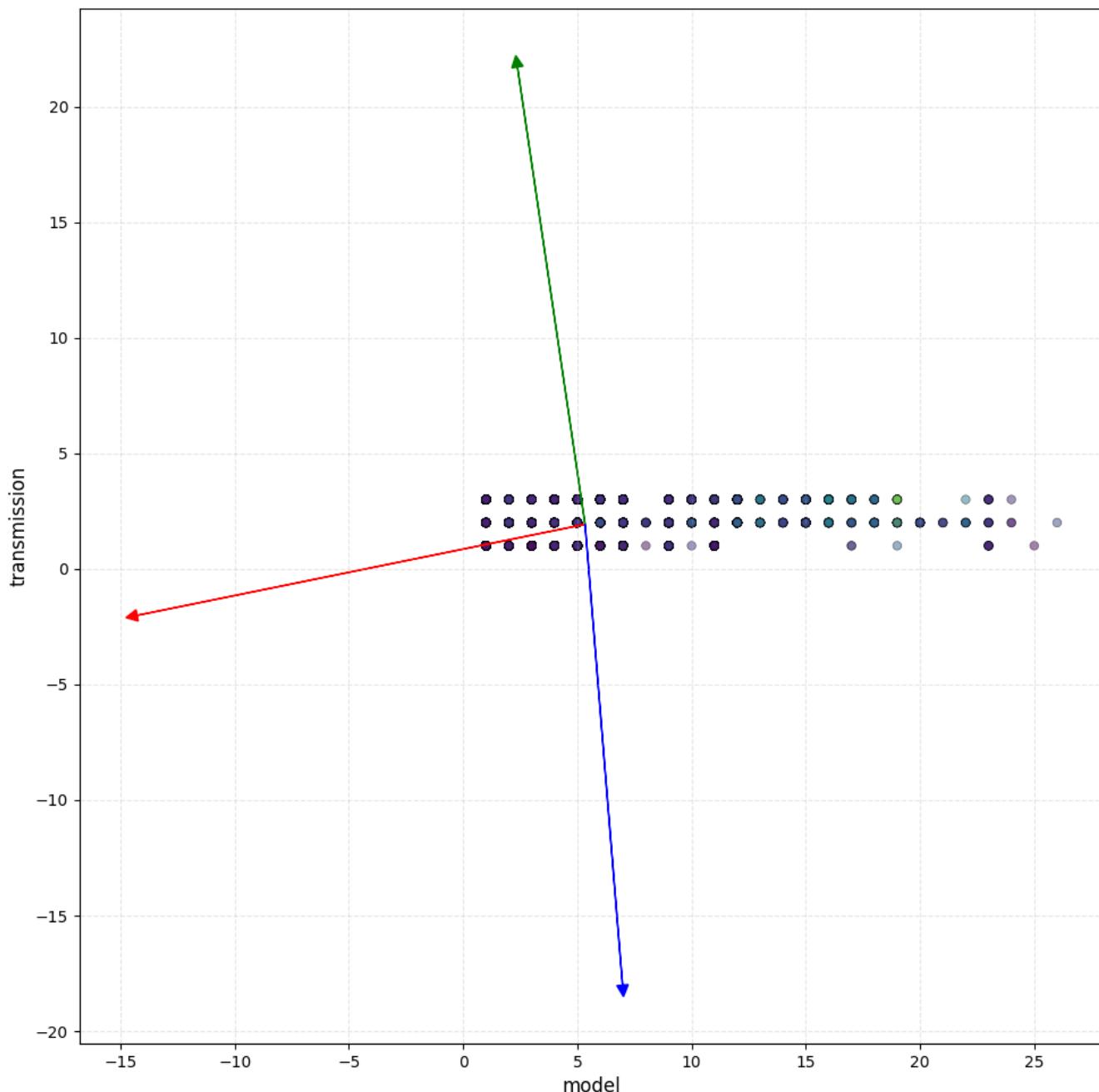
- Projected the principal components onto the pair plots as vectors. This visualization technique helps understand how each principal component contributes to the overall variance and provides a geometric interpretation of their significance in the dataset.

### c. Interpretation of Vector Directions:

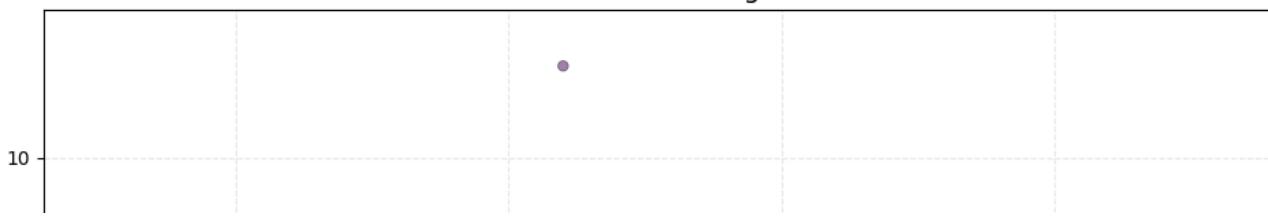


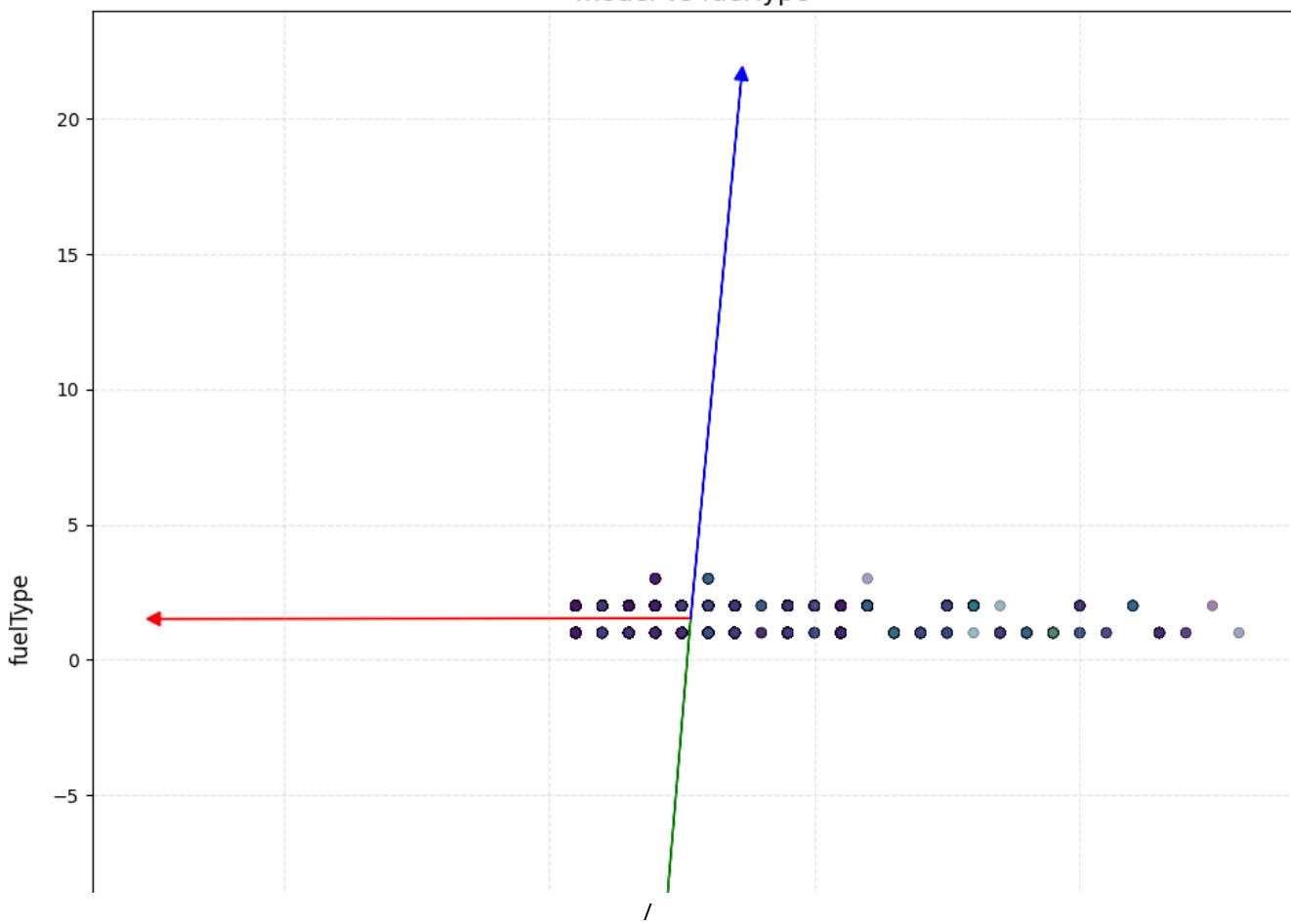
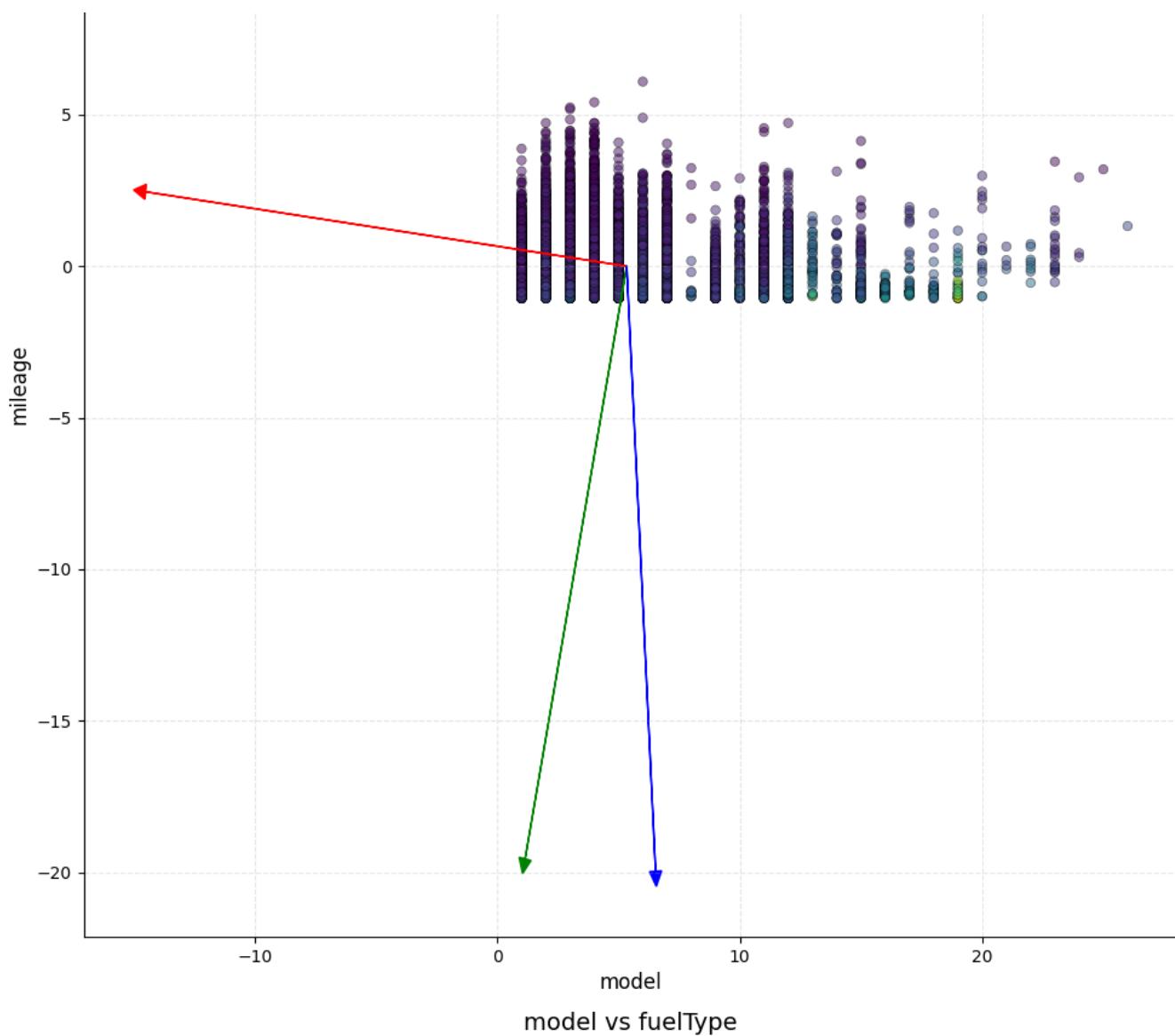


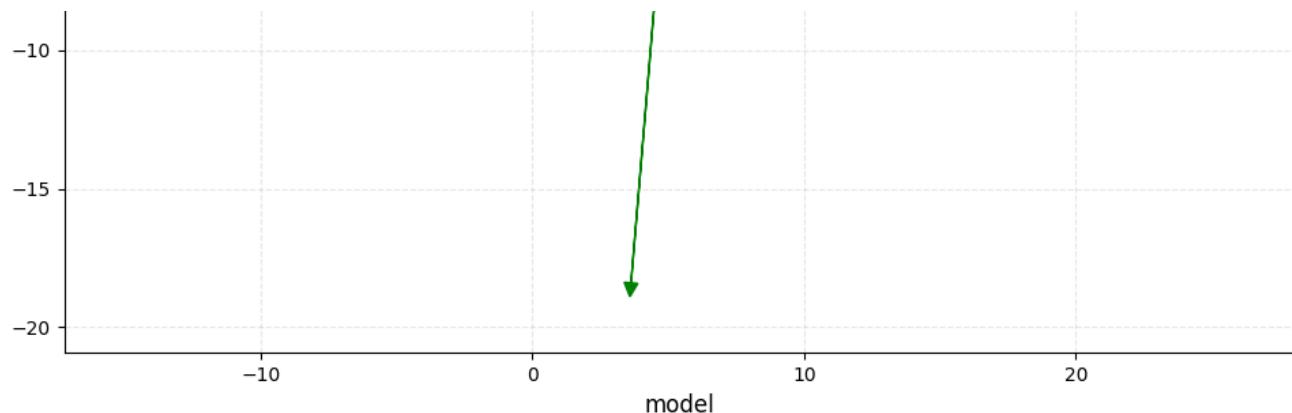
model vs transmission



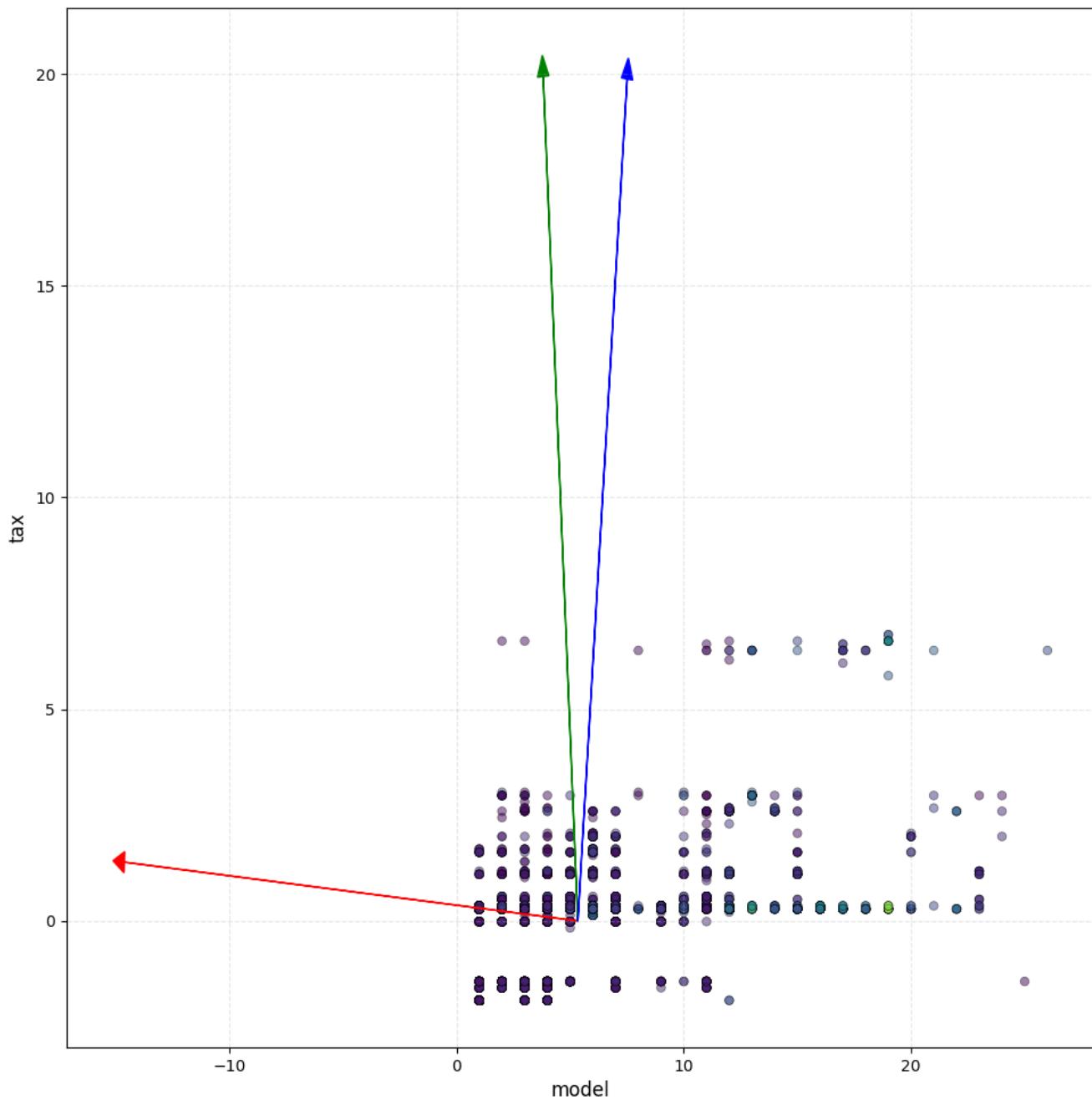
model vs mileage



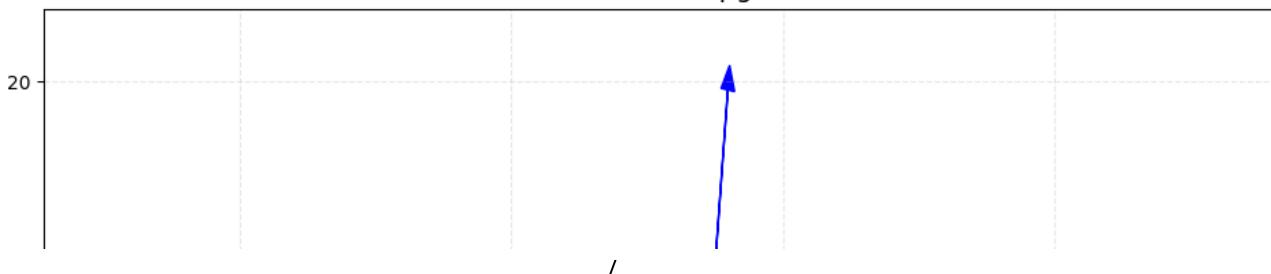


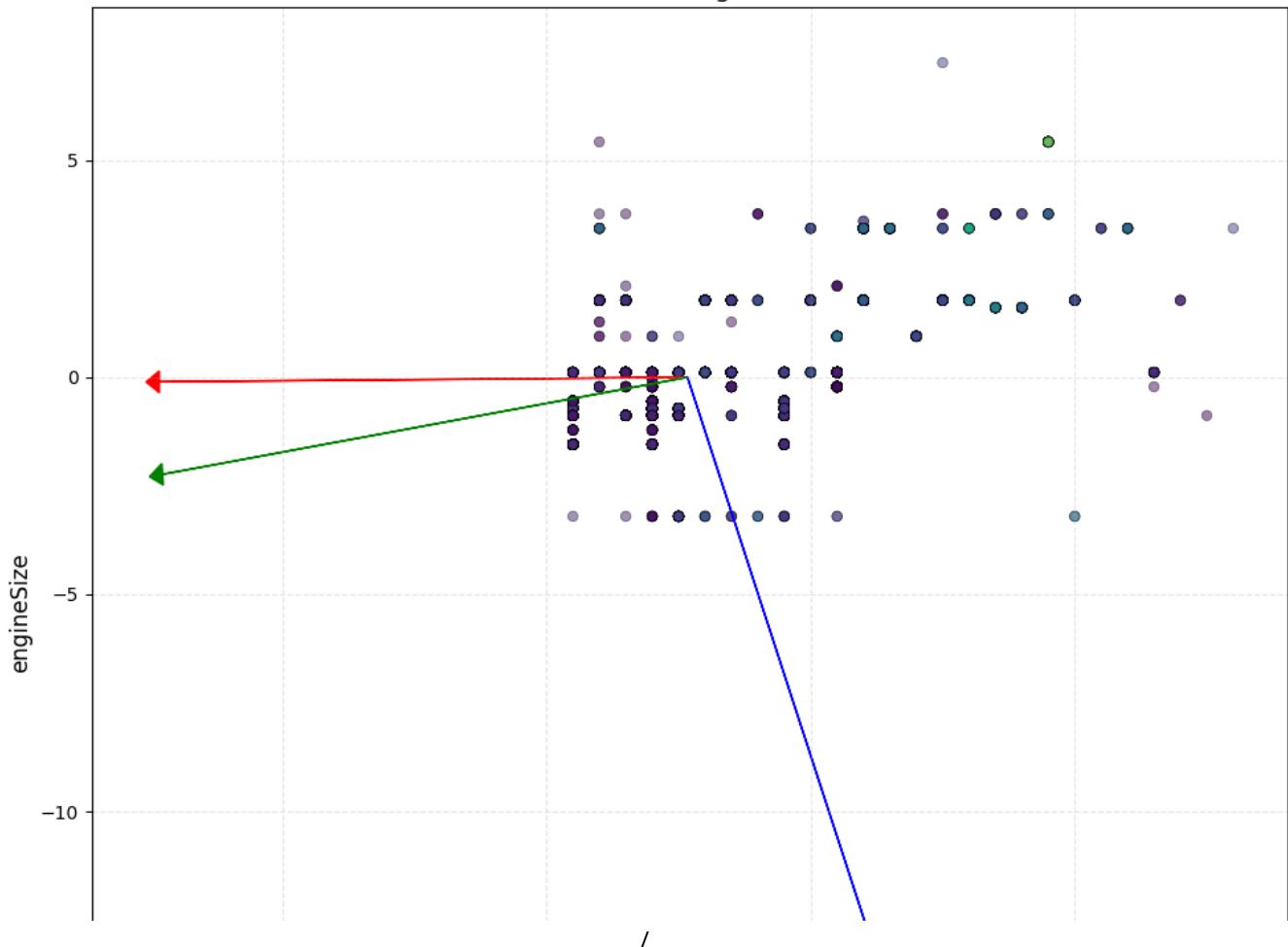
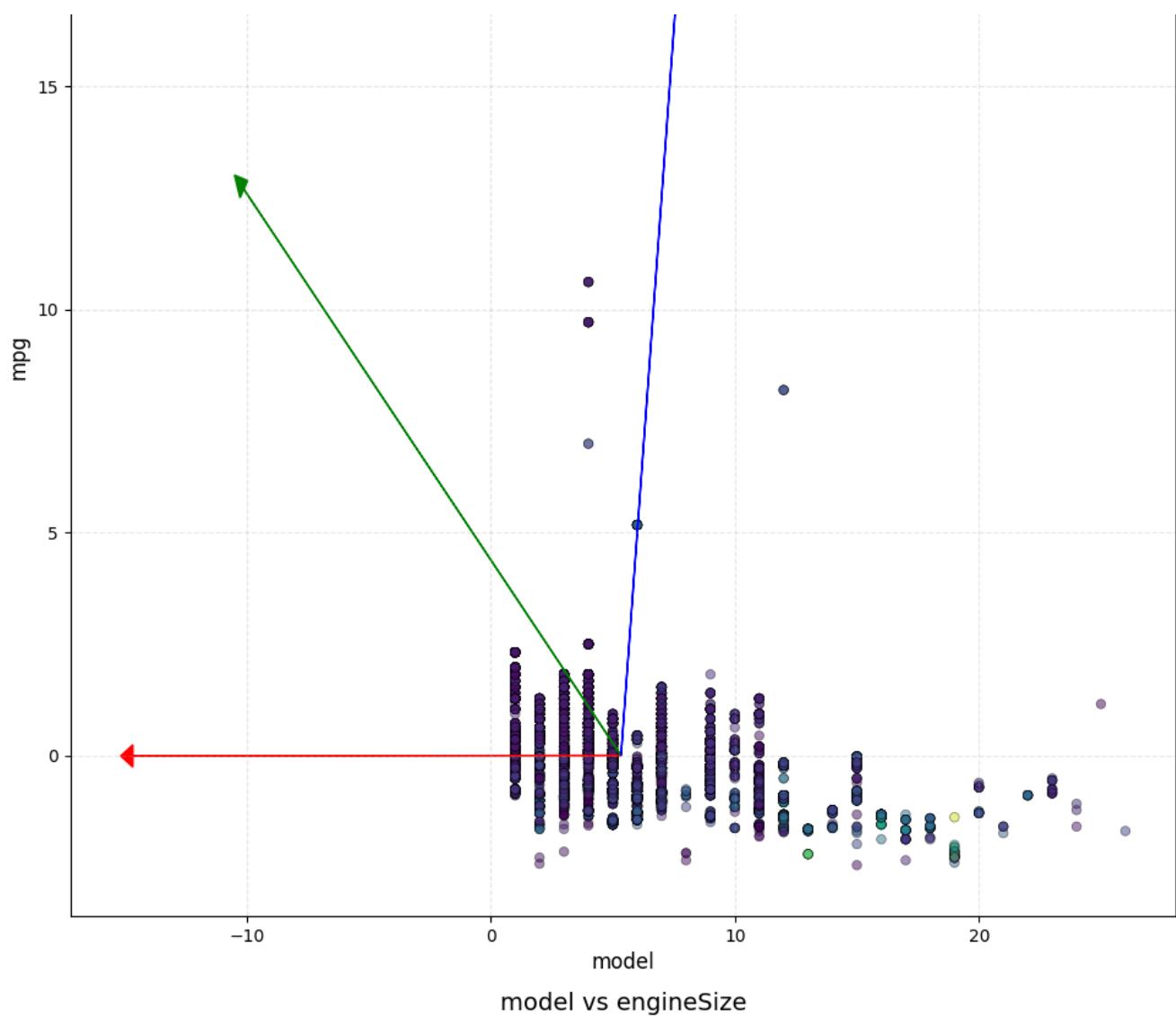


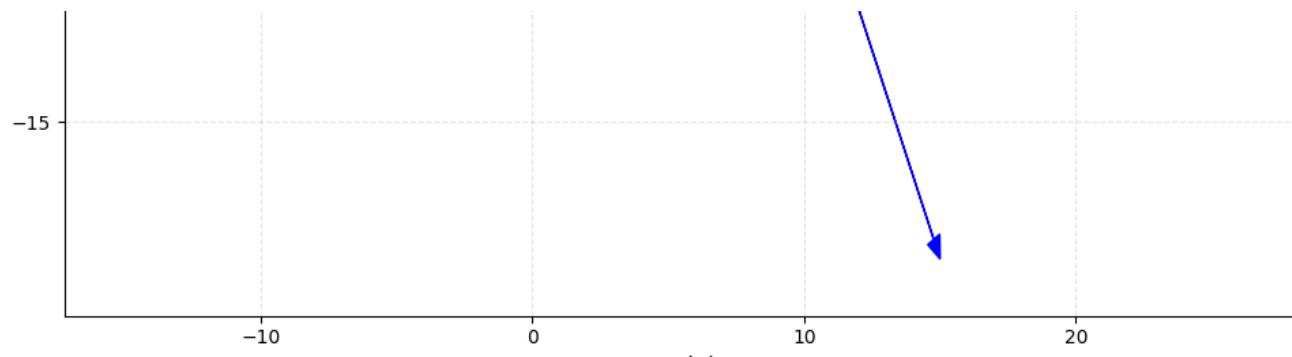
model vs tax



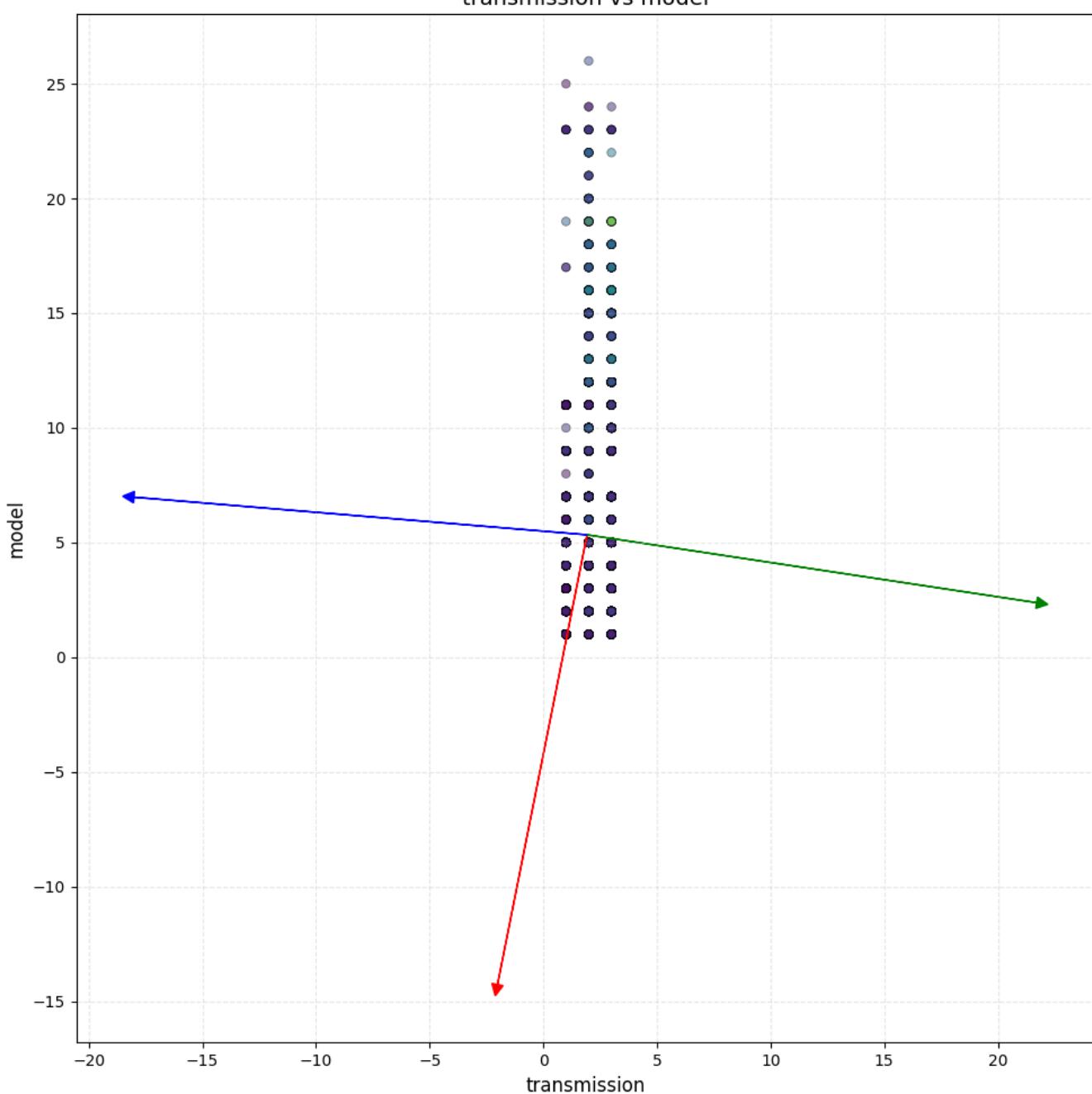
model vs mpg





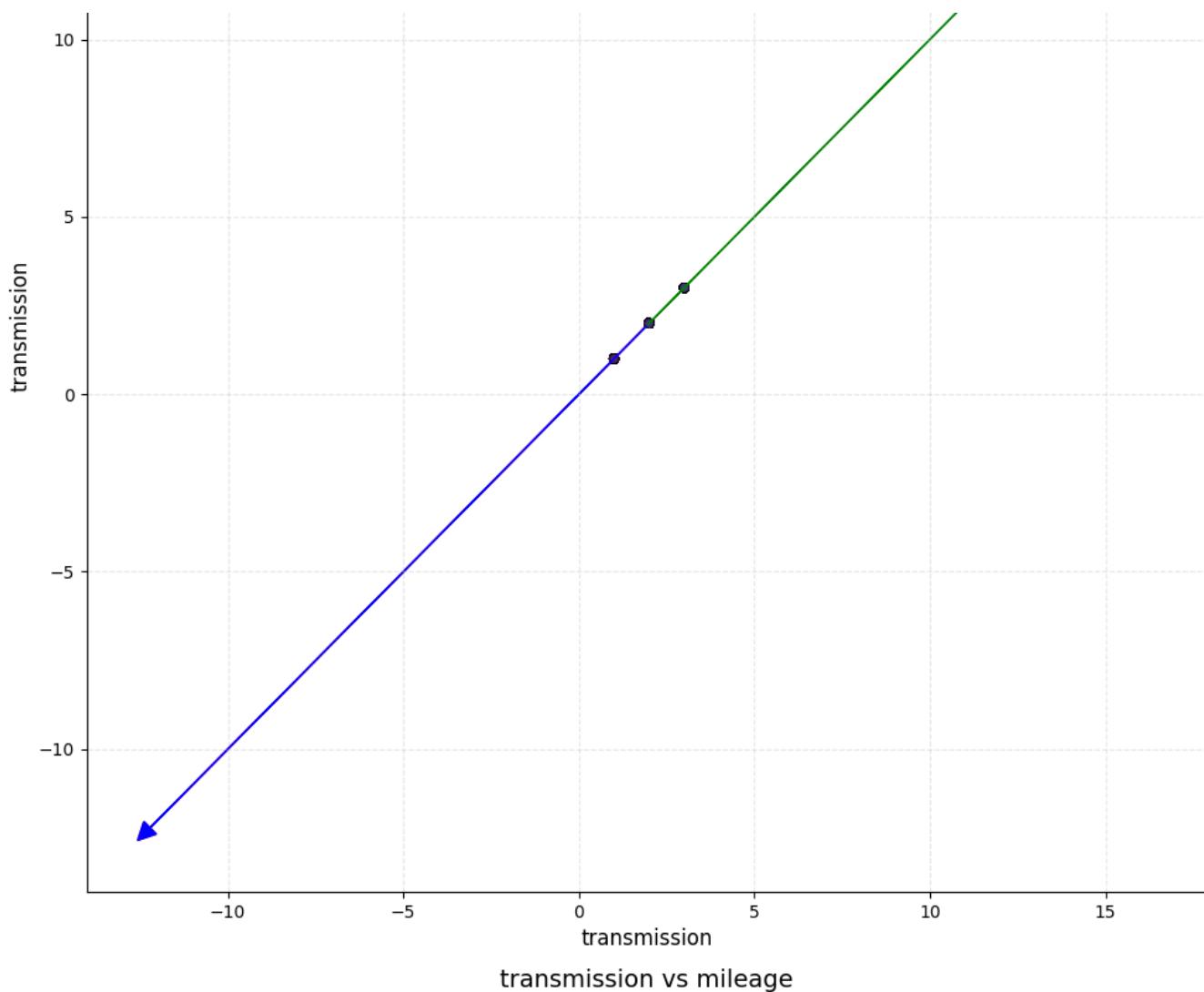


transmission vs model

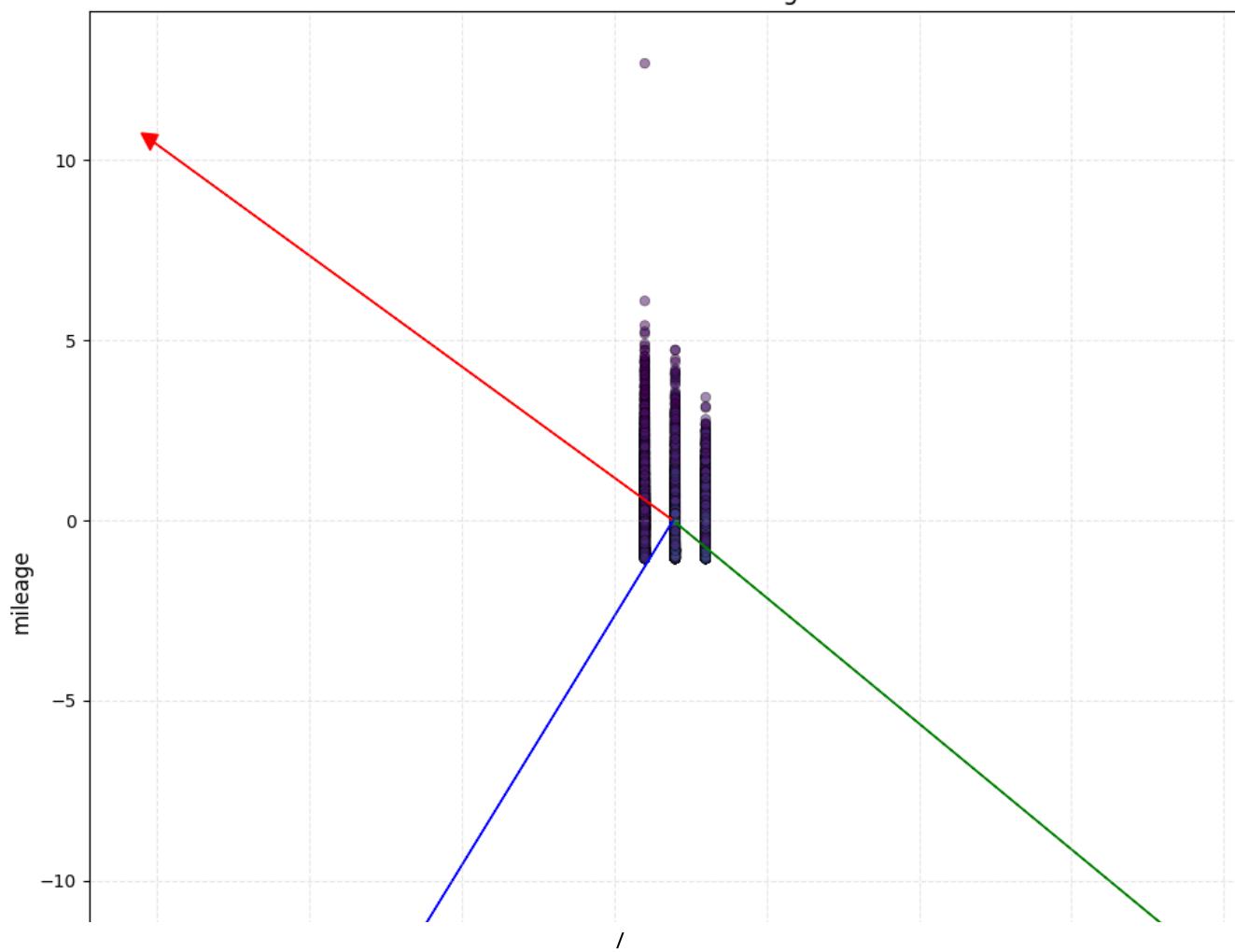


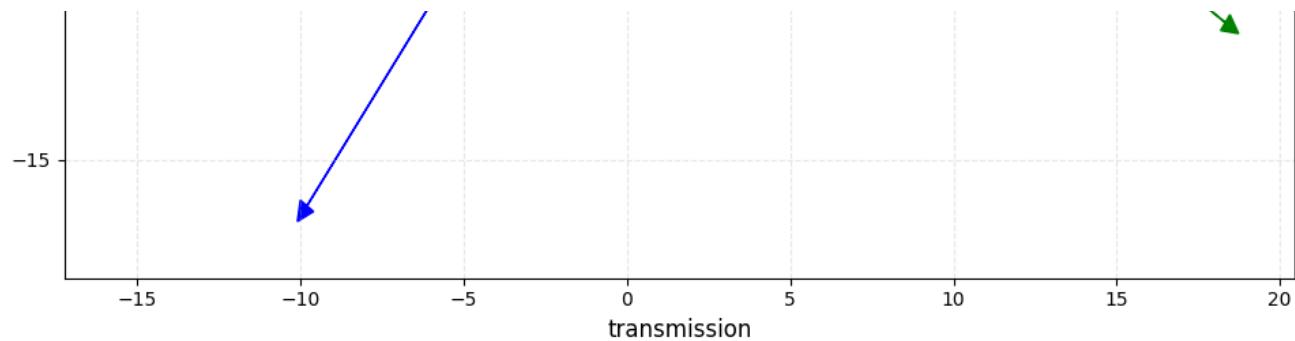
transmission vs model



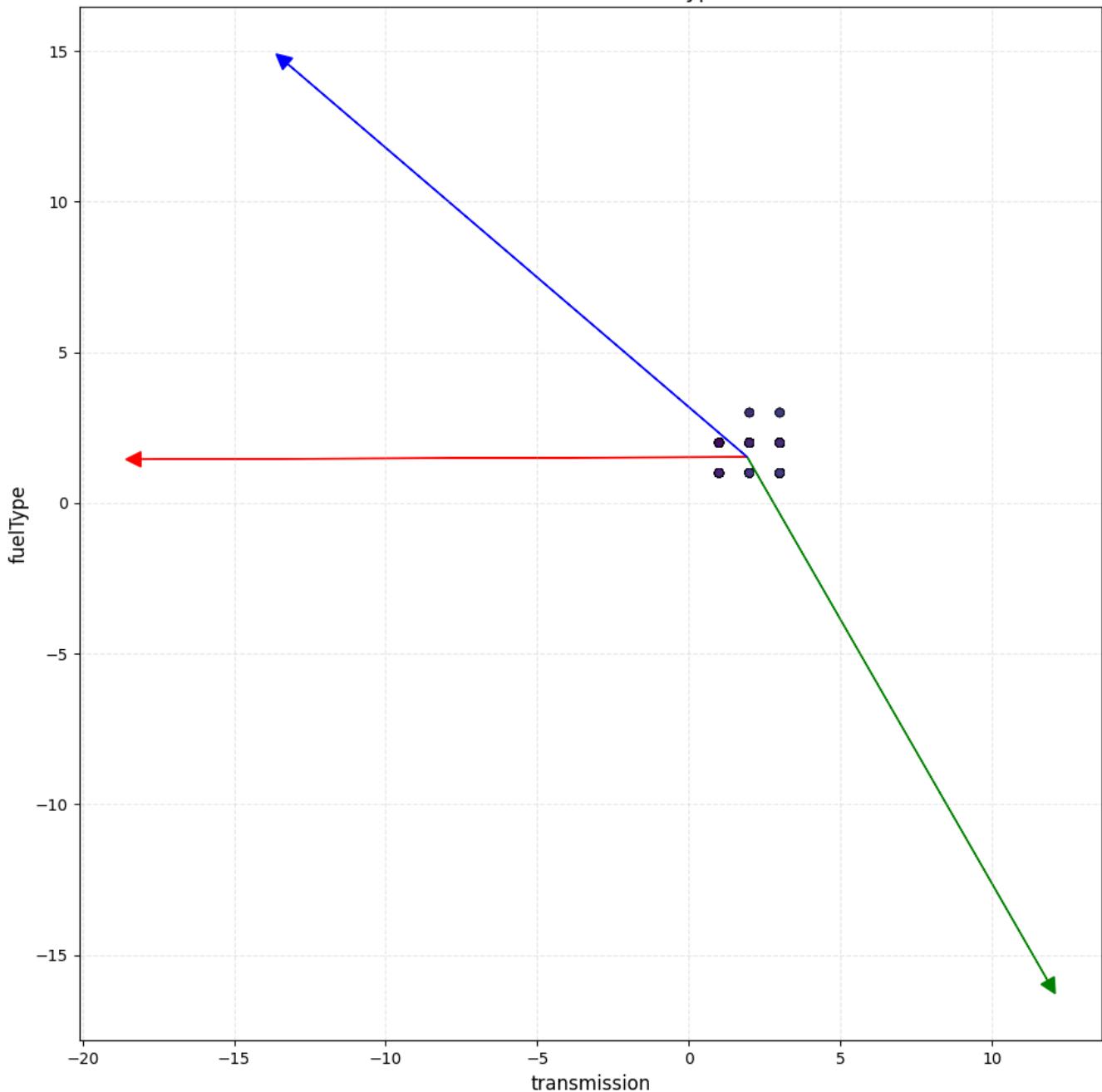


transmission vs mileage

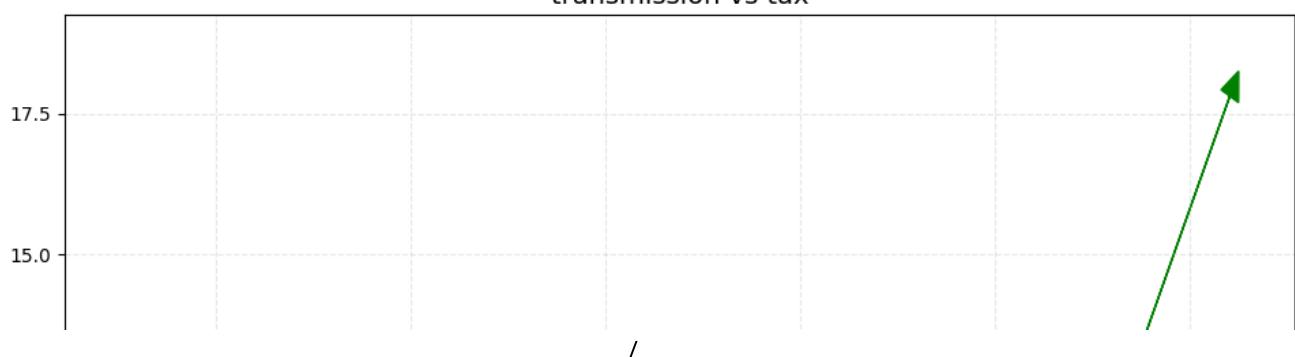


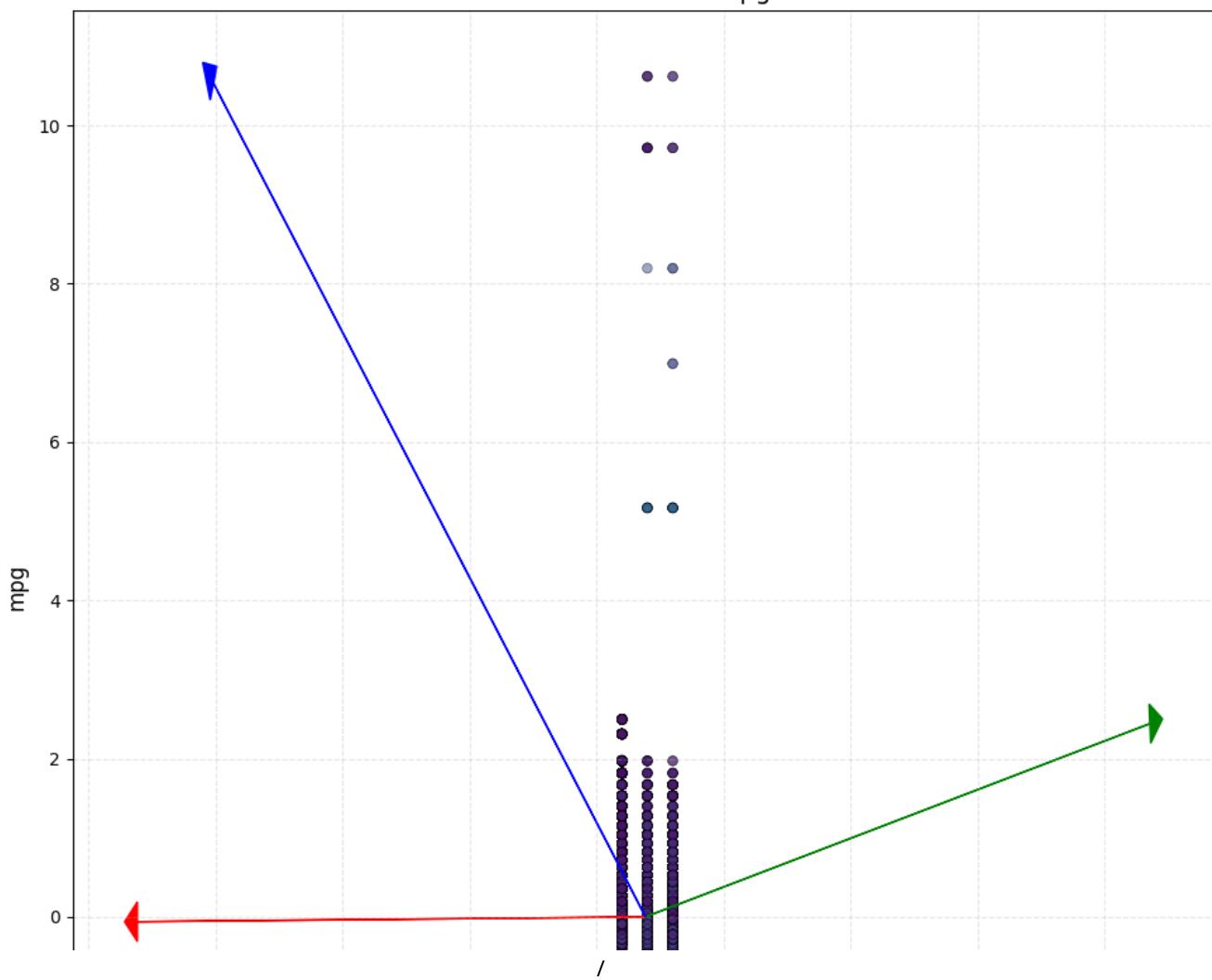
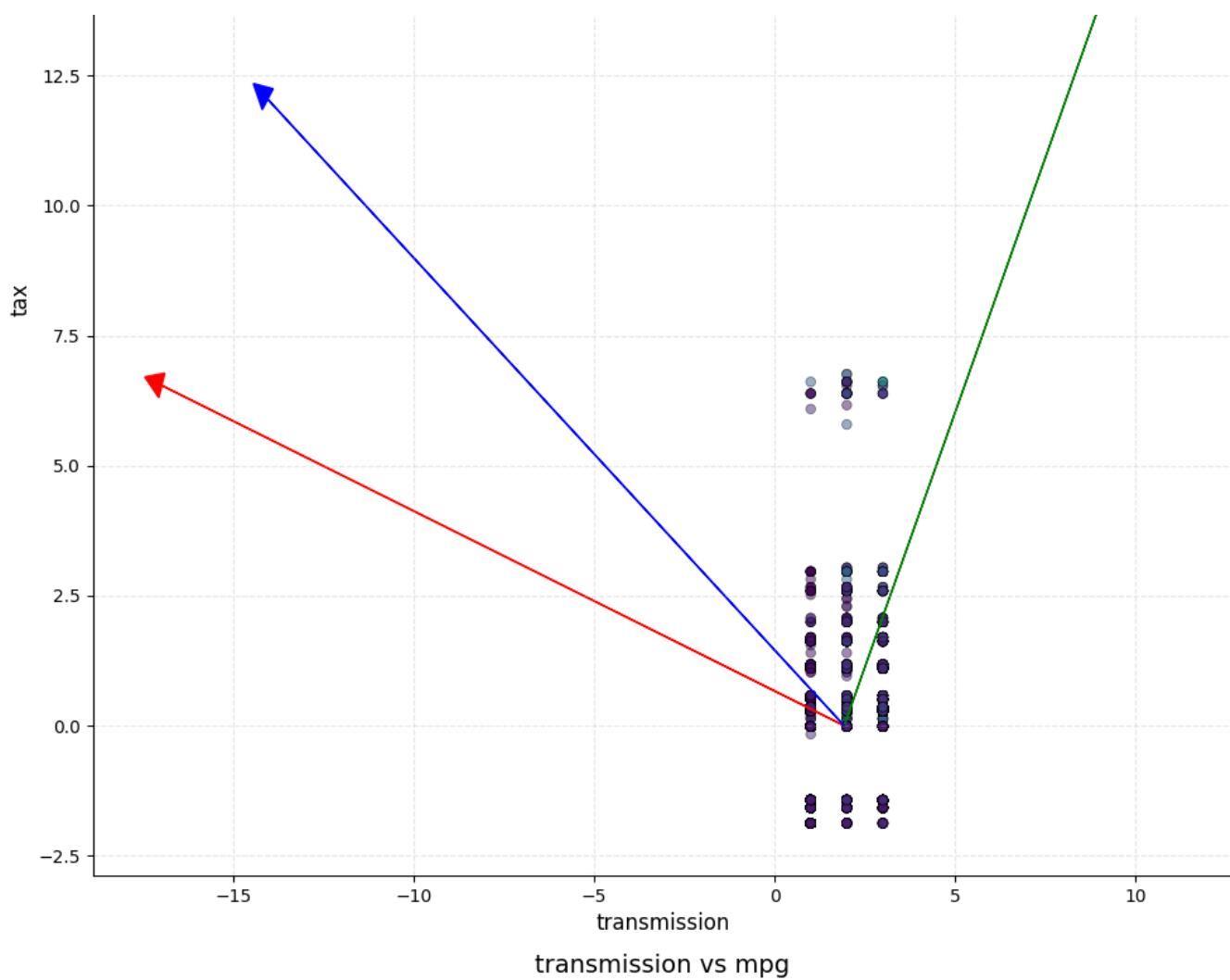


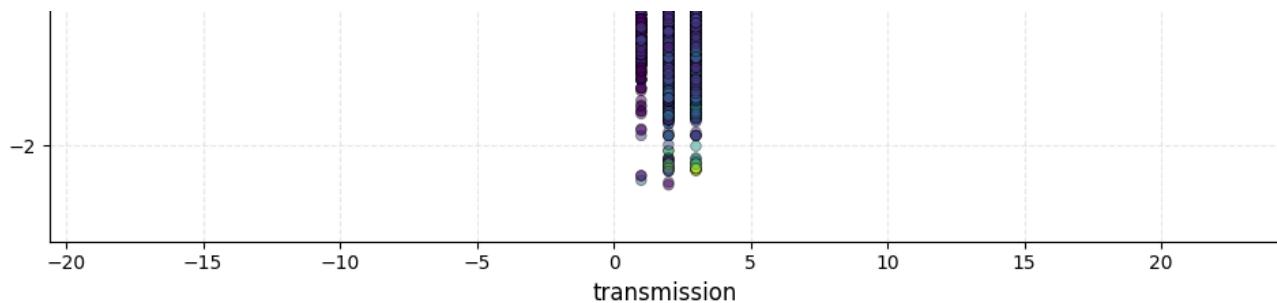
transmission vs fuelType



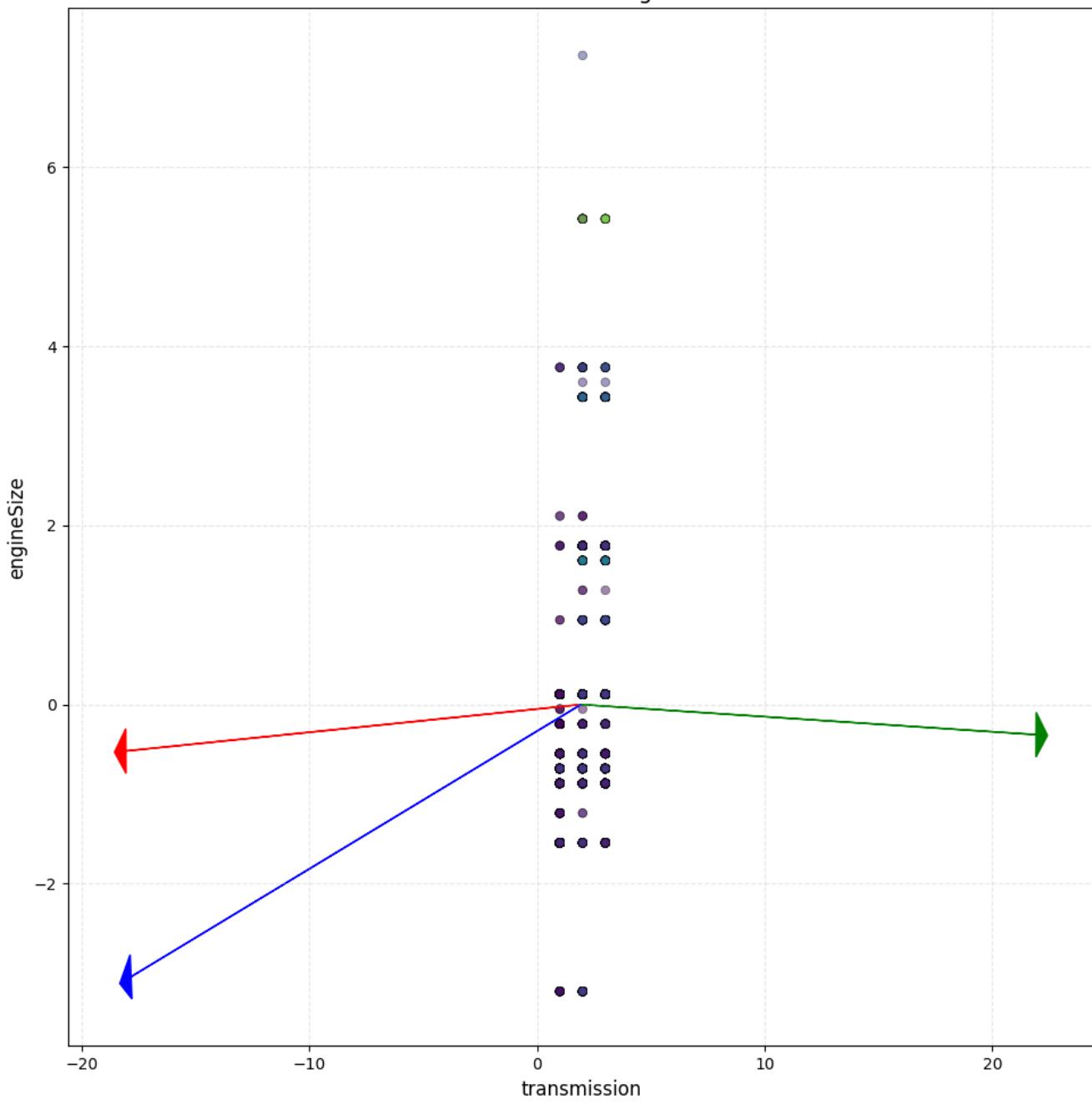
transmission vs tax



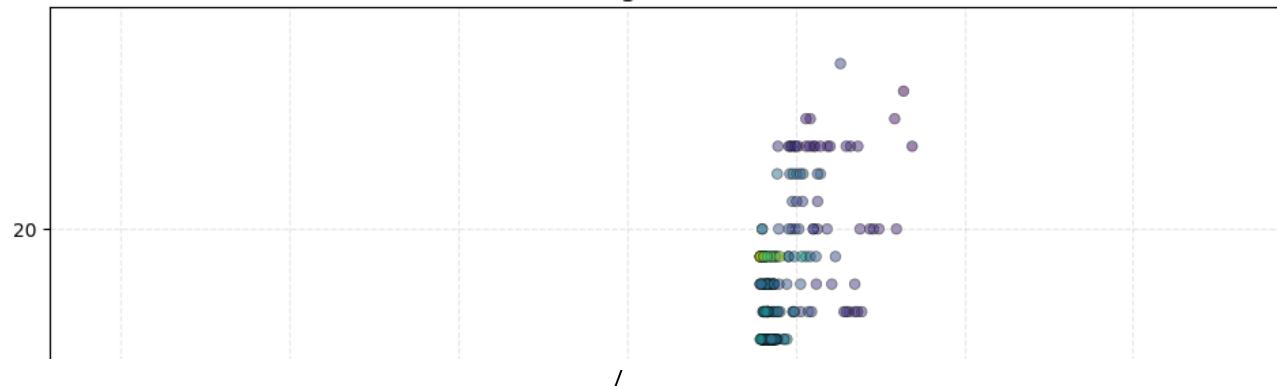


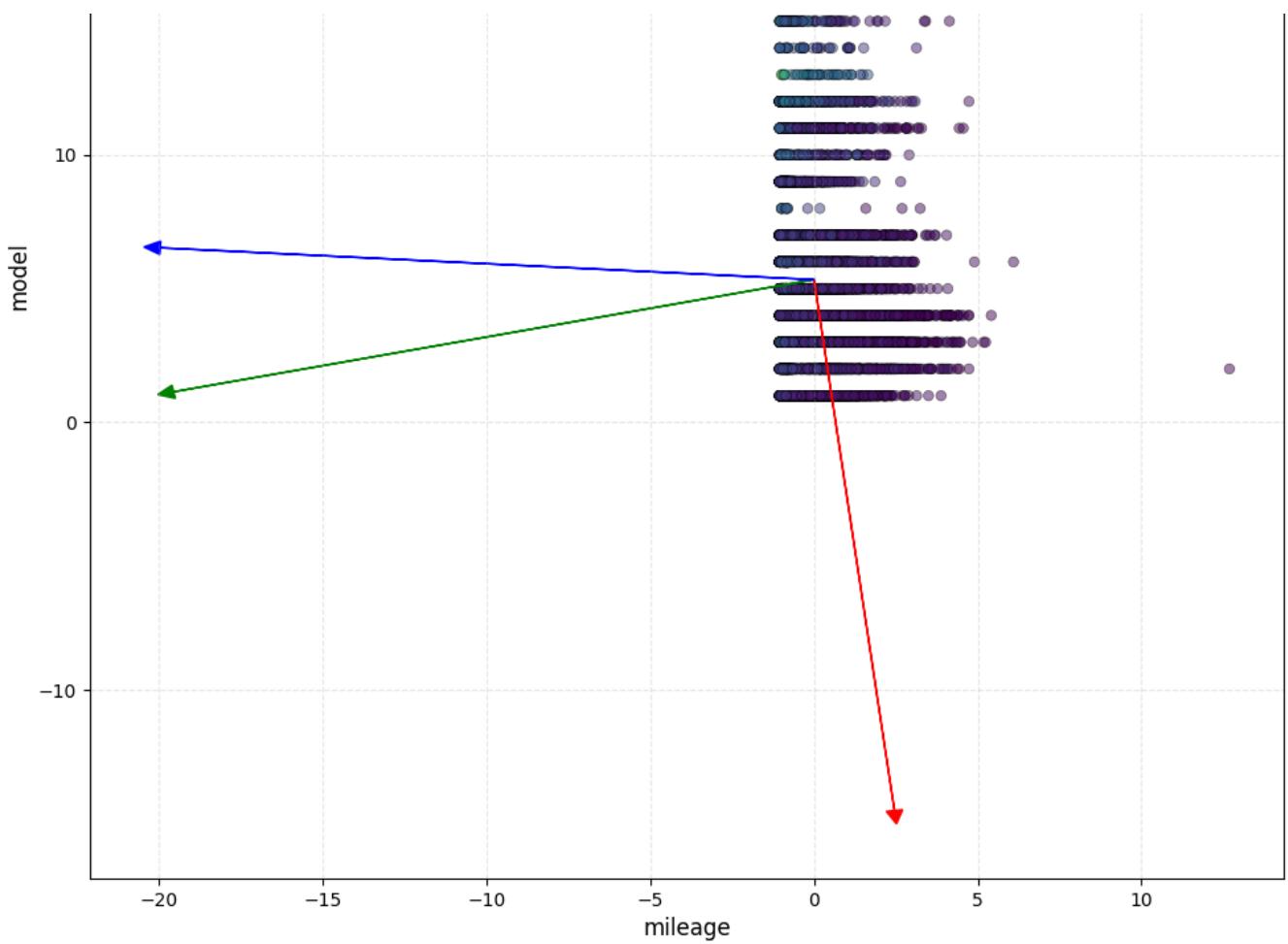


transmission vs engineSize

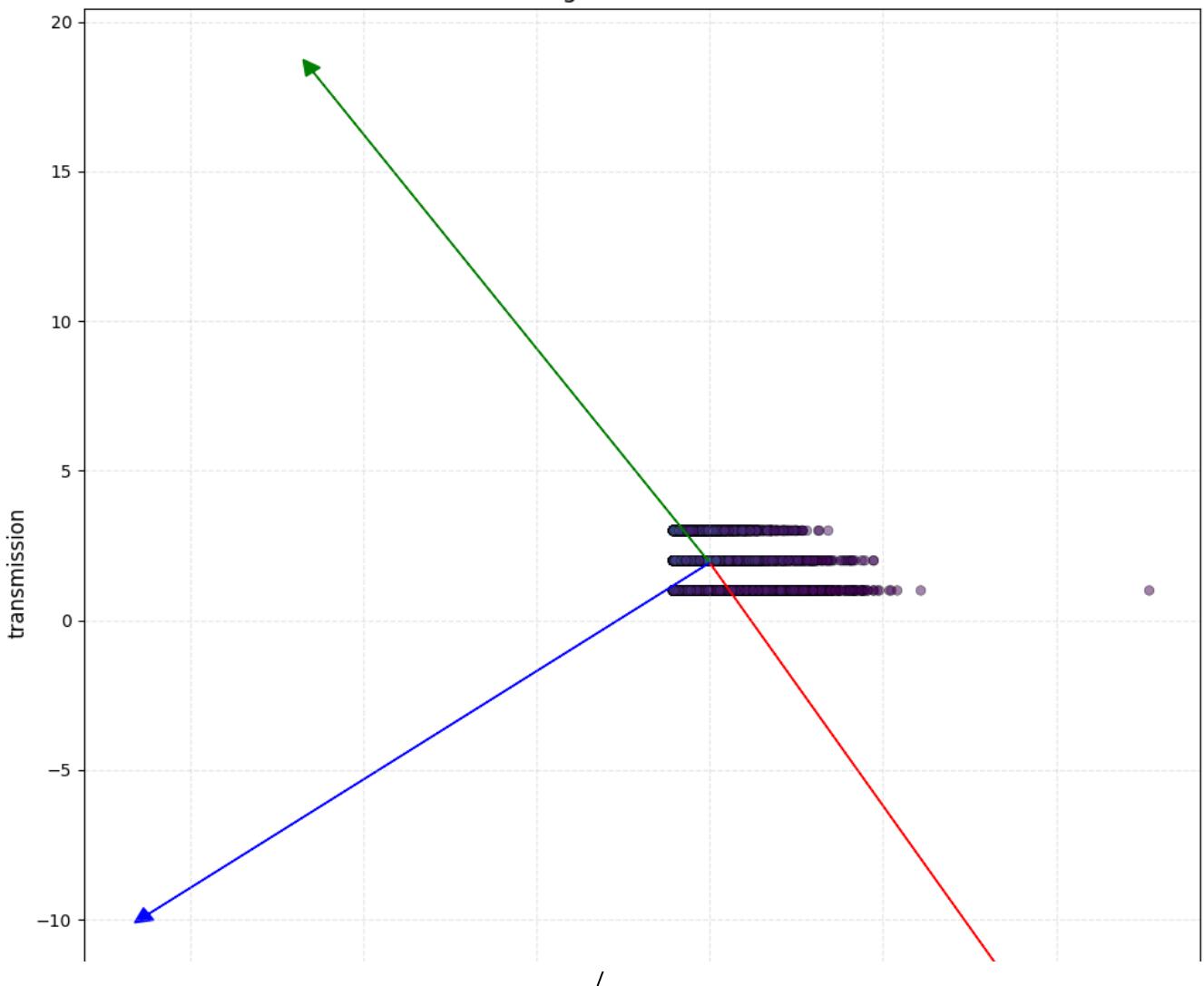


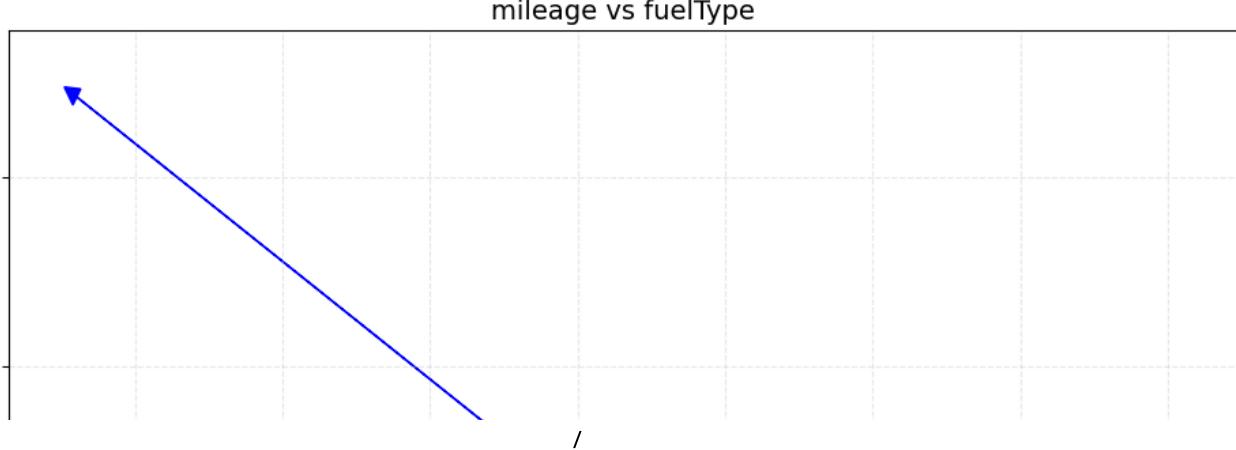
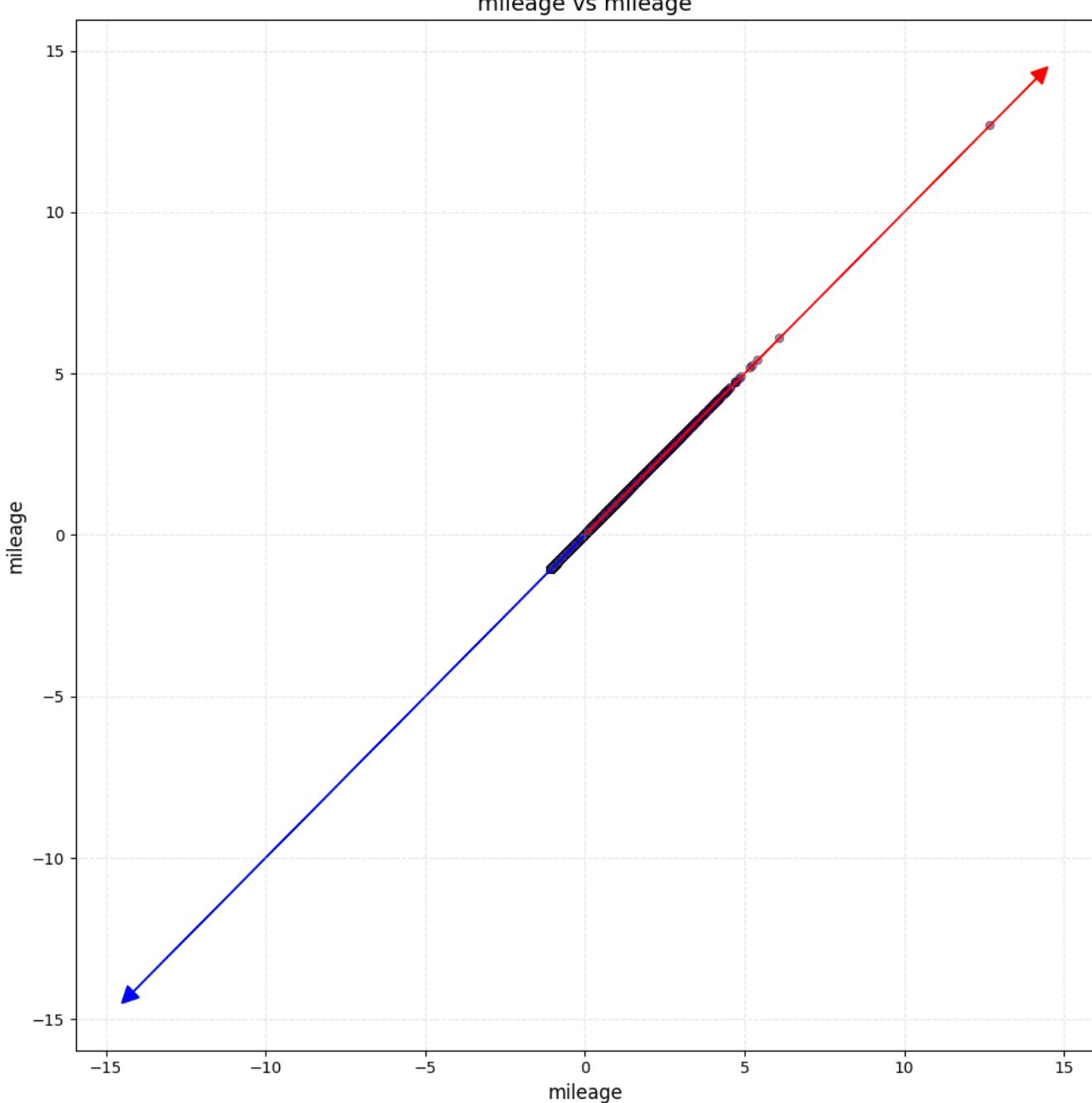
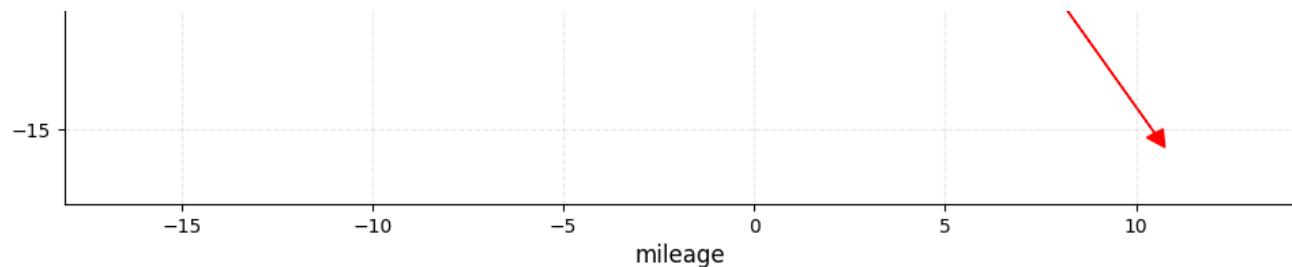
mileage vs model

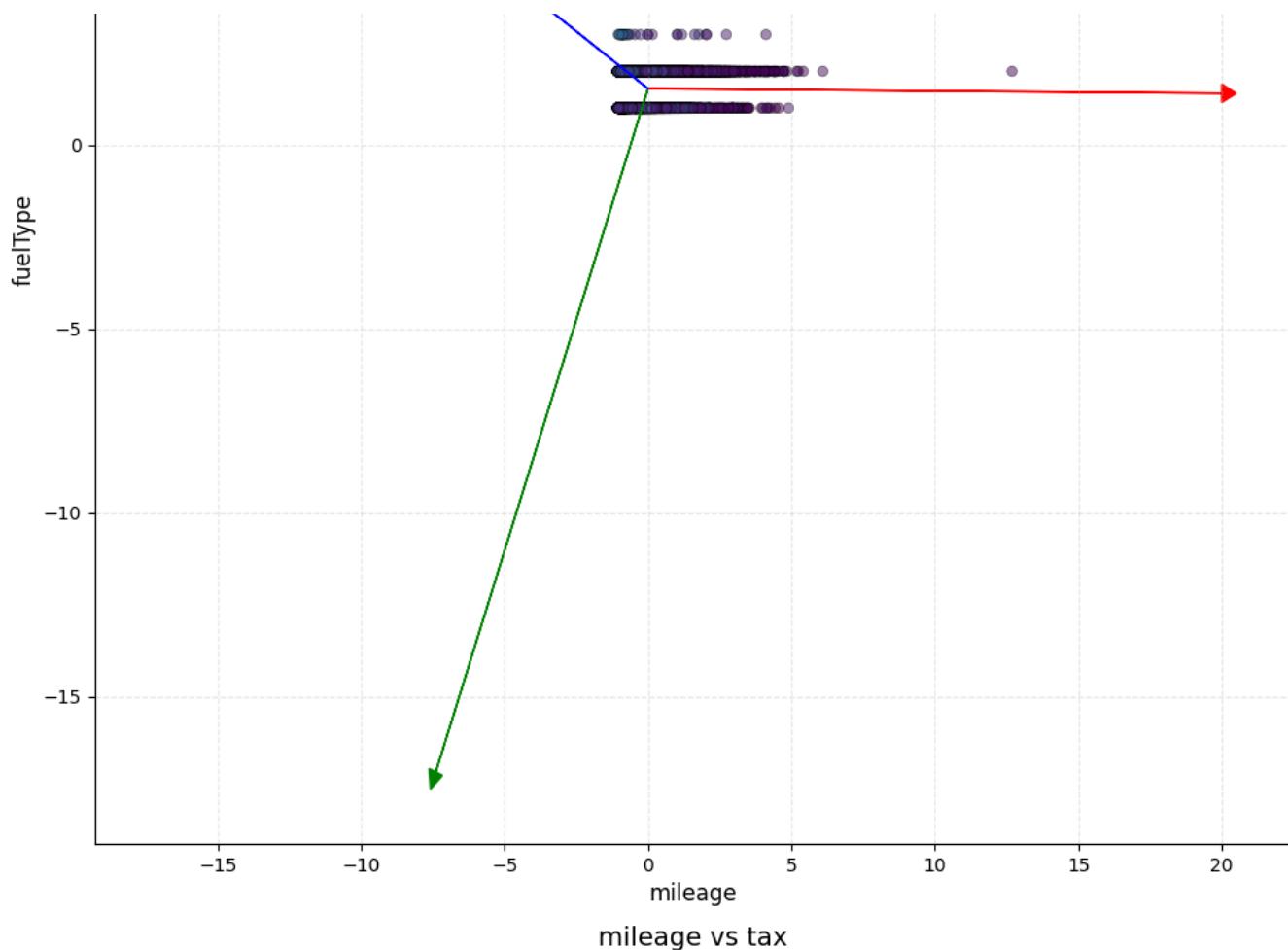




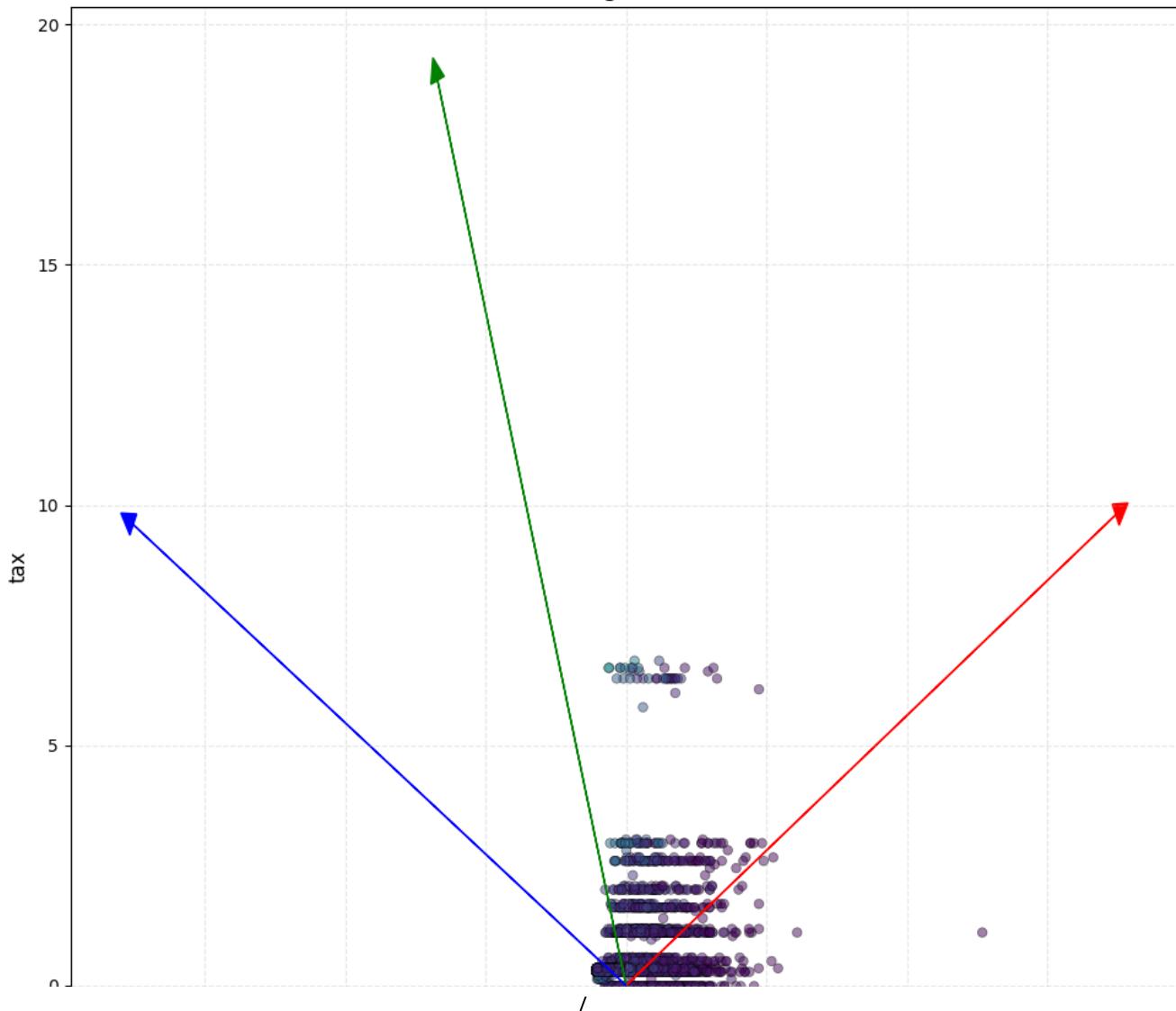
mileage vs transmission

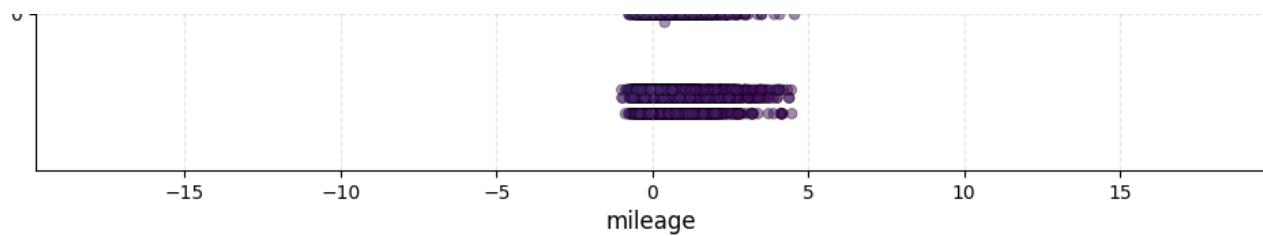




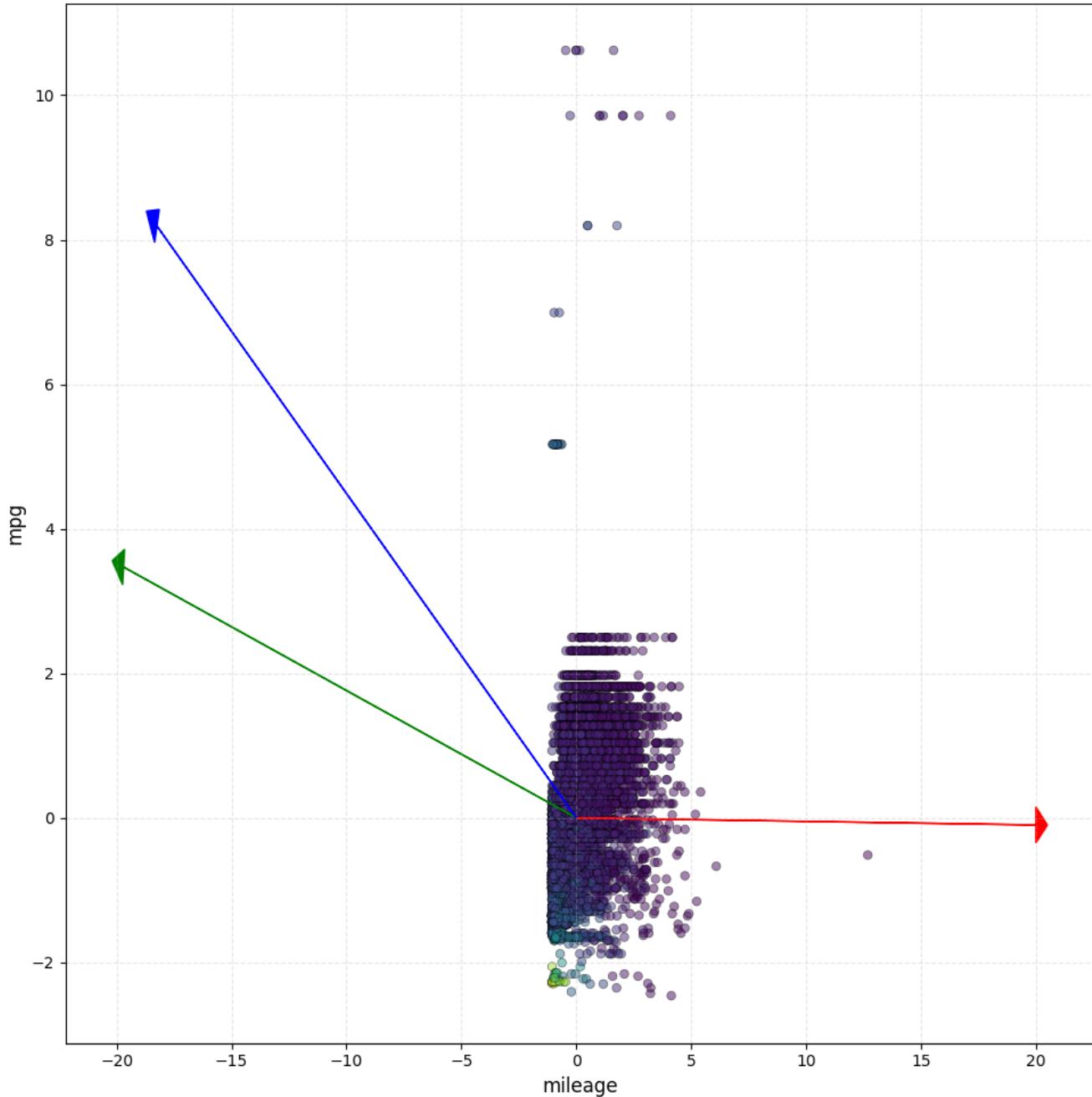


mileage vs tax

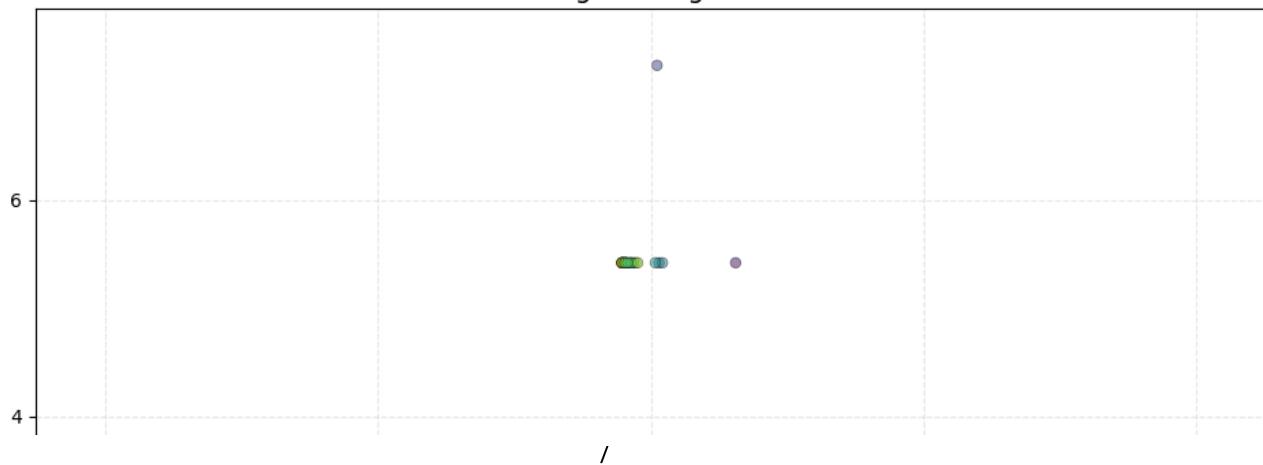


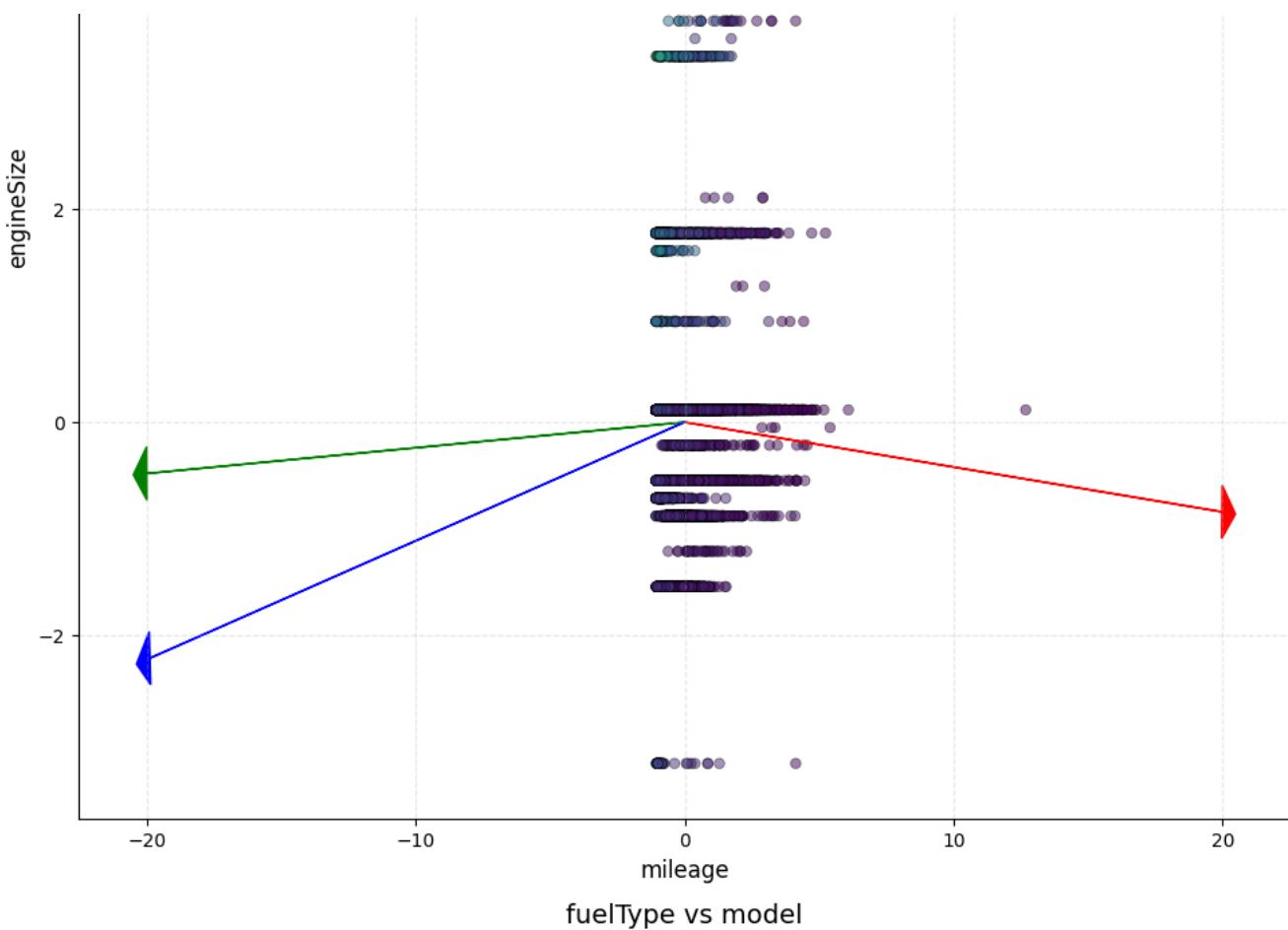


mileage vs mpg

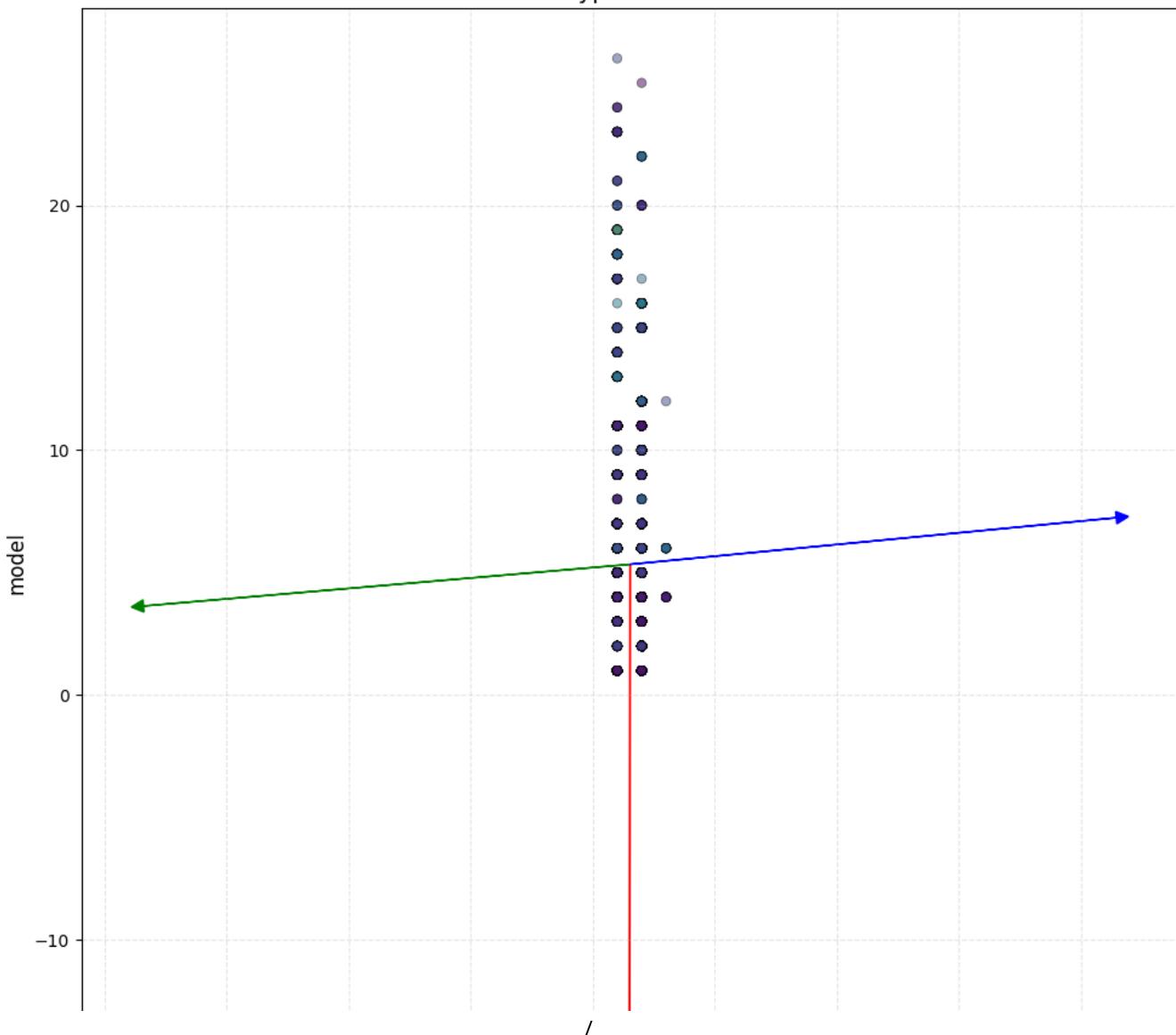


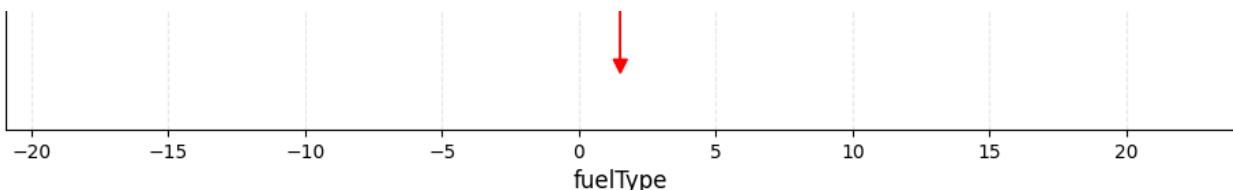
mileage vs engineSize



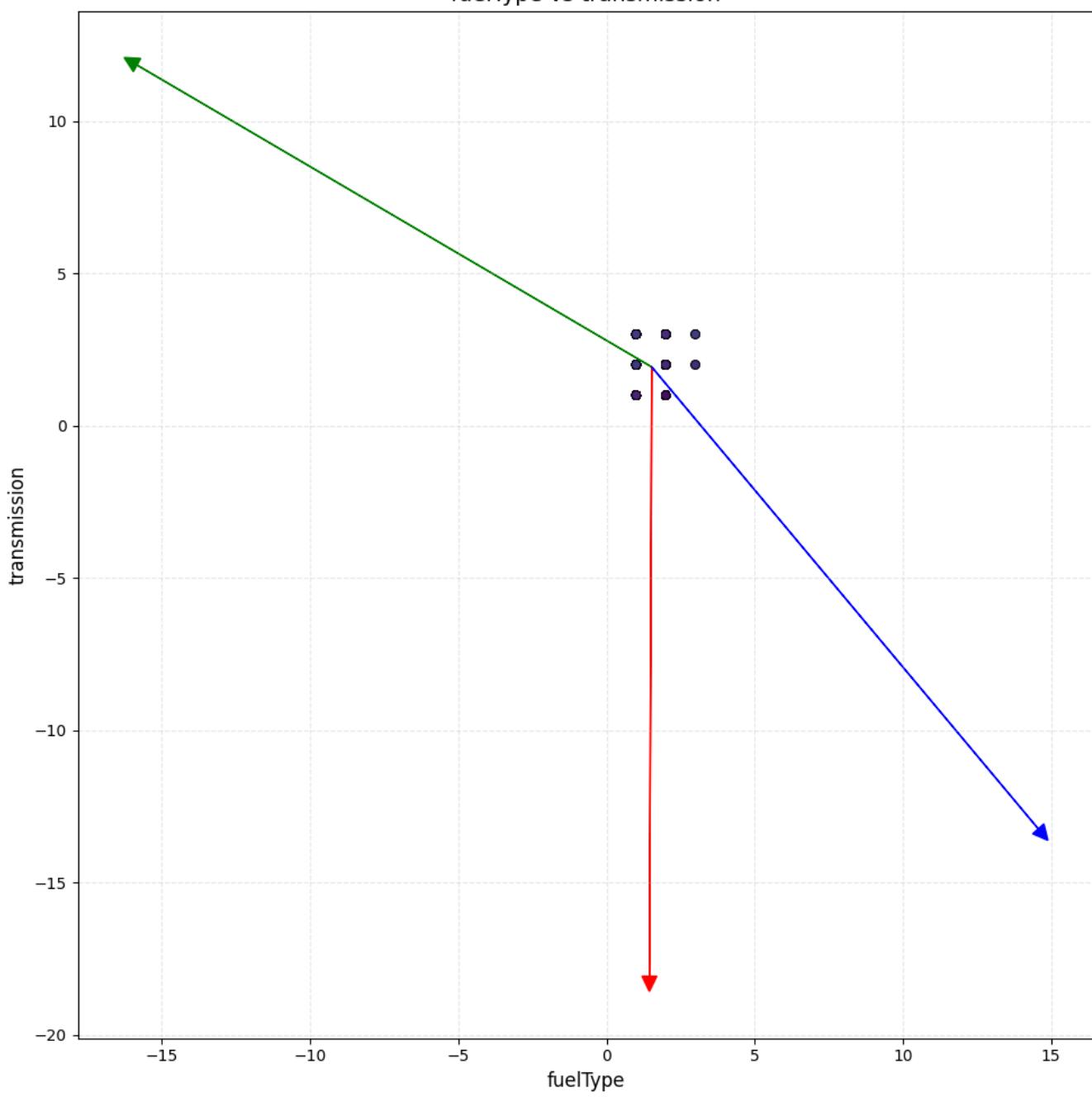


fuelType vs model

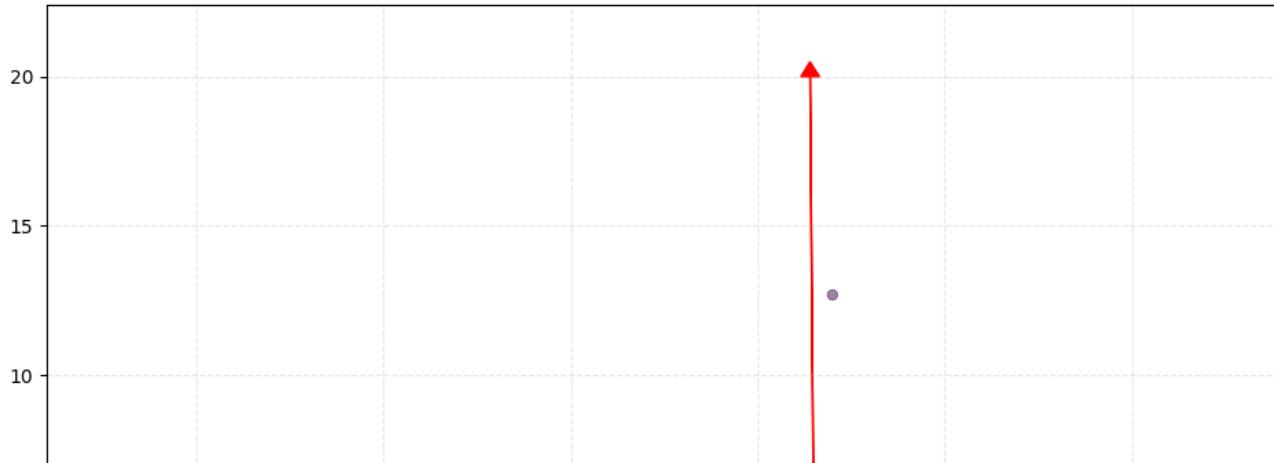


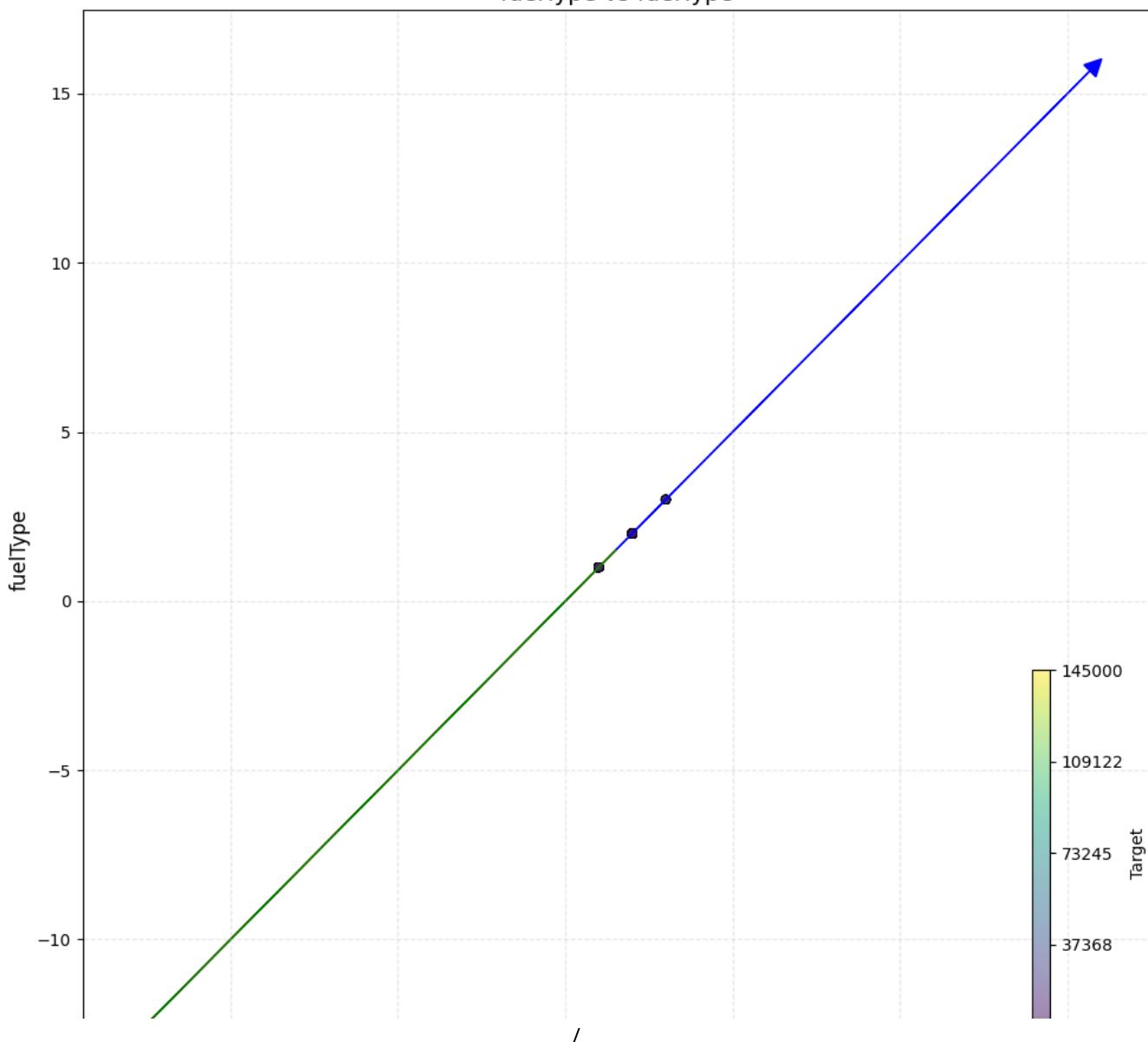
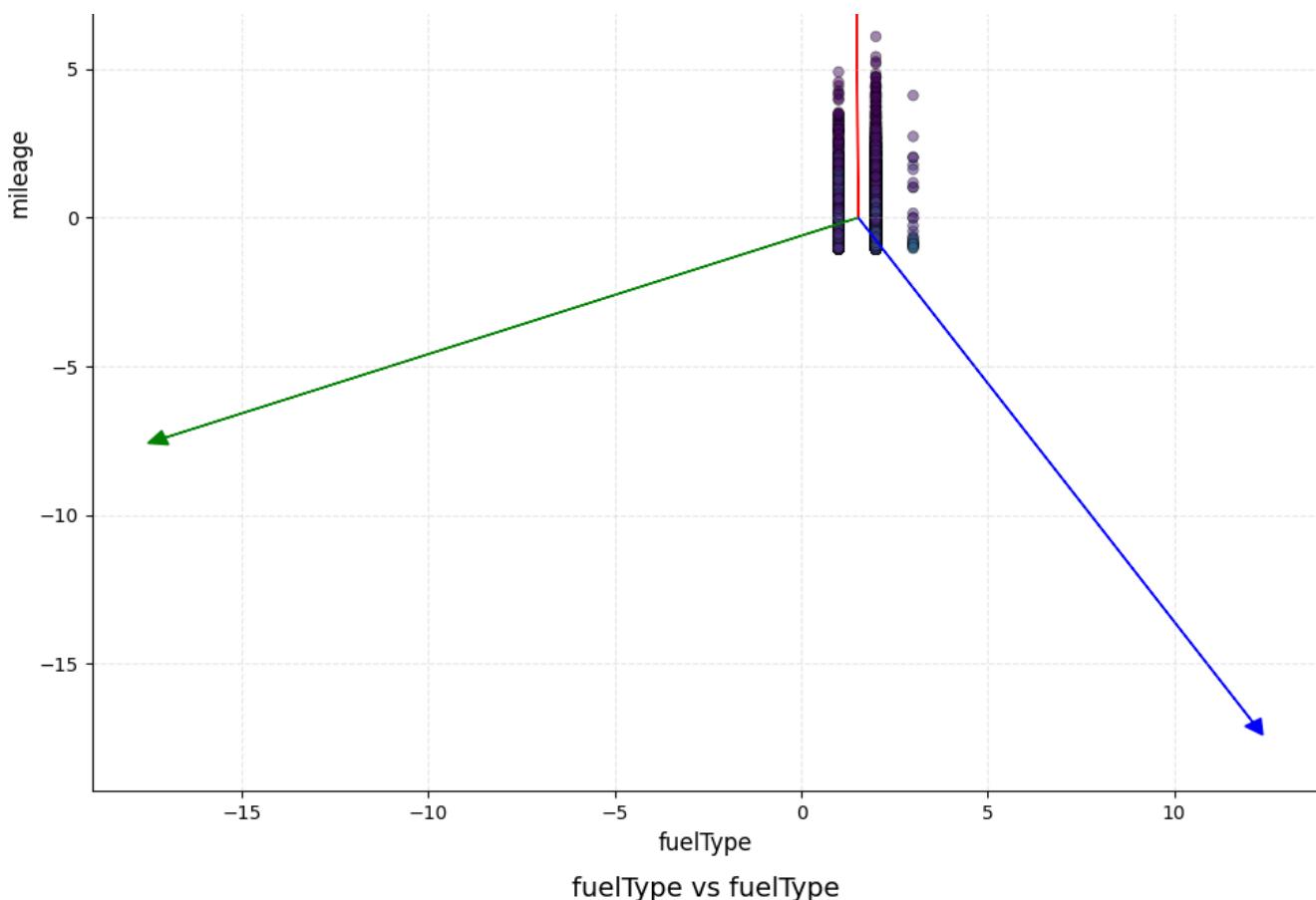


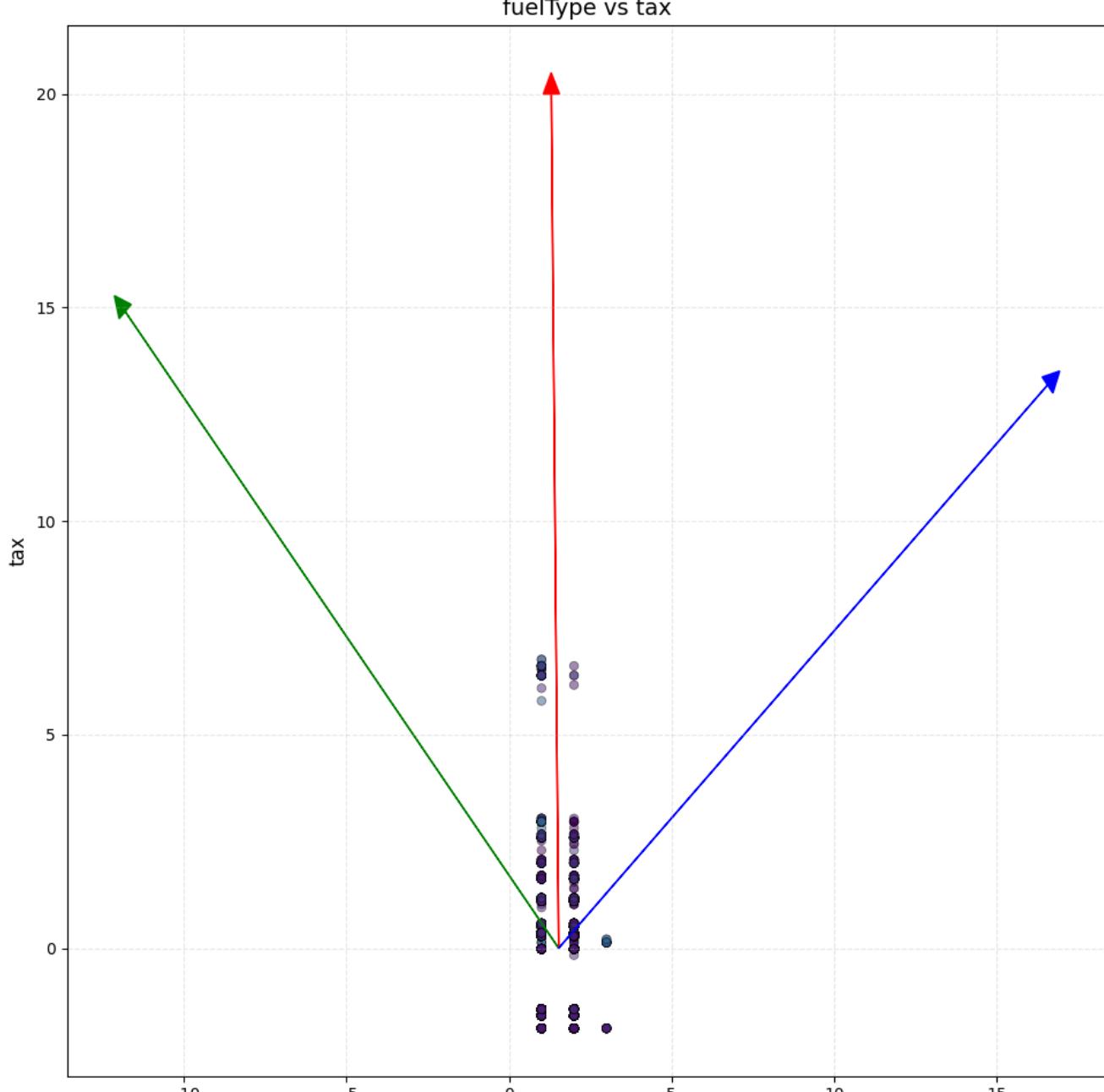
fuelType vs transmission



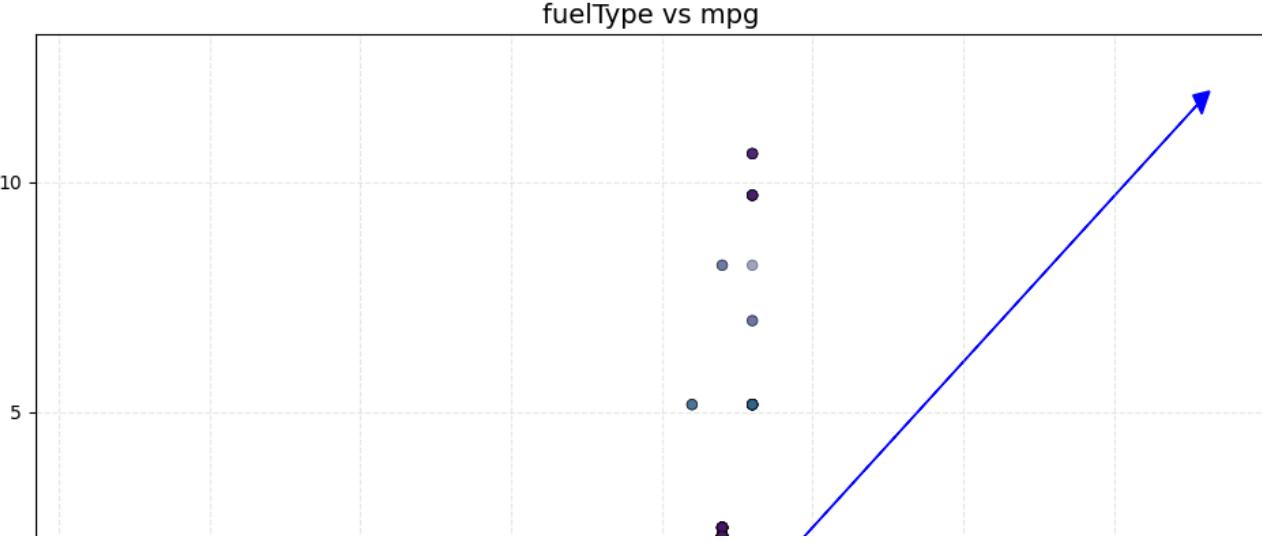
fuelType vs mileage

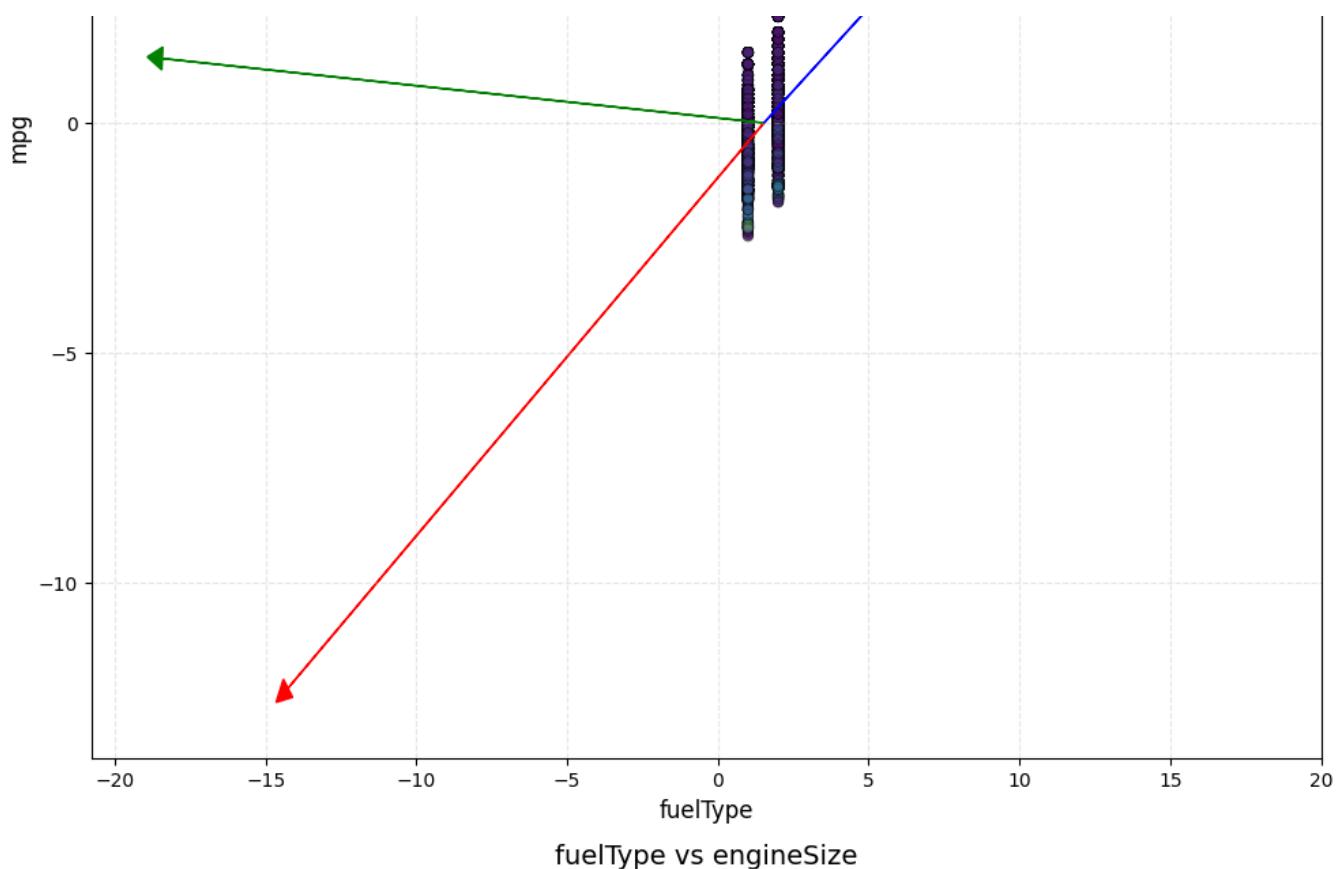




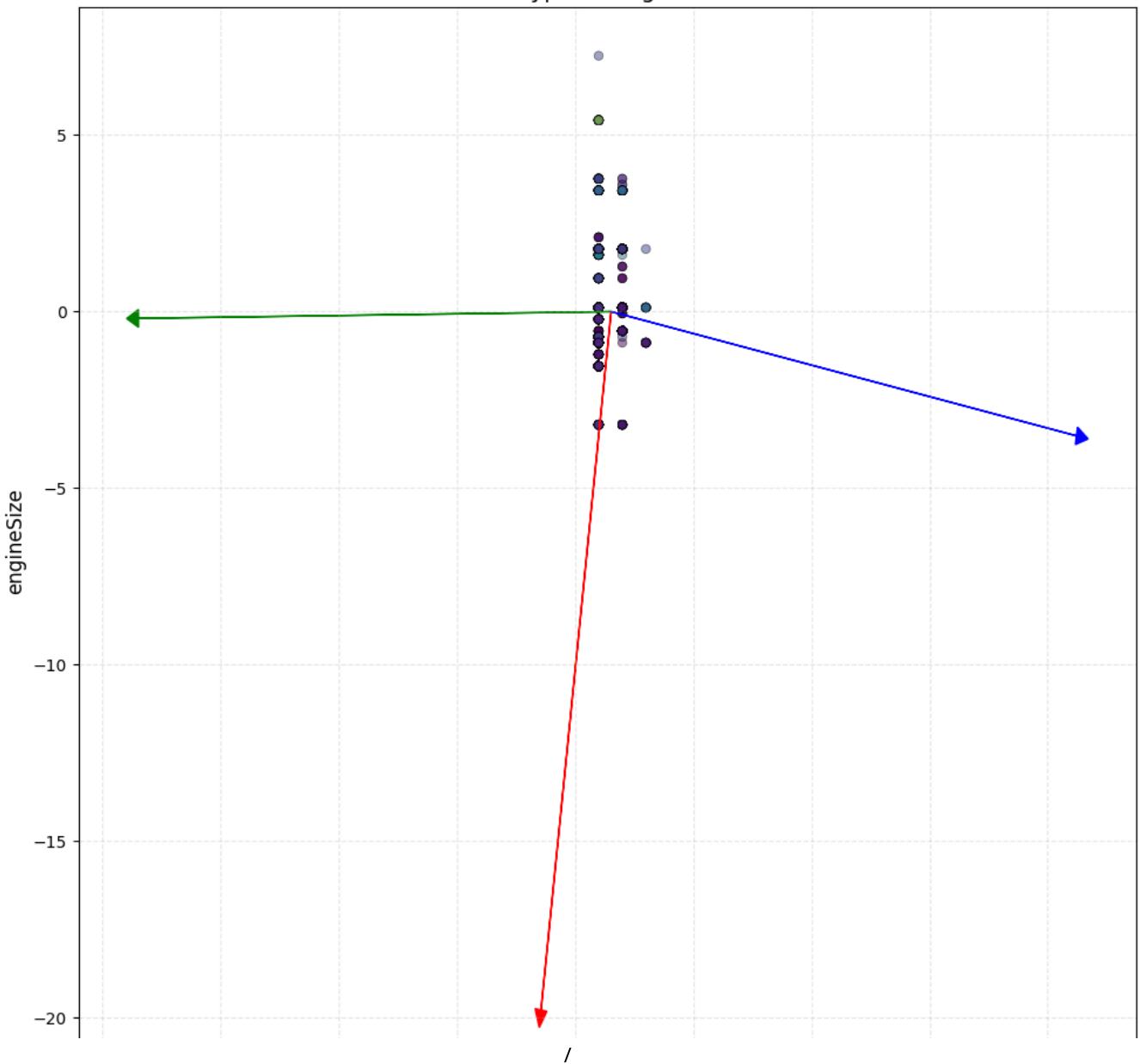


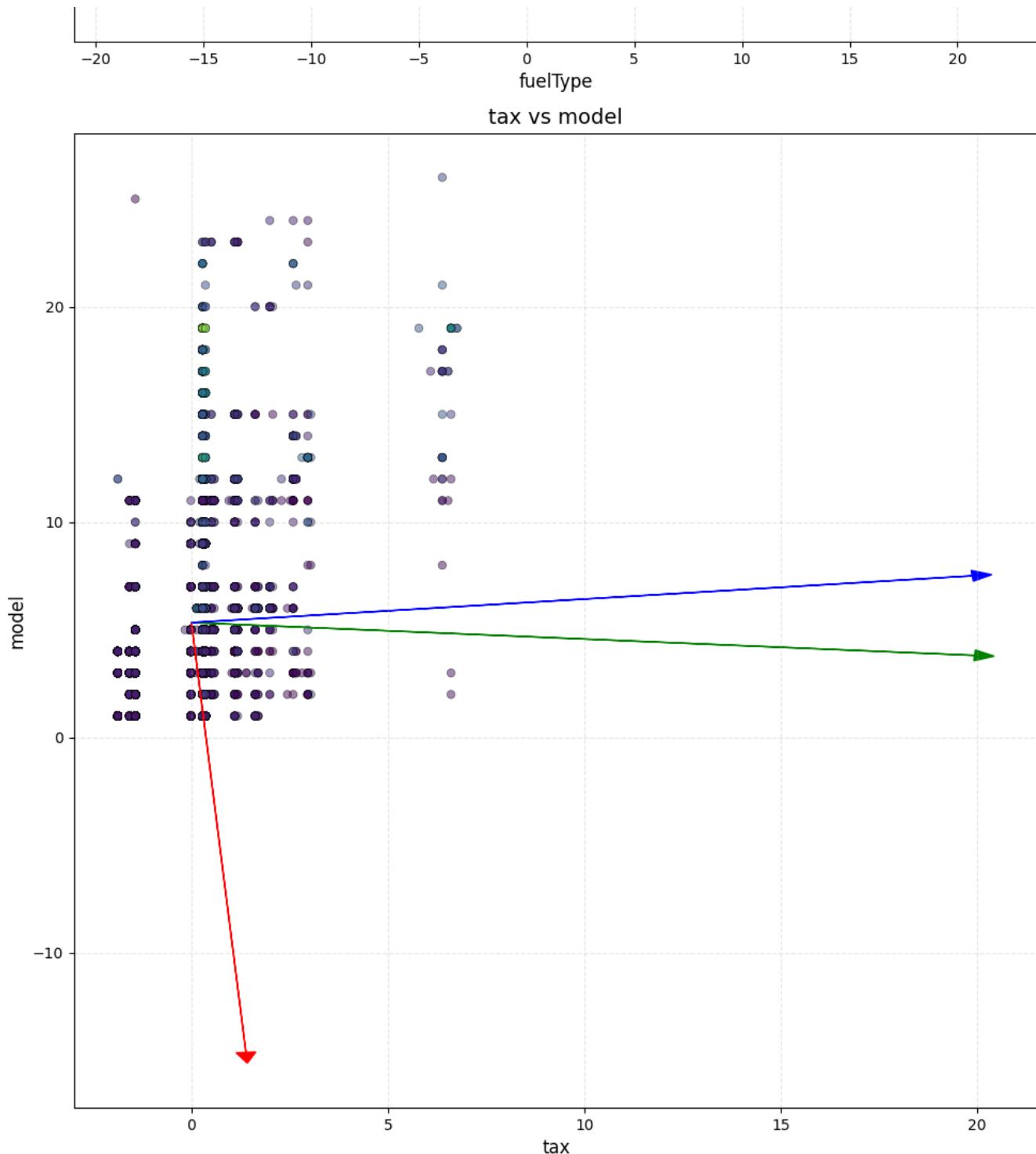
fuelType vs mpg



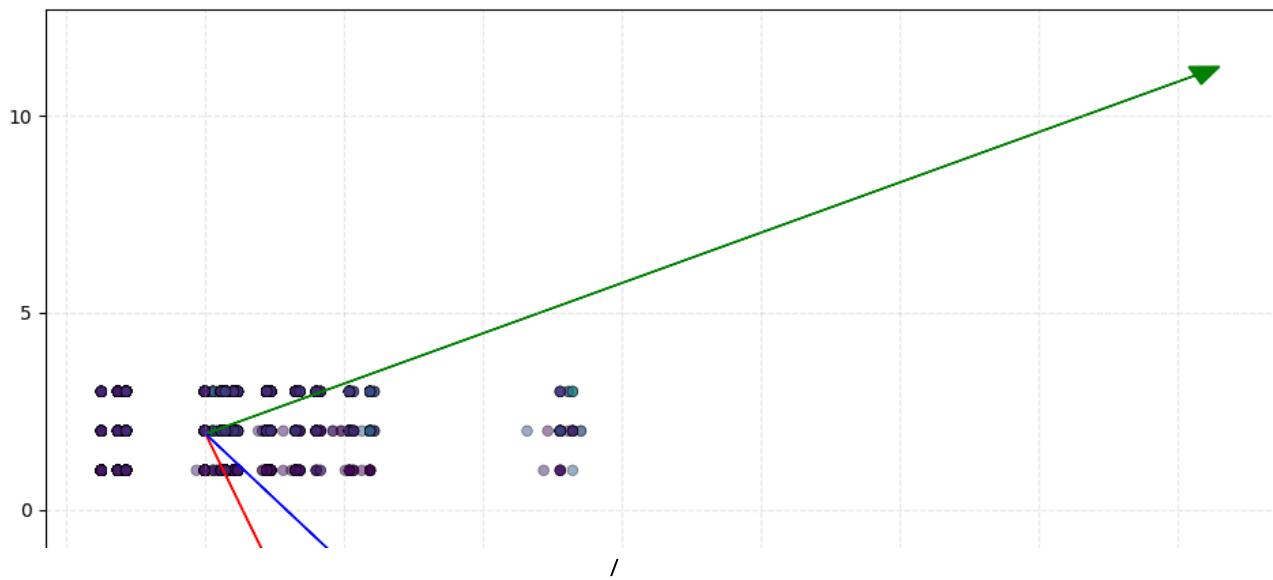


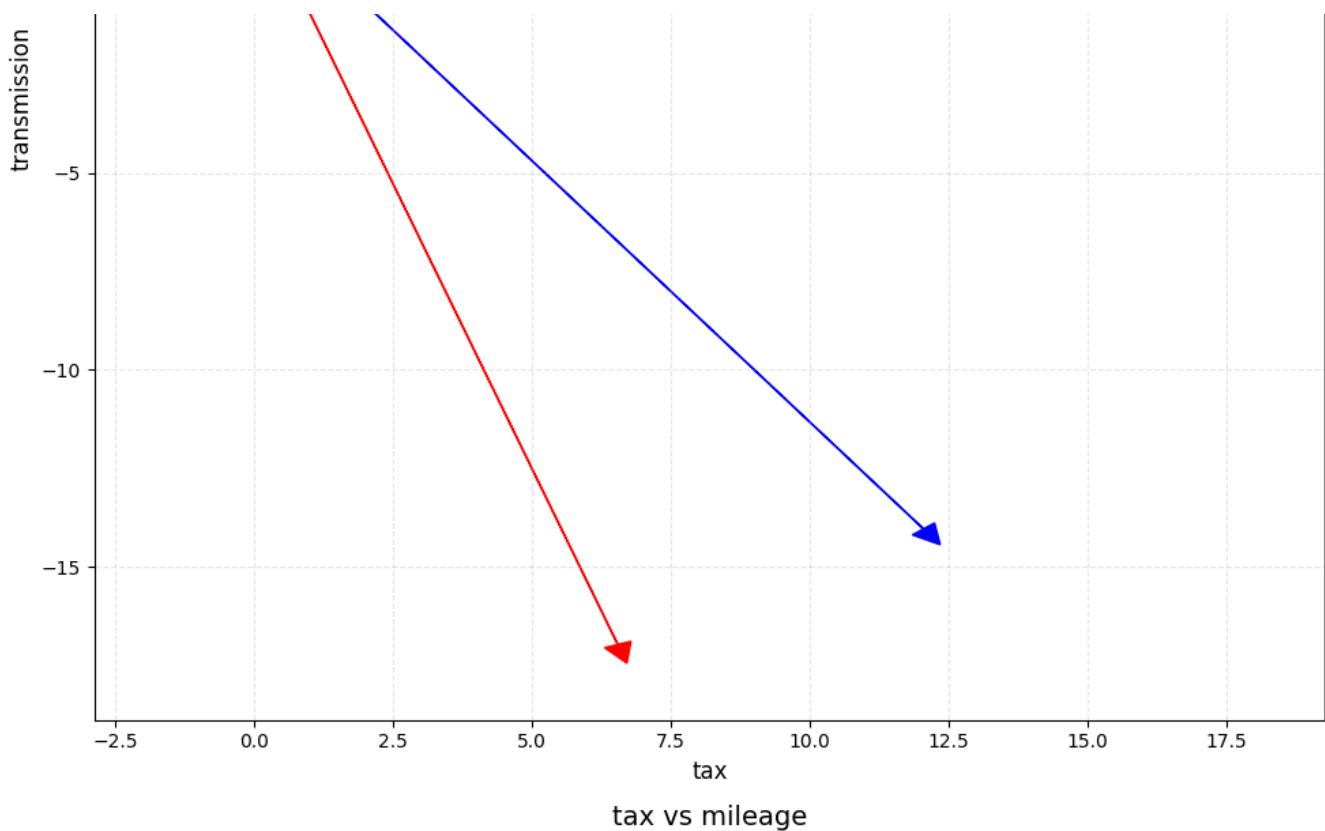
fuelType vs engineSize



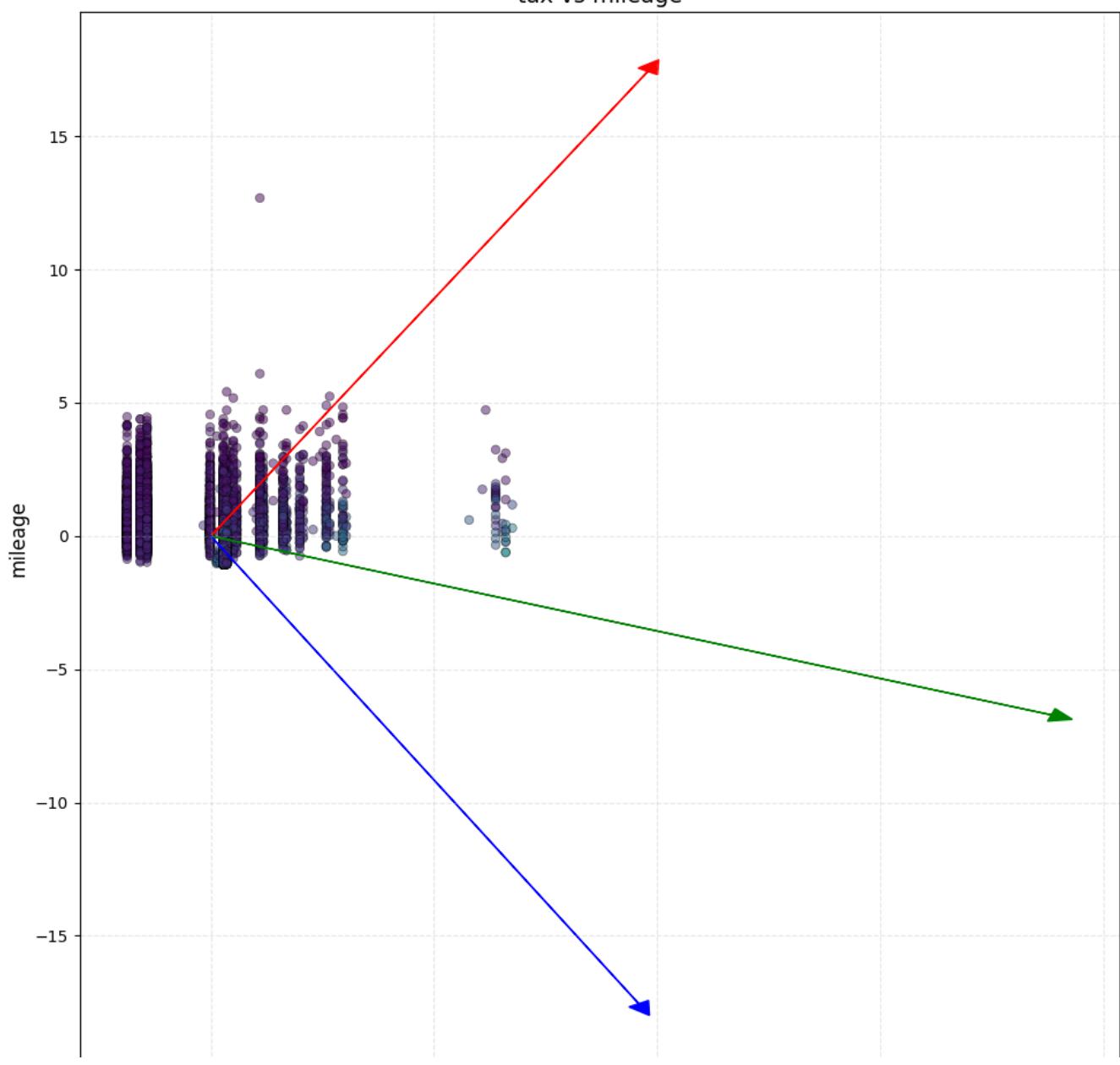


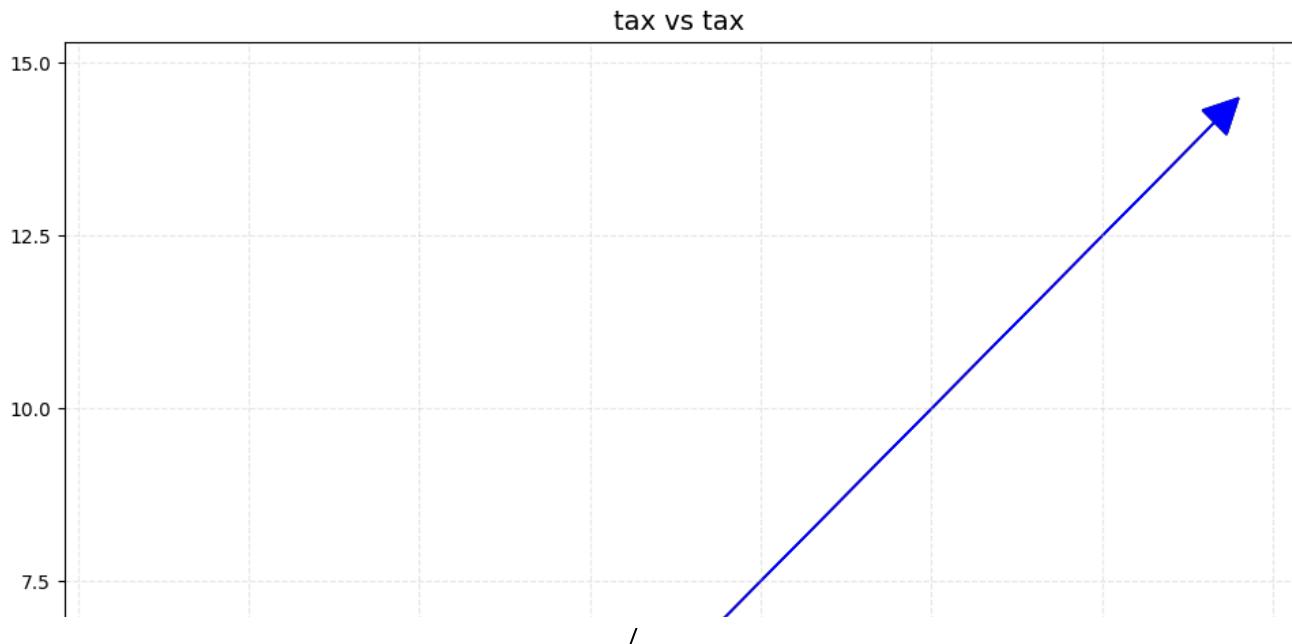
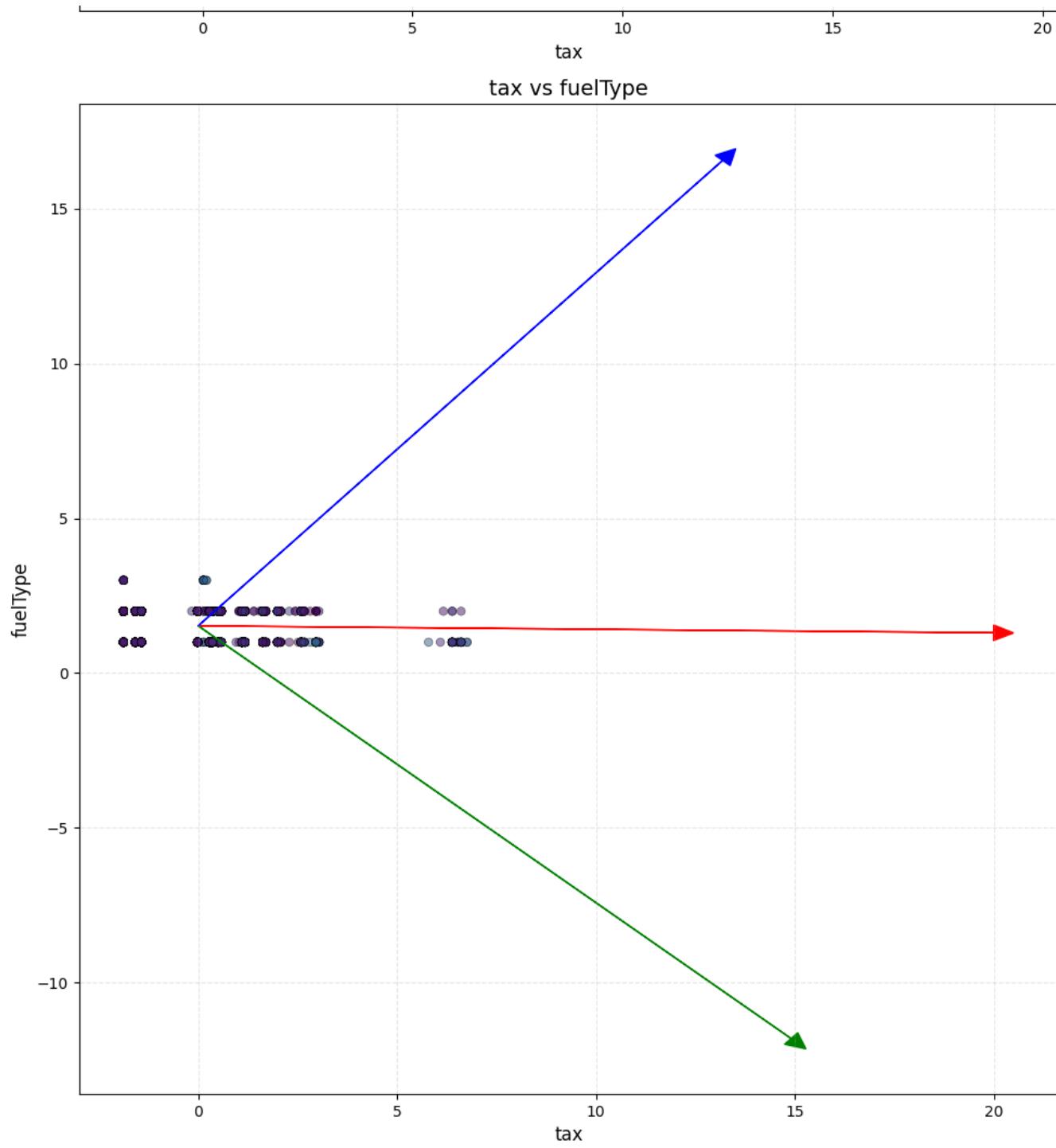
tax vs transmission

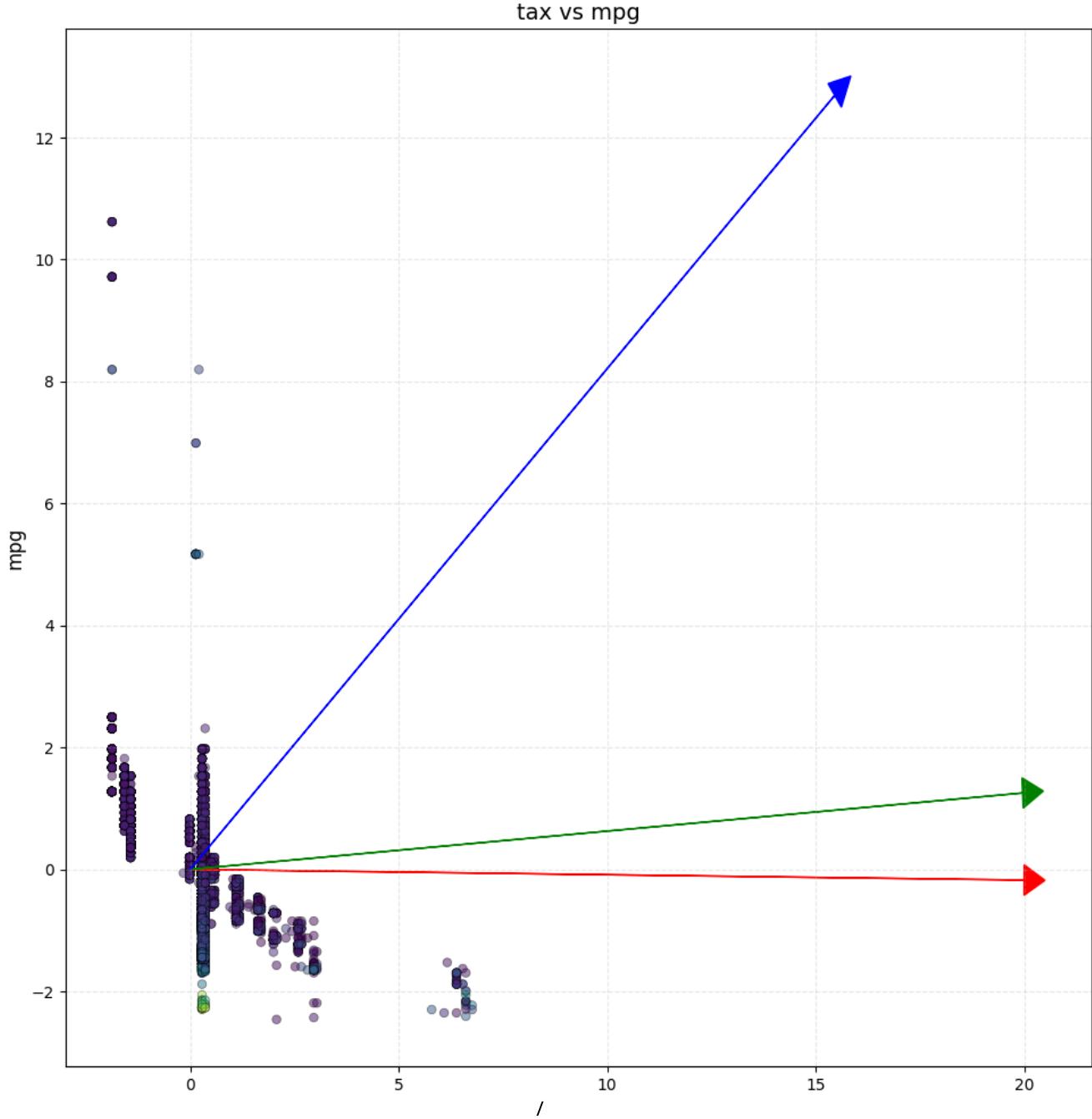
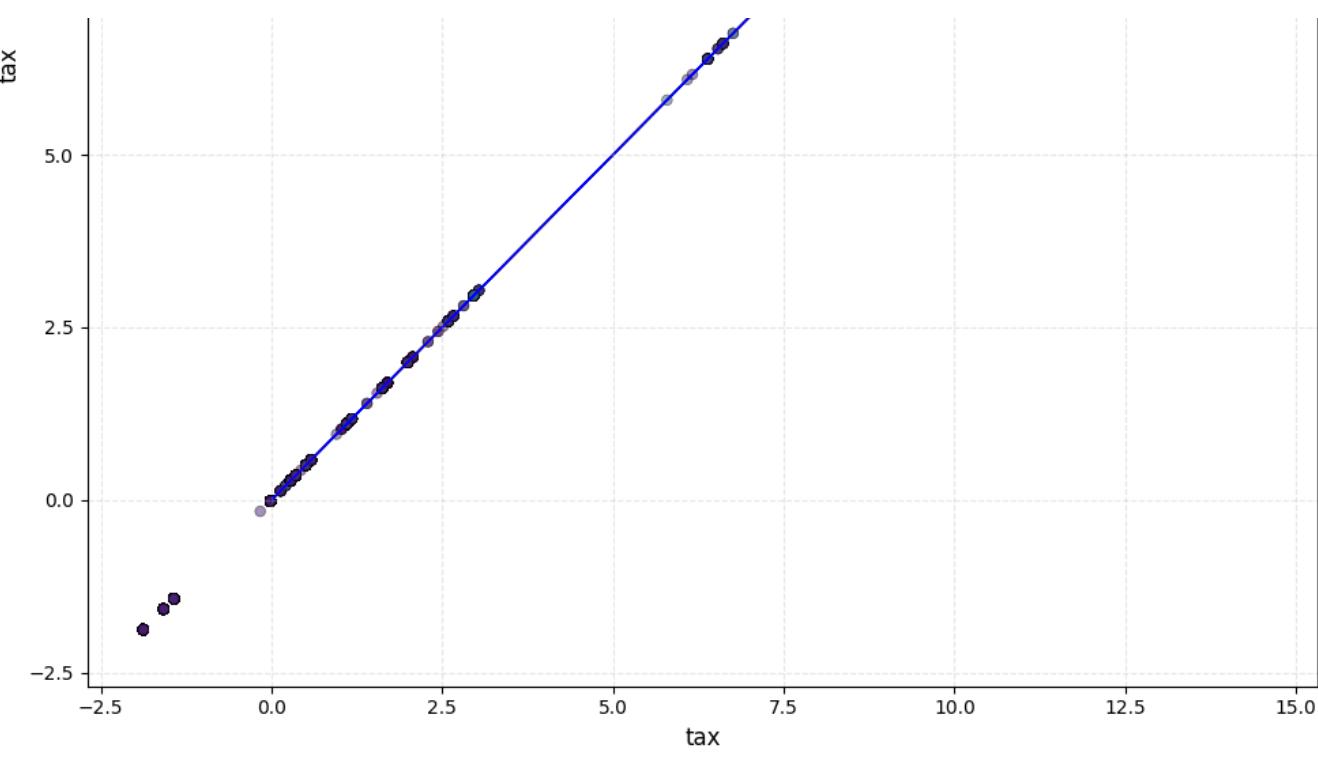


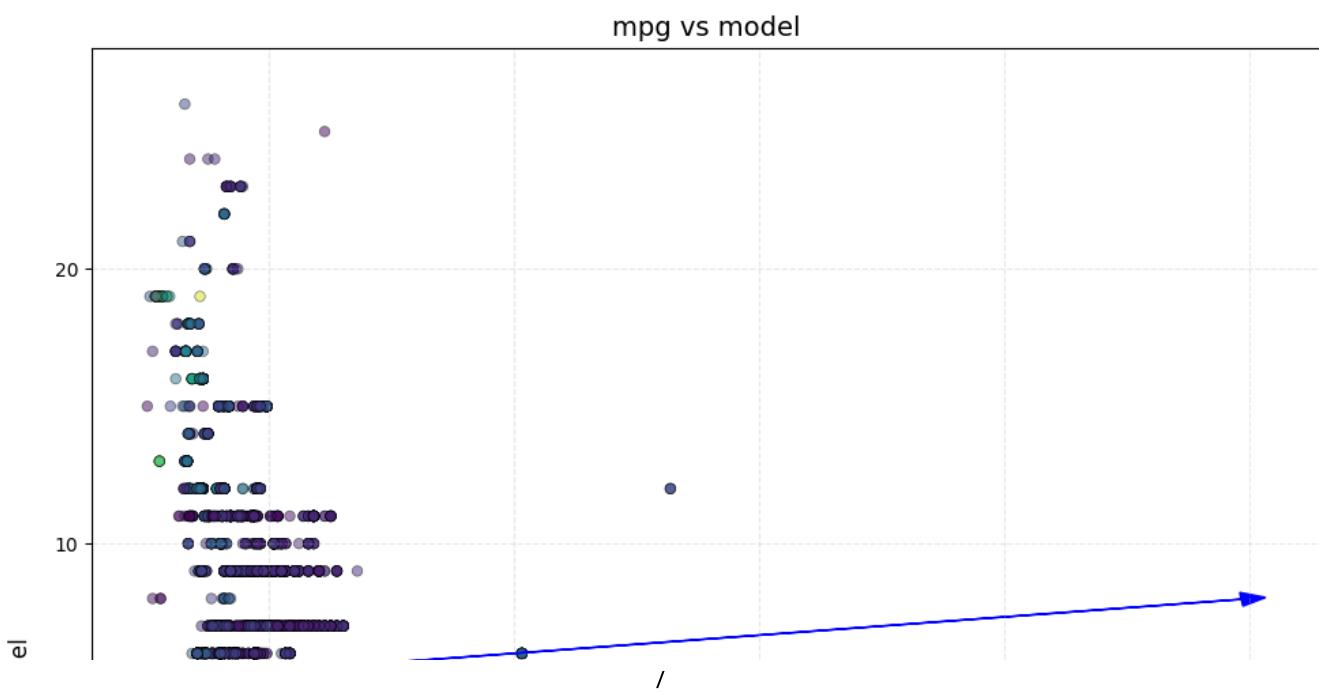
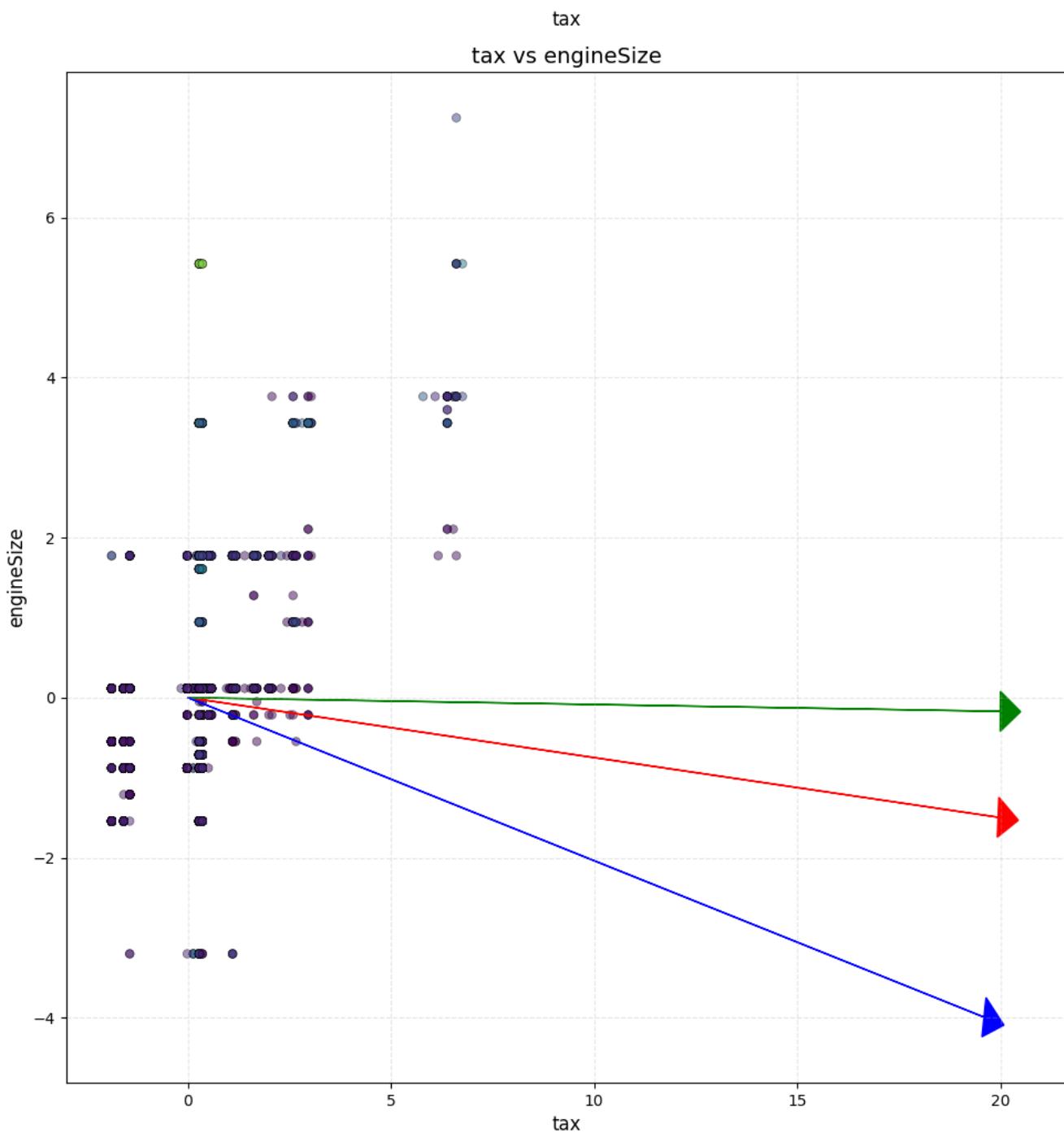


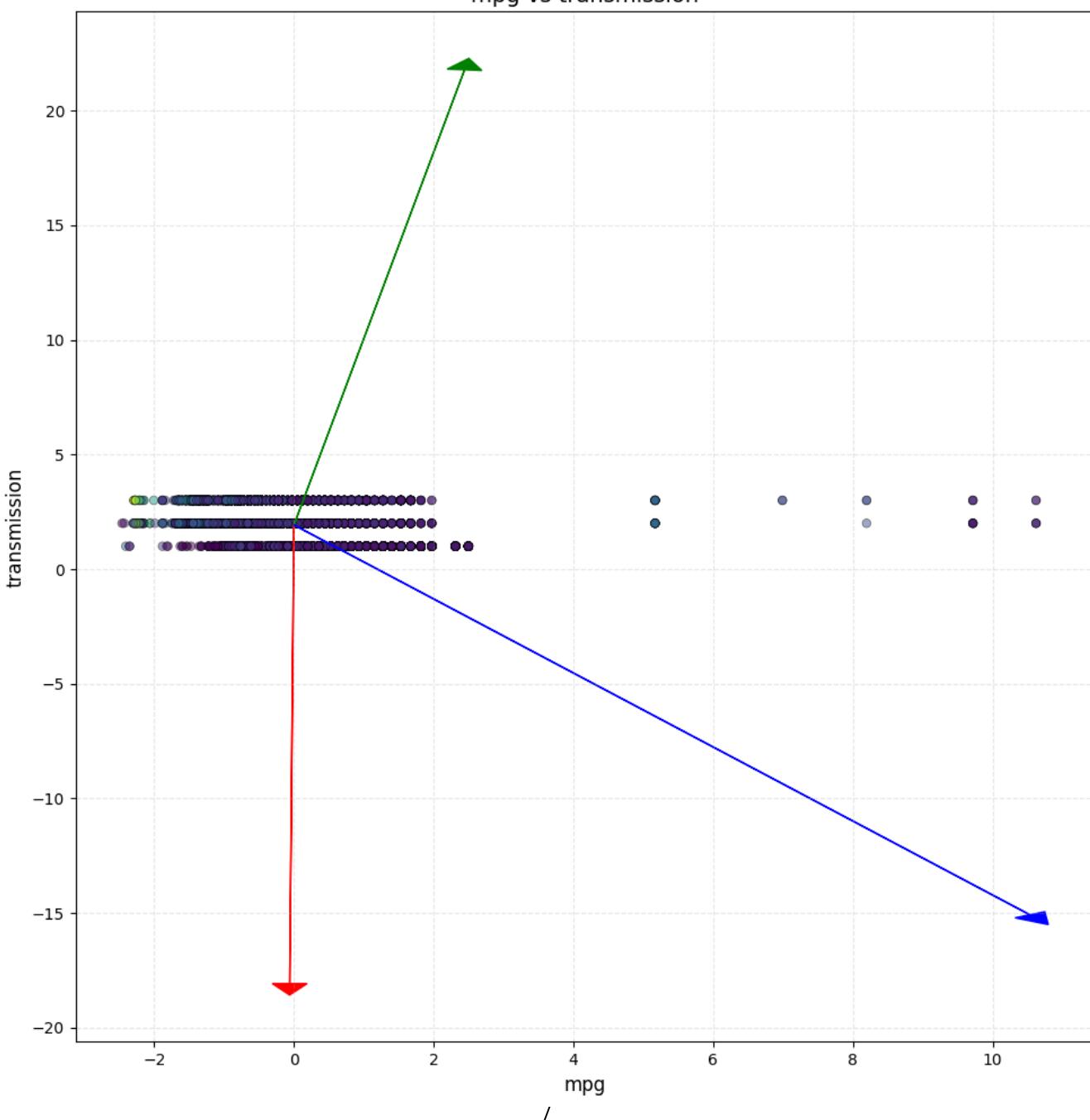
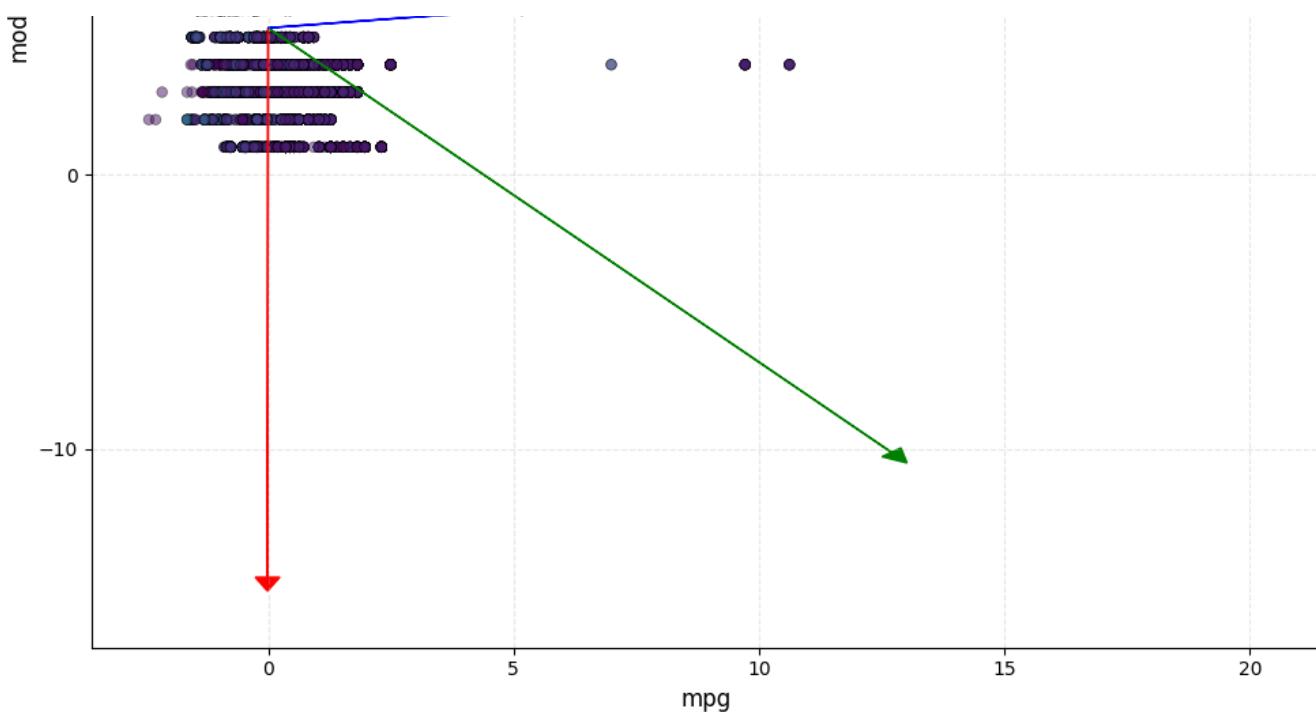
tax vs mileage

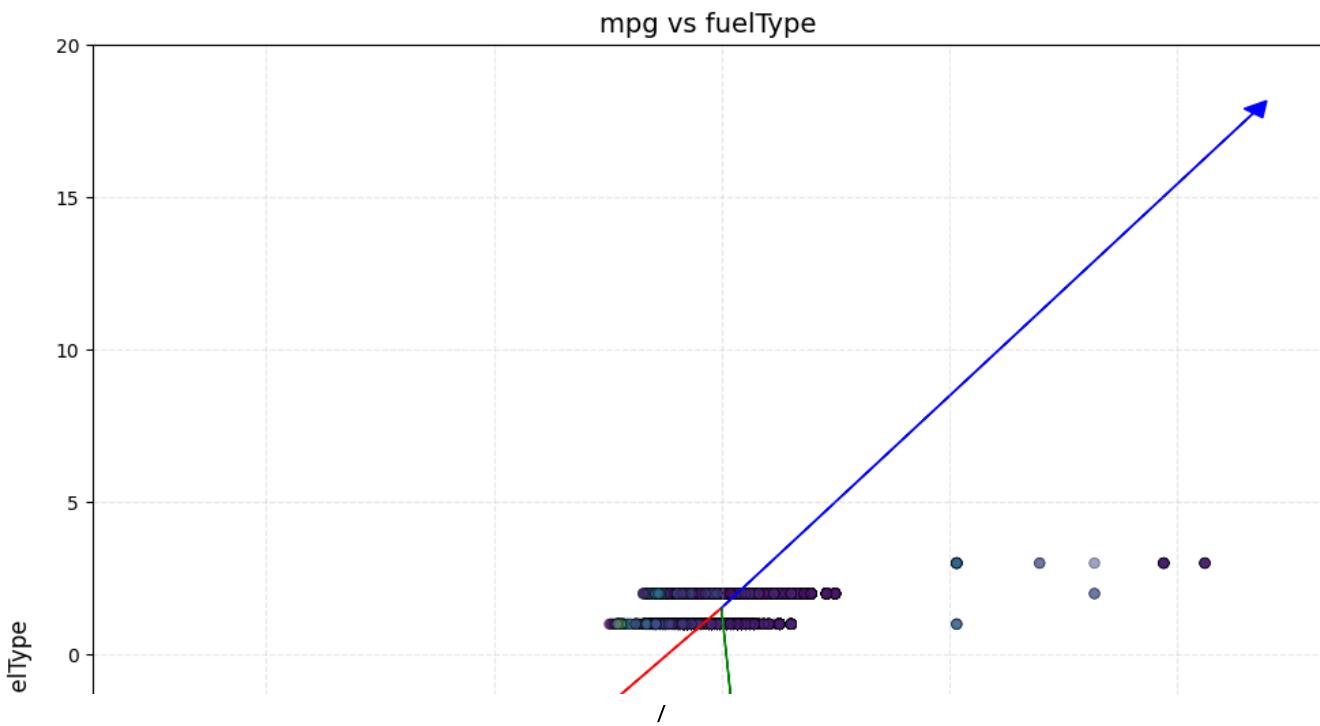
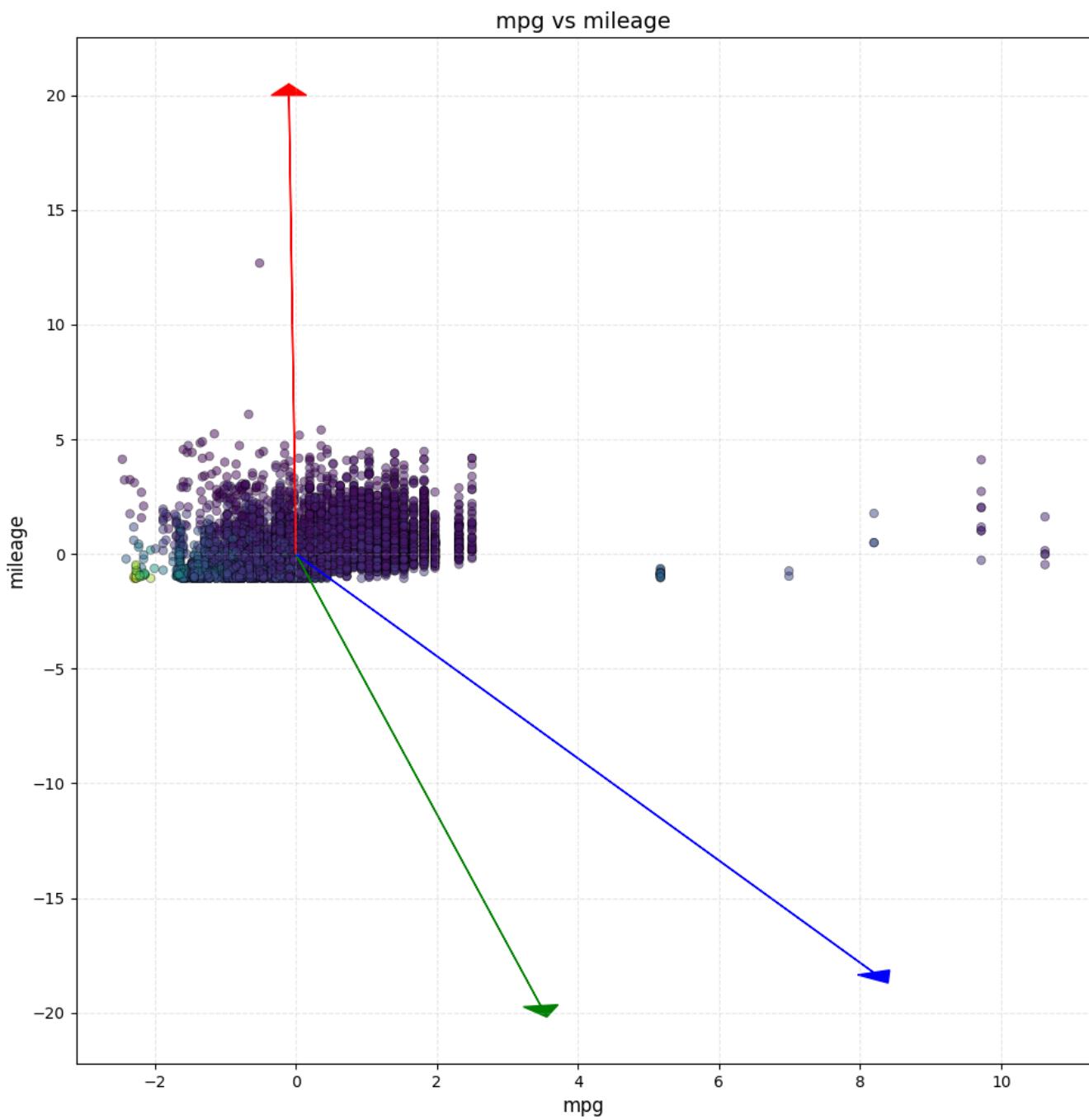


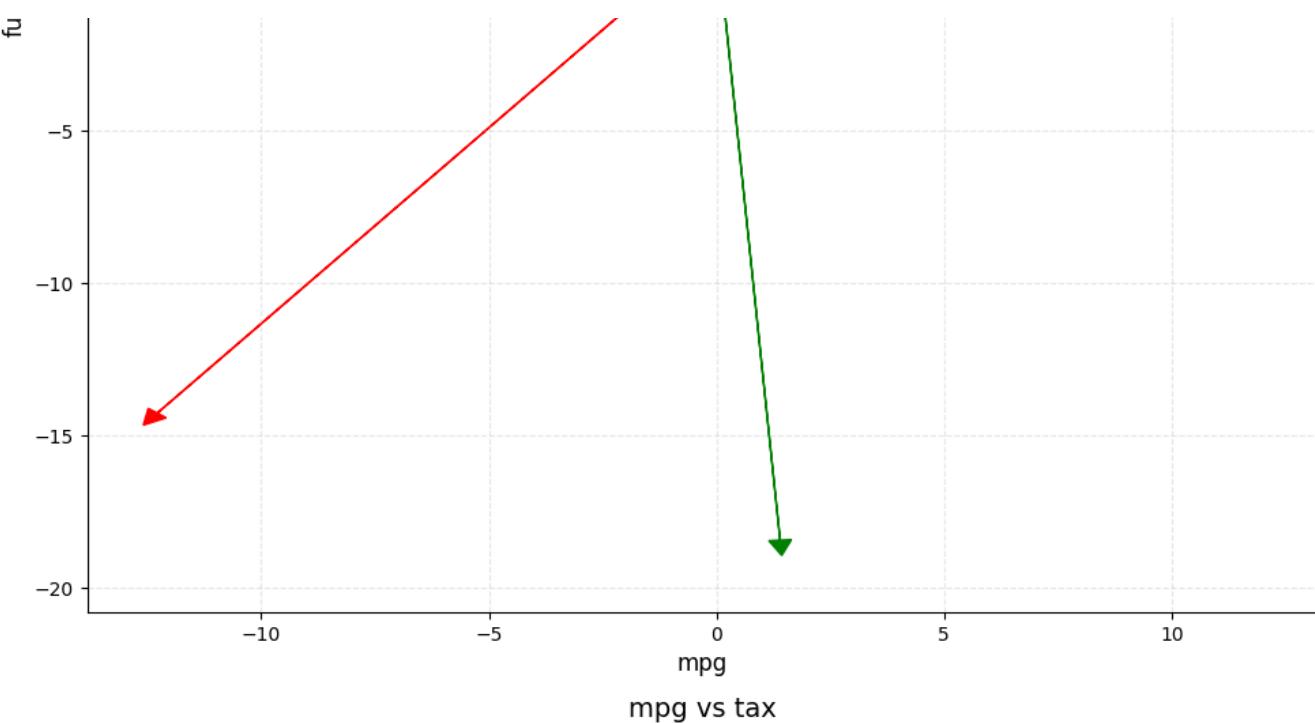




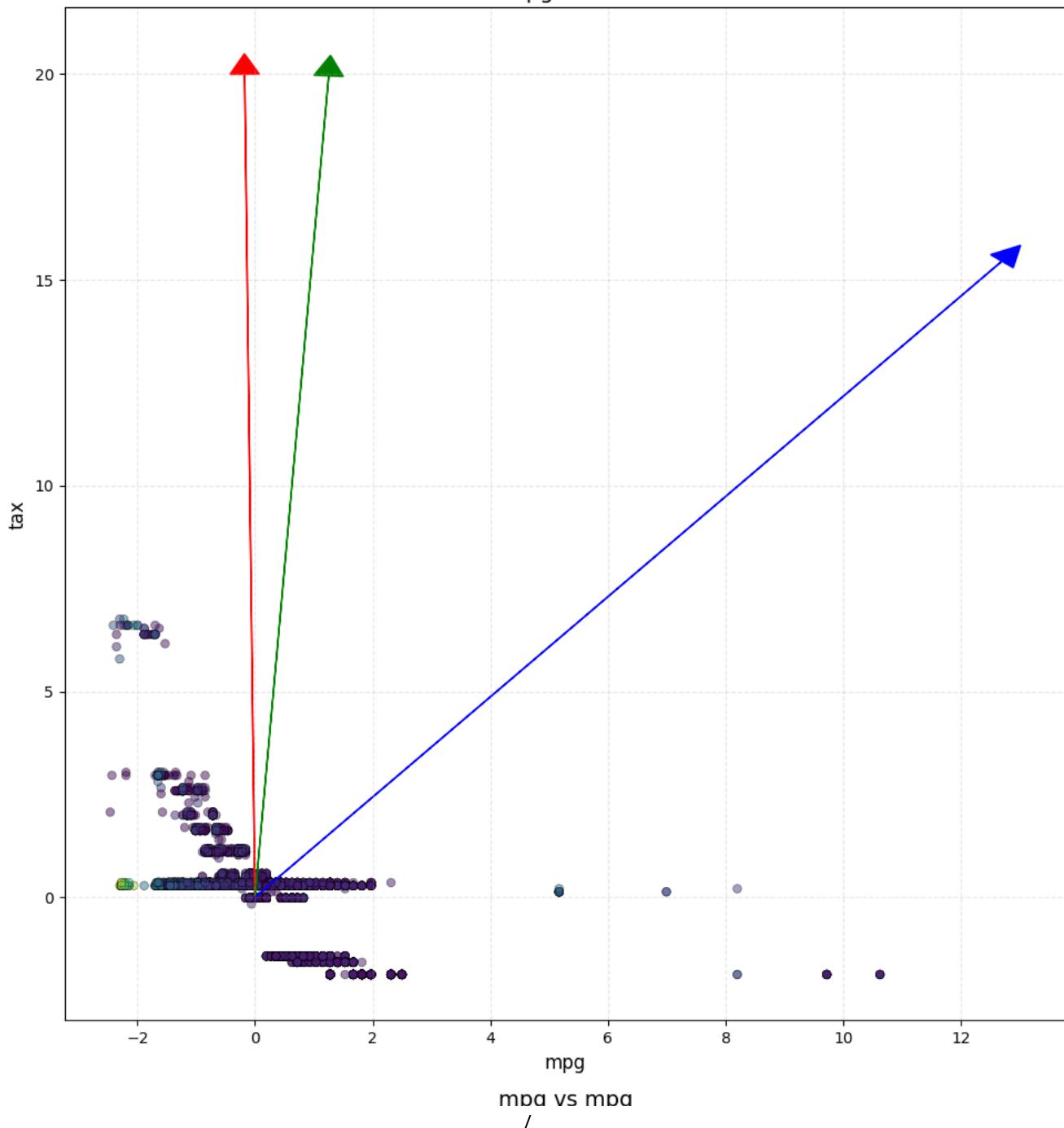




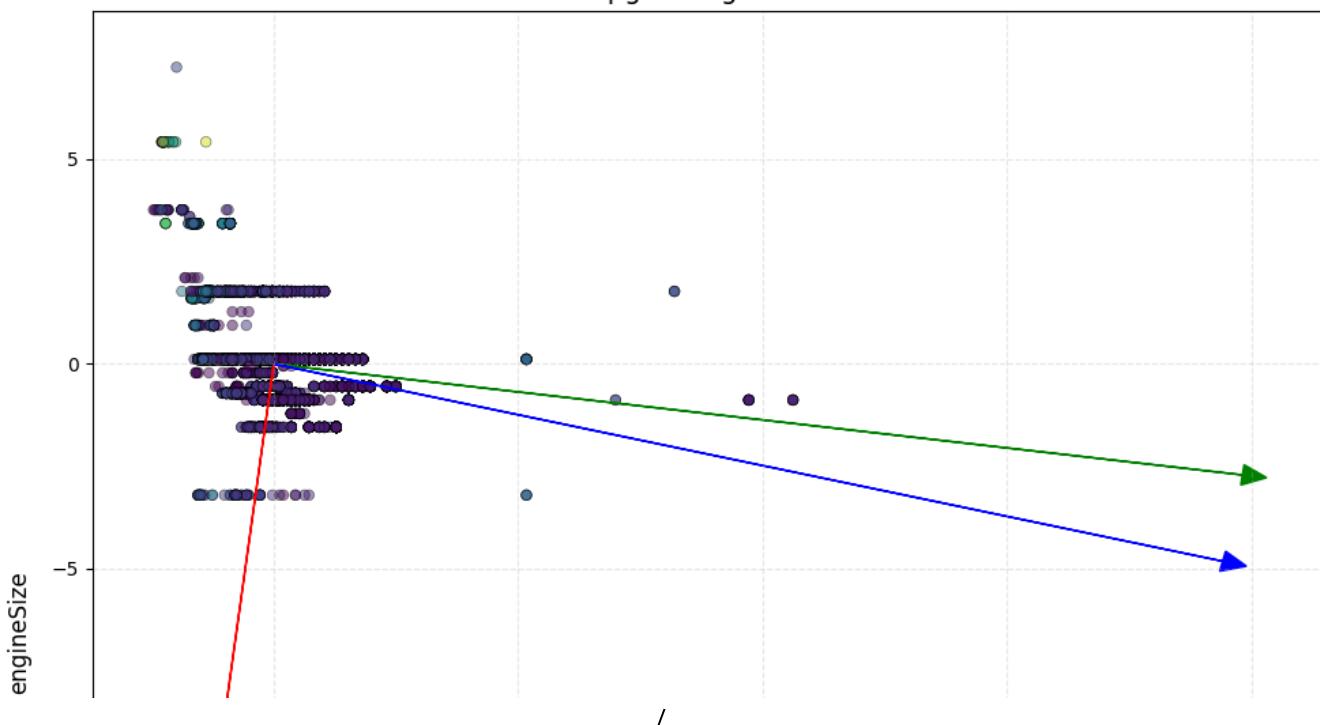
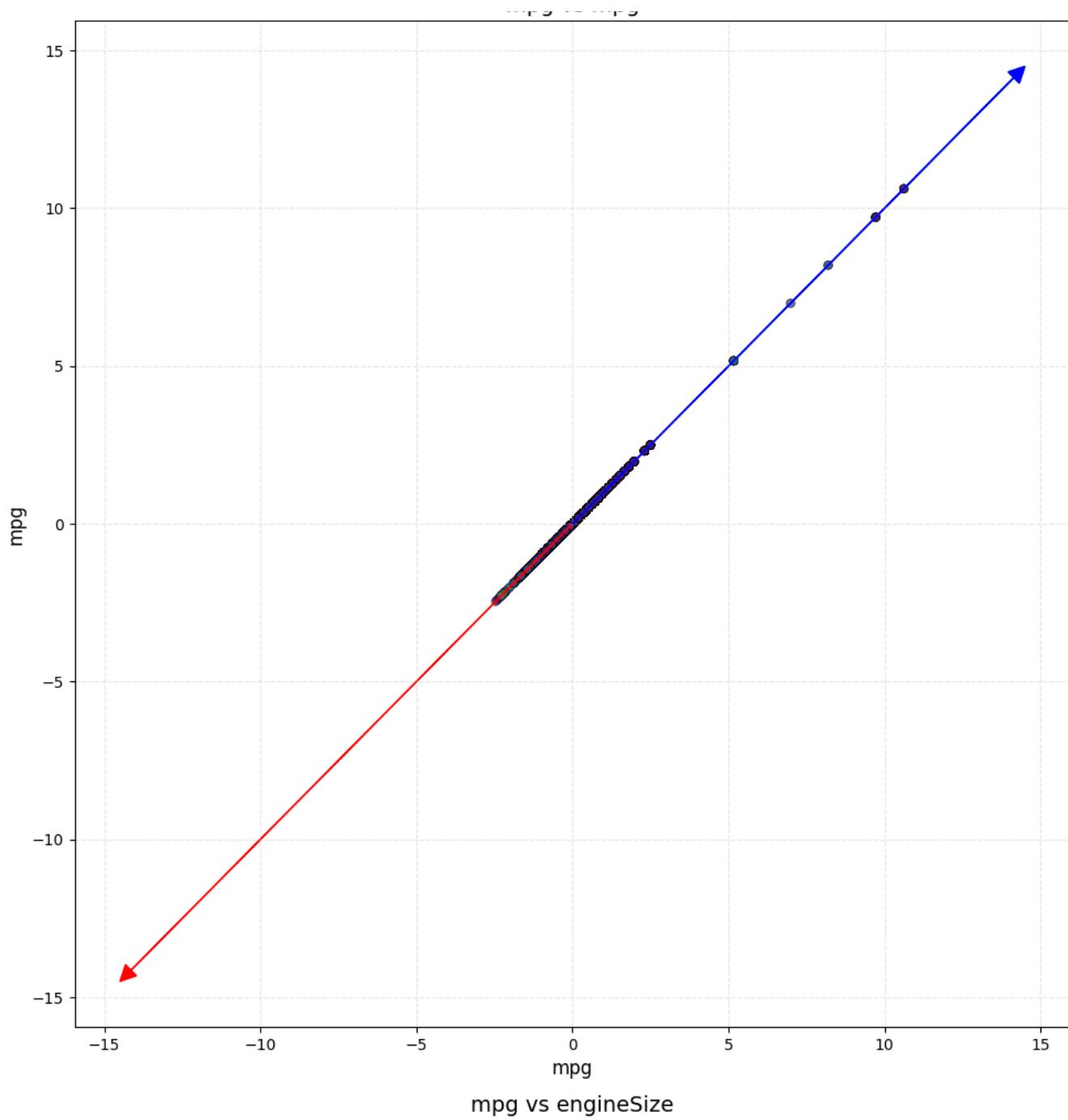


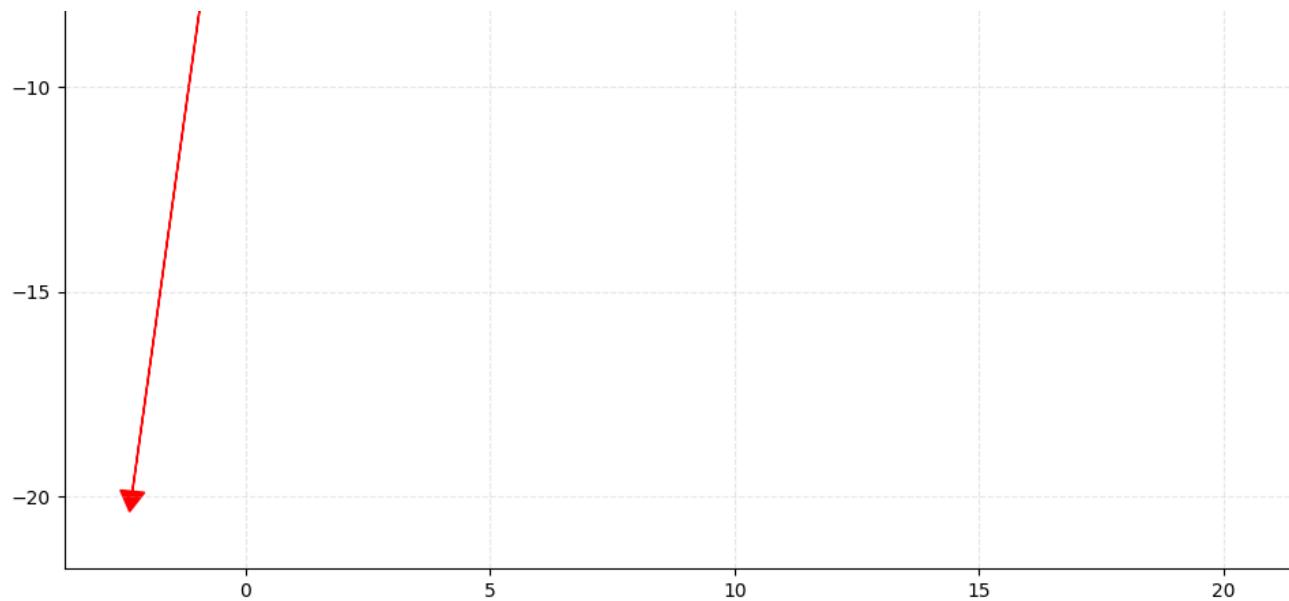


mpg vs tax

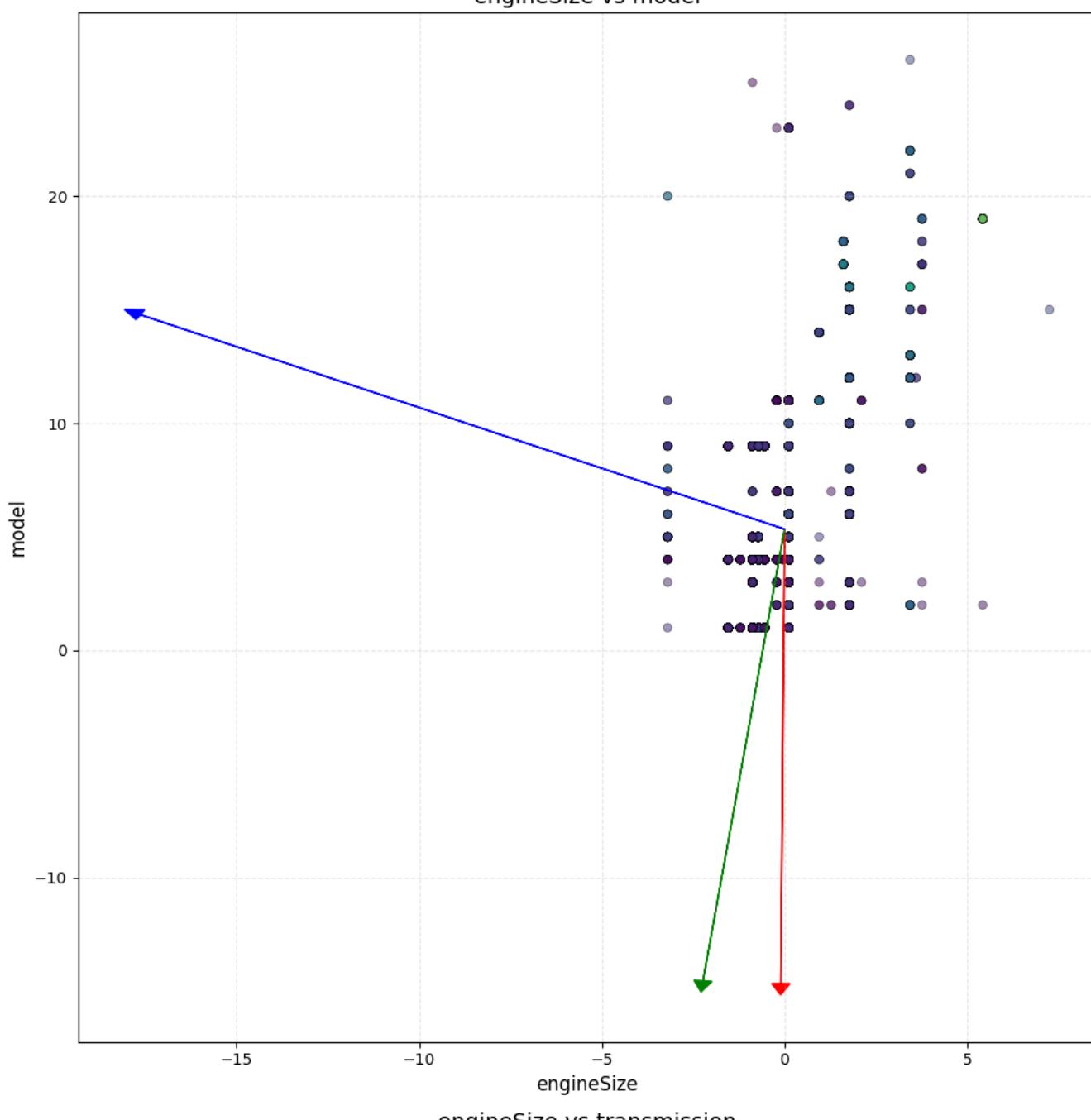


mpg vs tax /

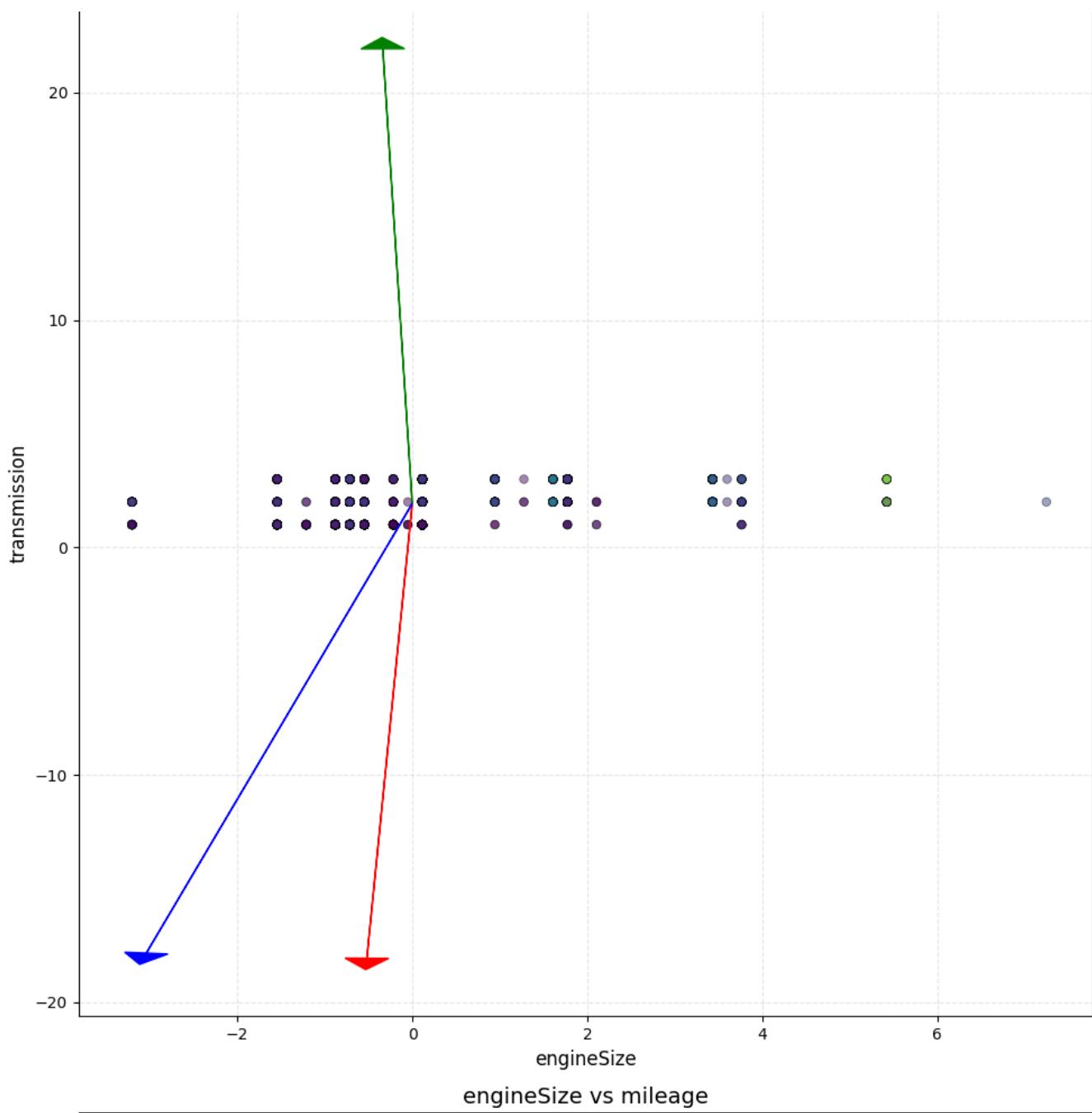




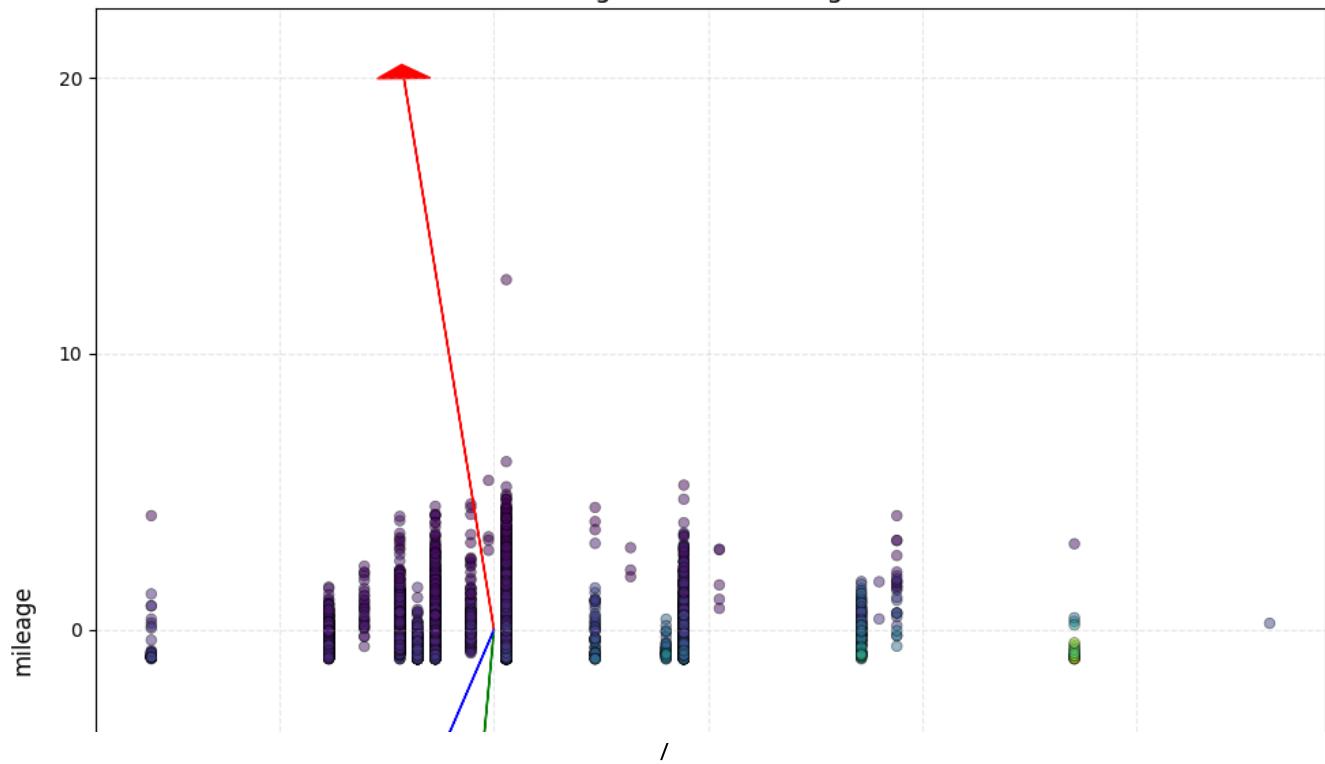
engineSize vs model

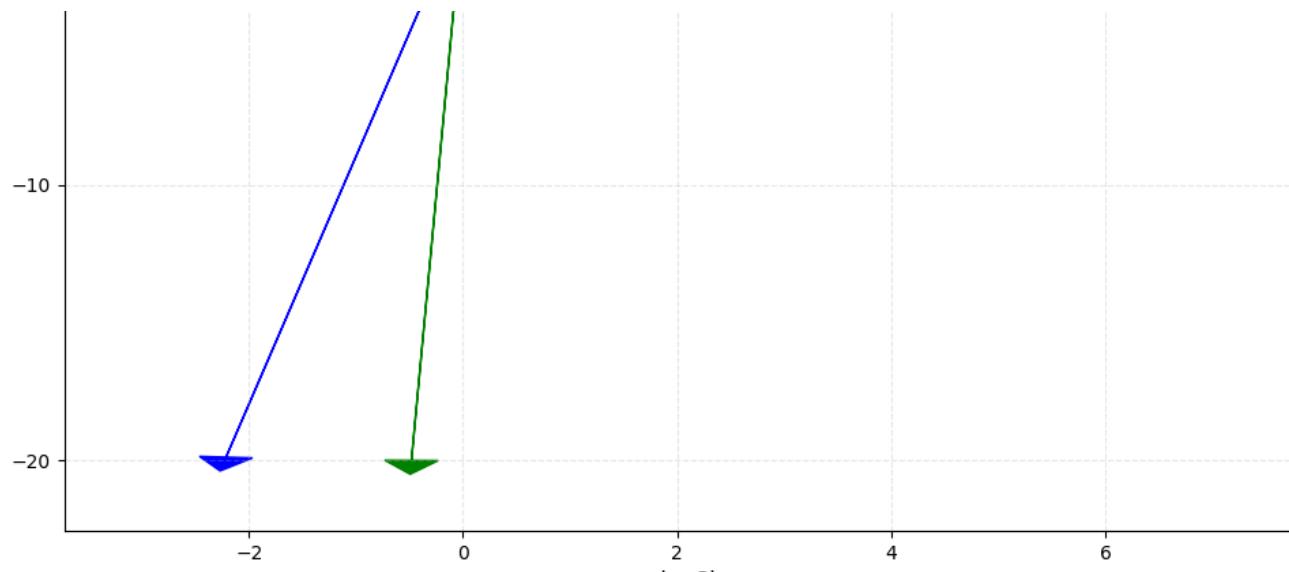


engineSize vs transmission

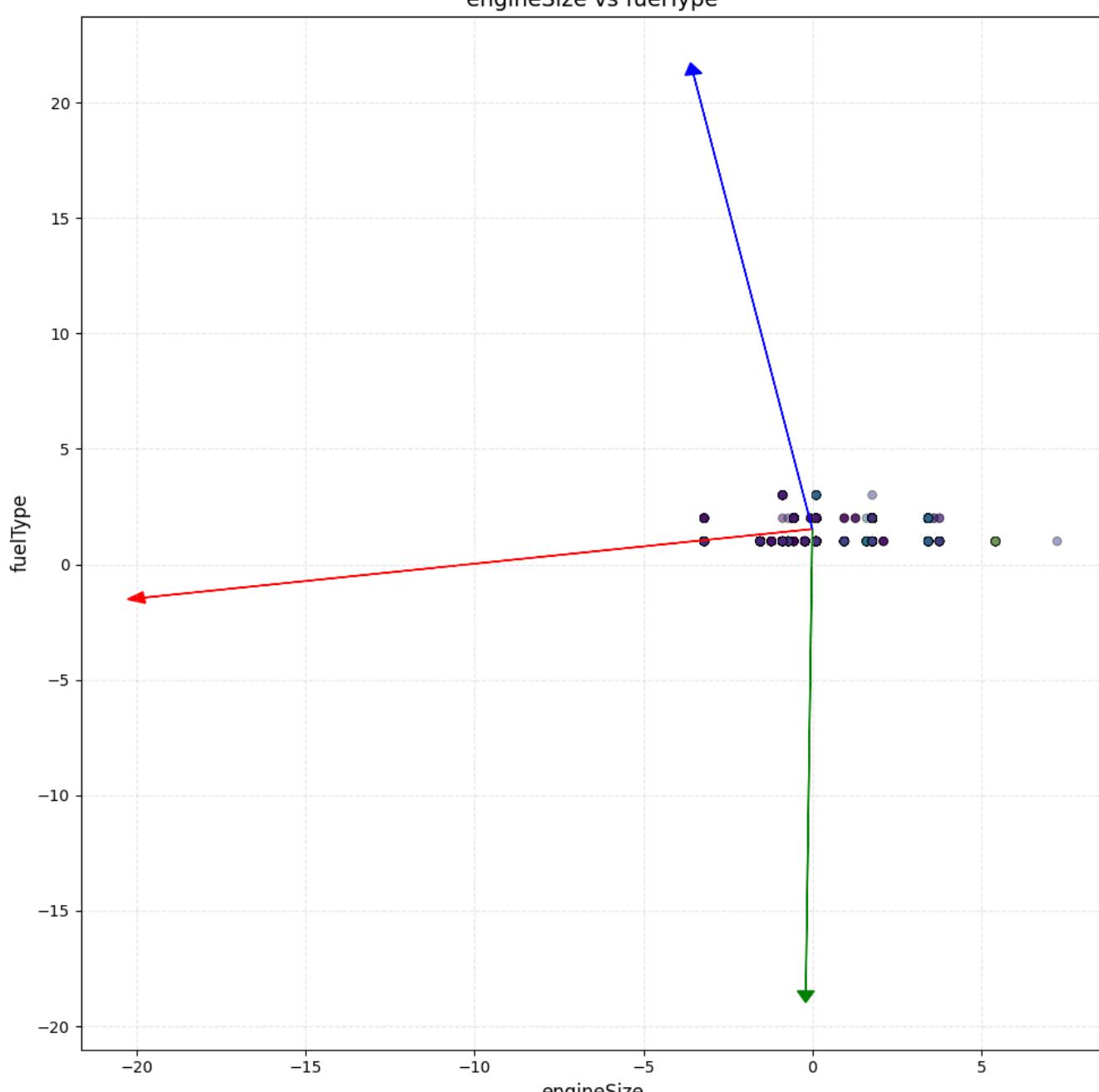


engineSize vs mileage

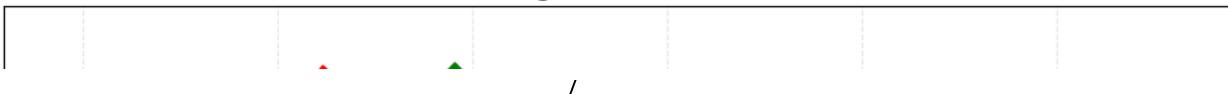


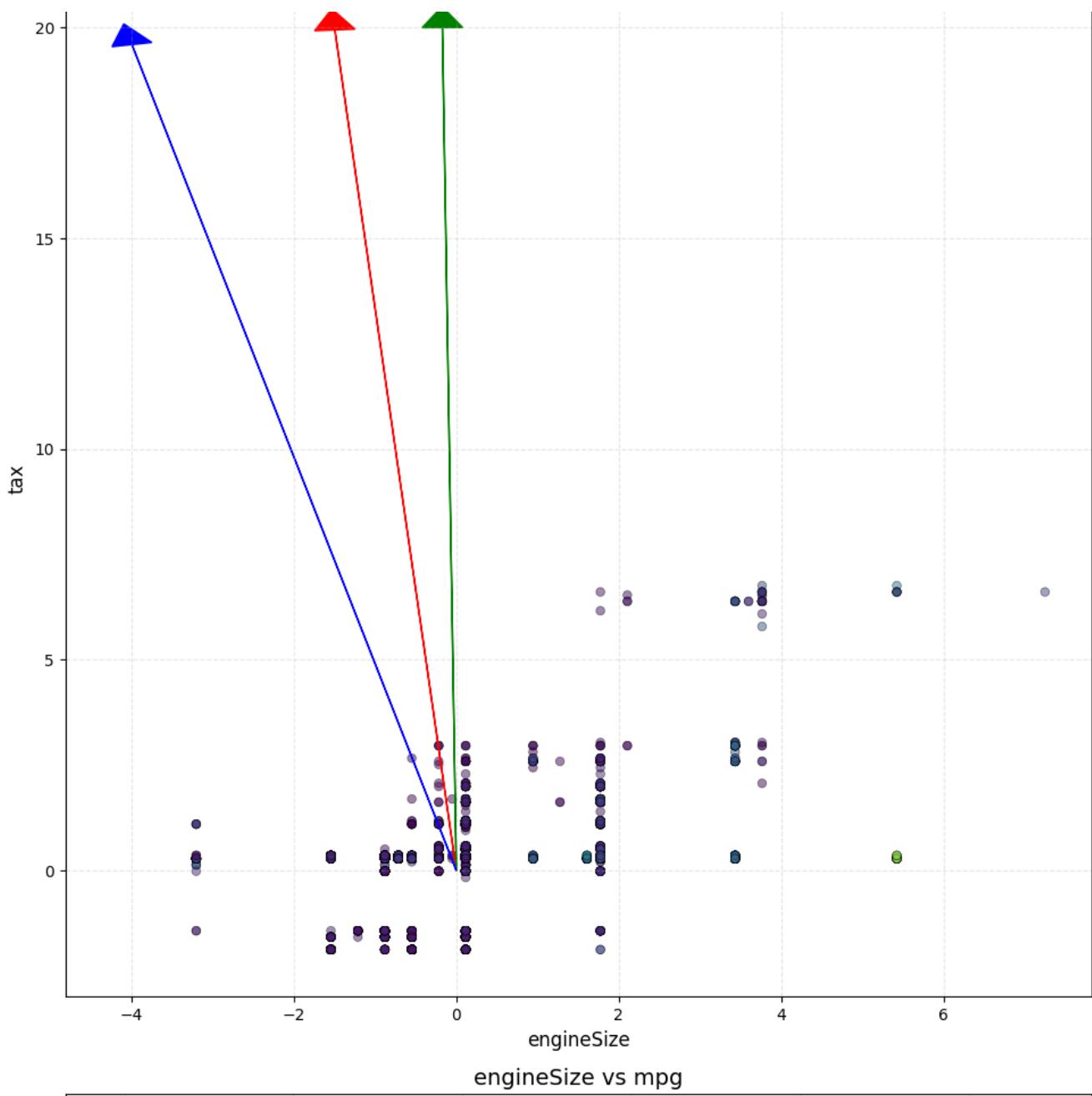


engineSize vs fuelType

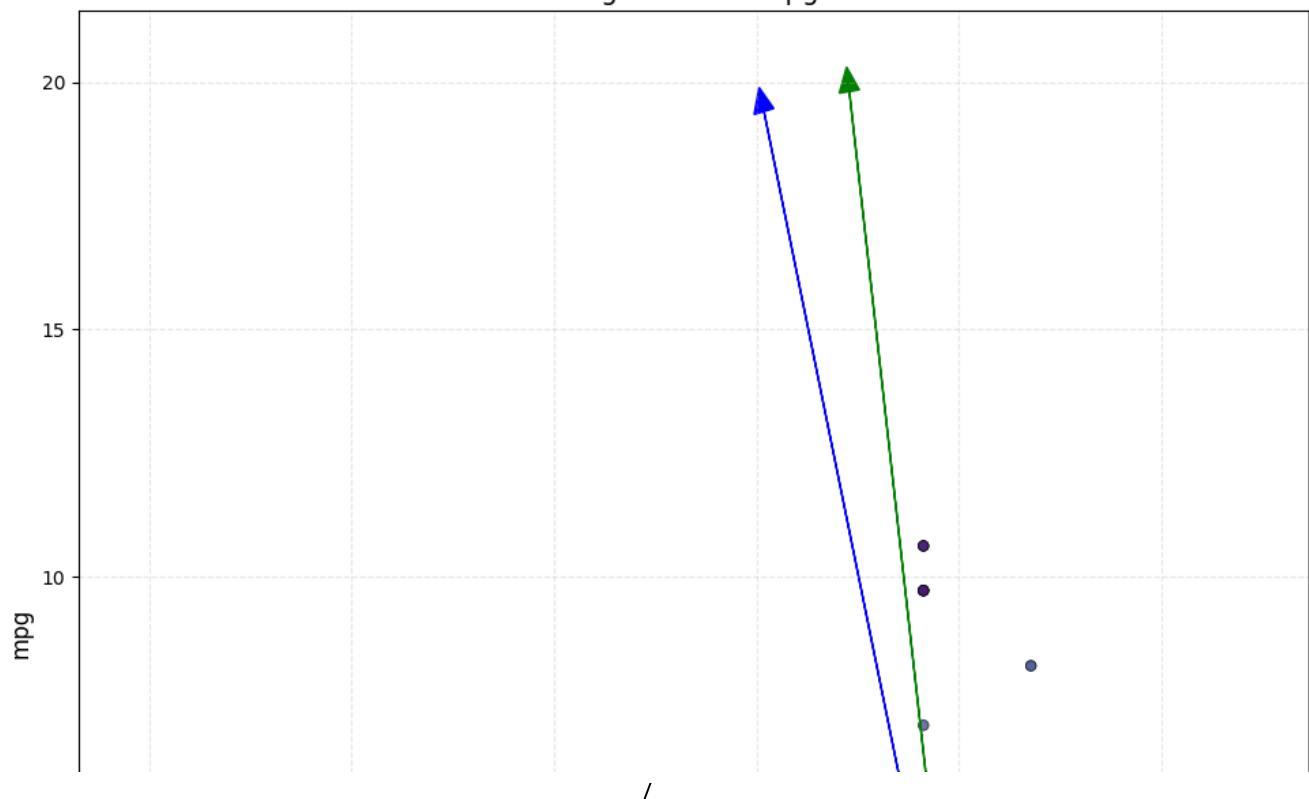


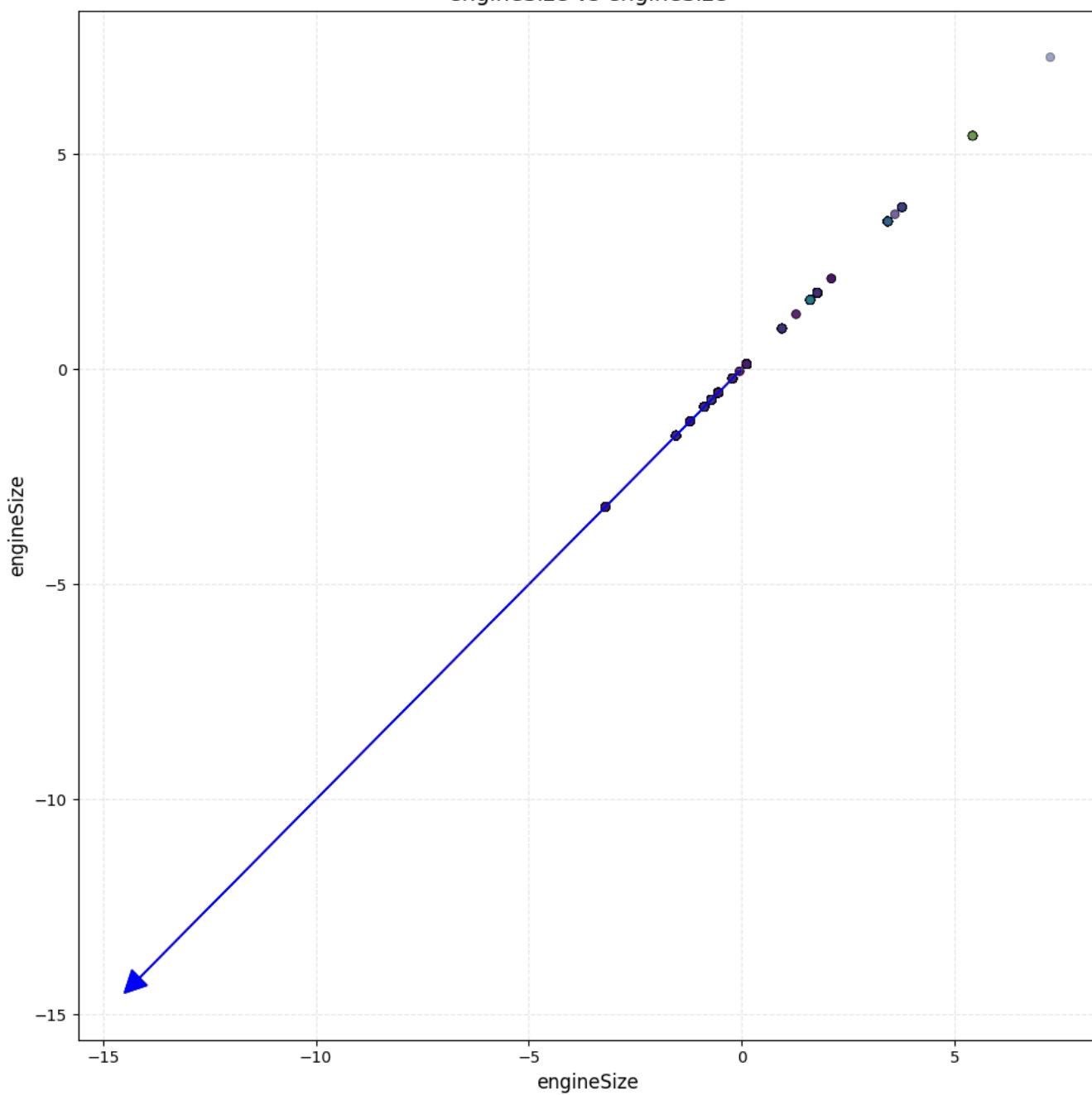
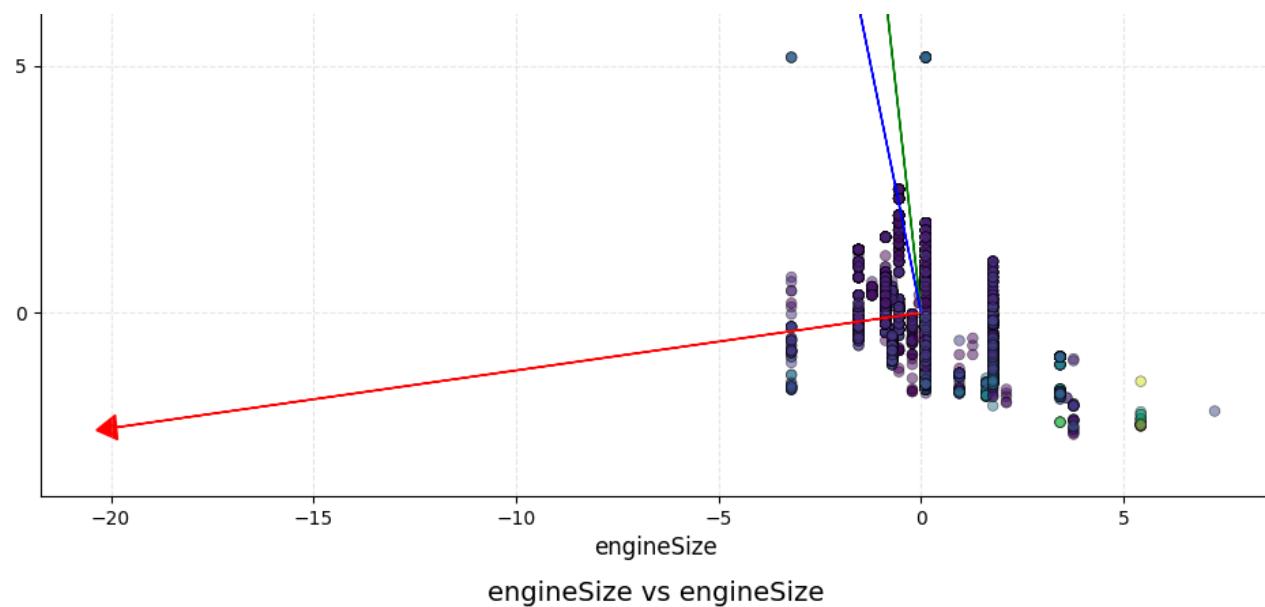
engineSize vs tax





engineSize vs mpg

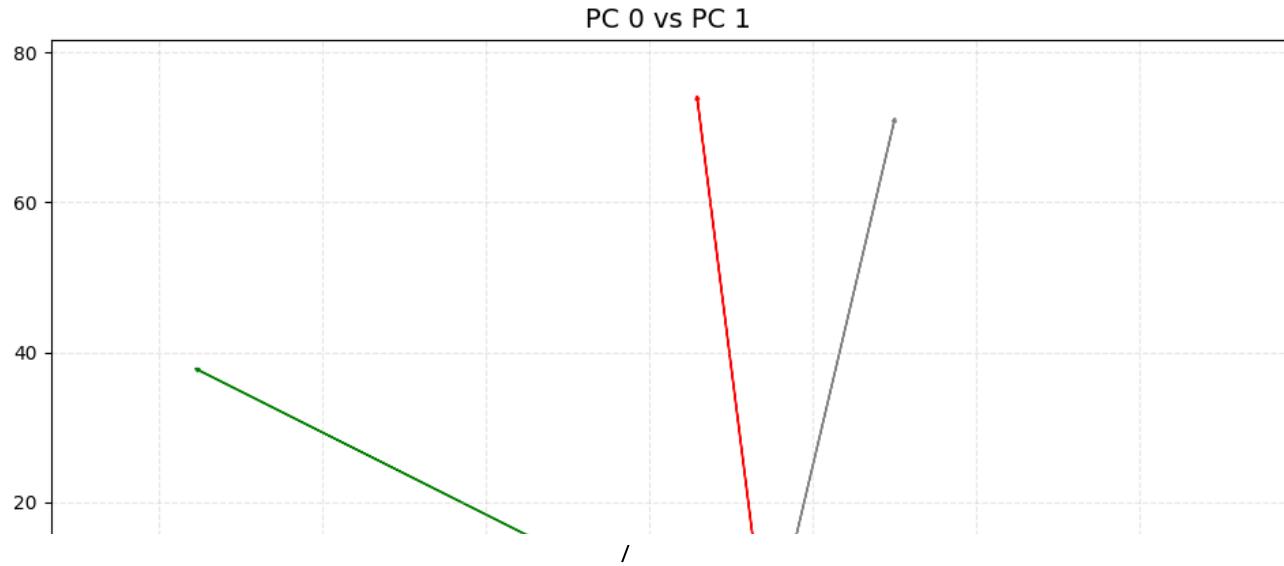
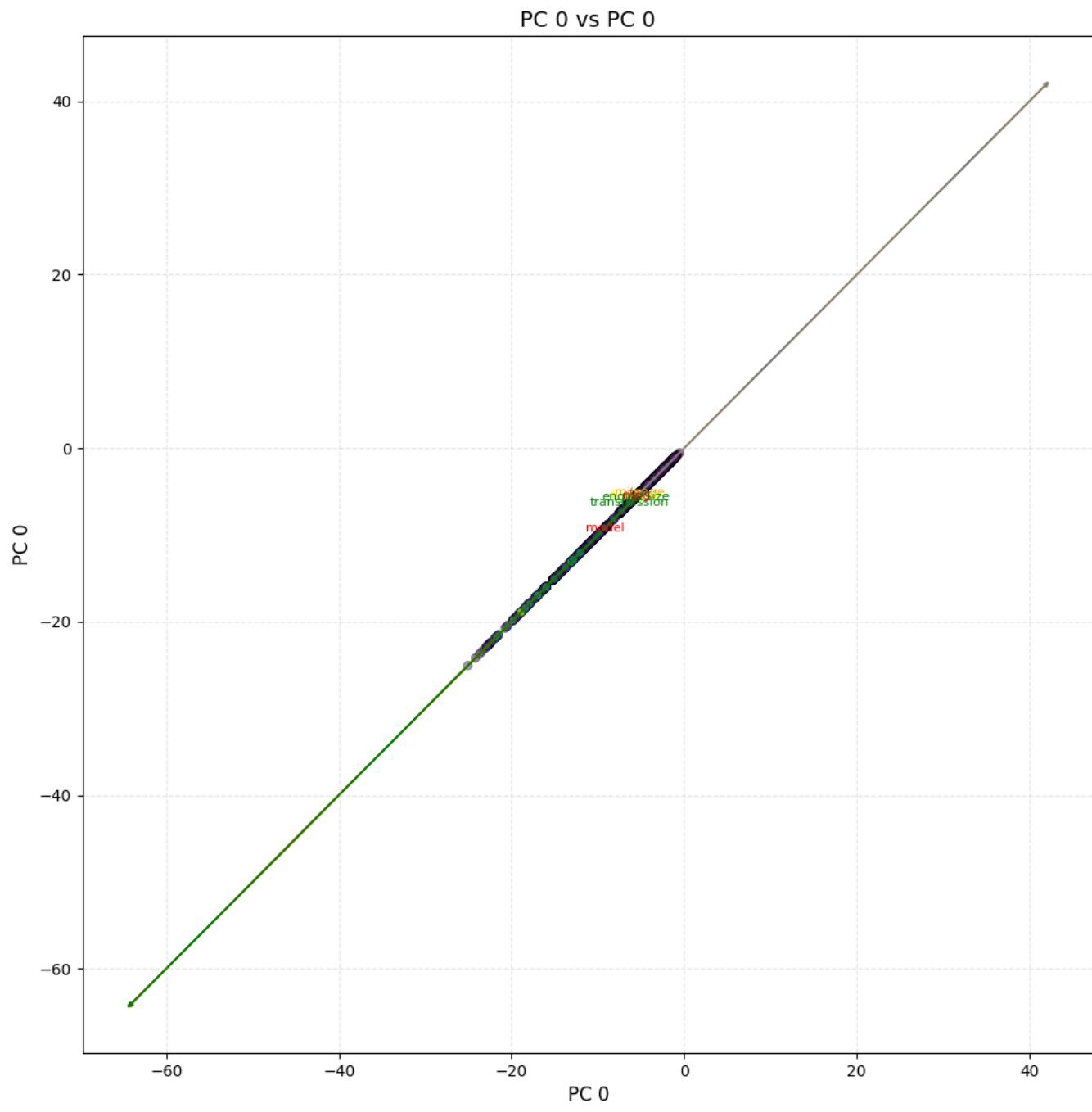


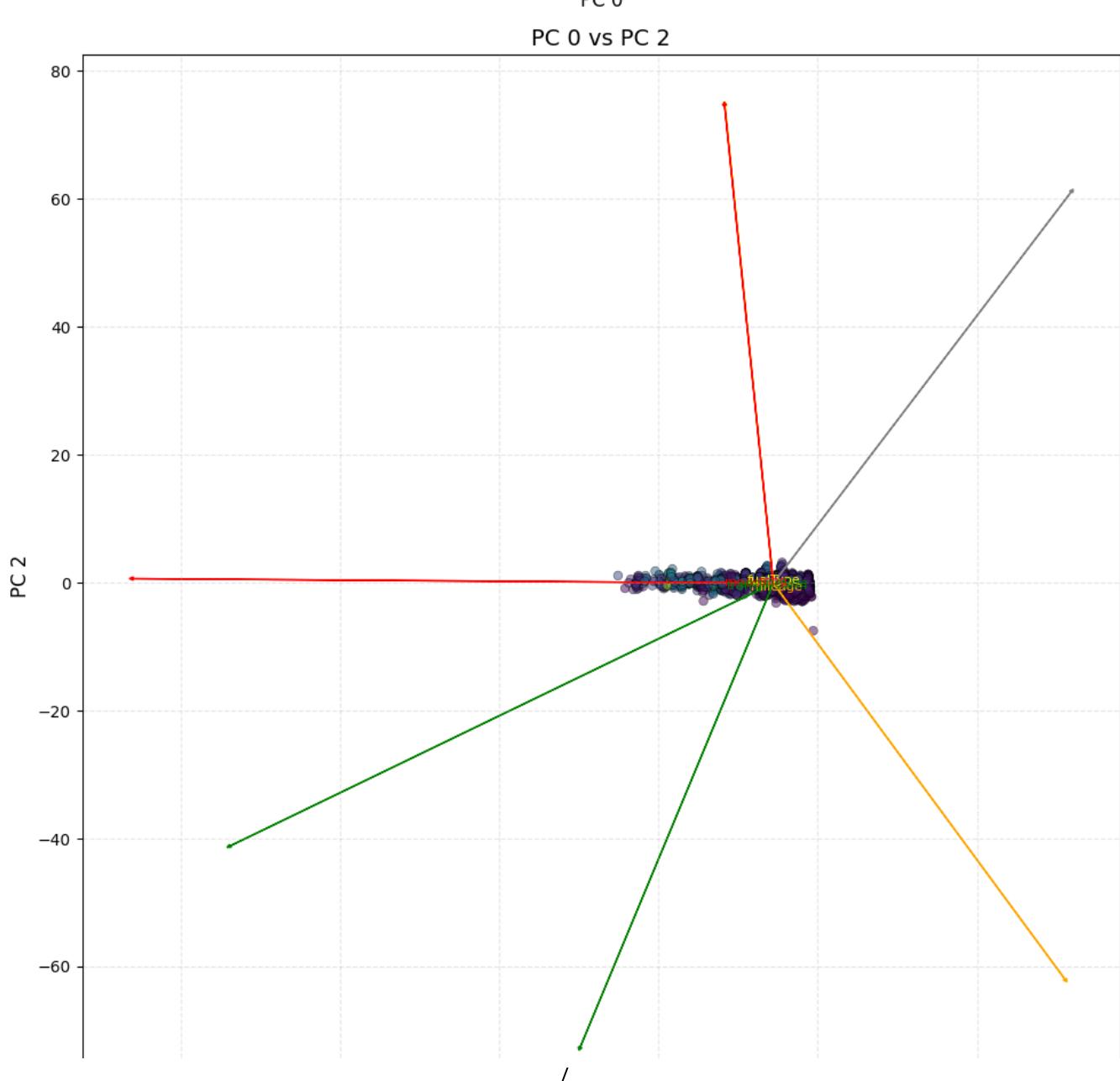
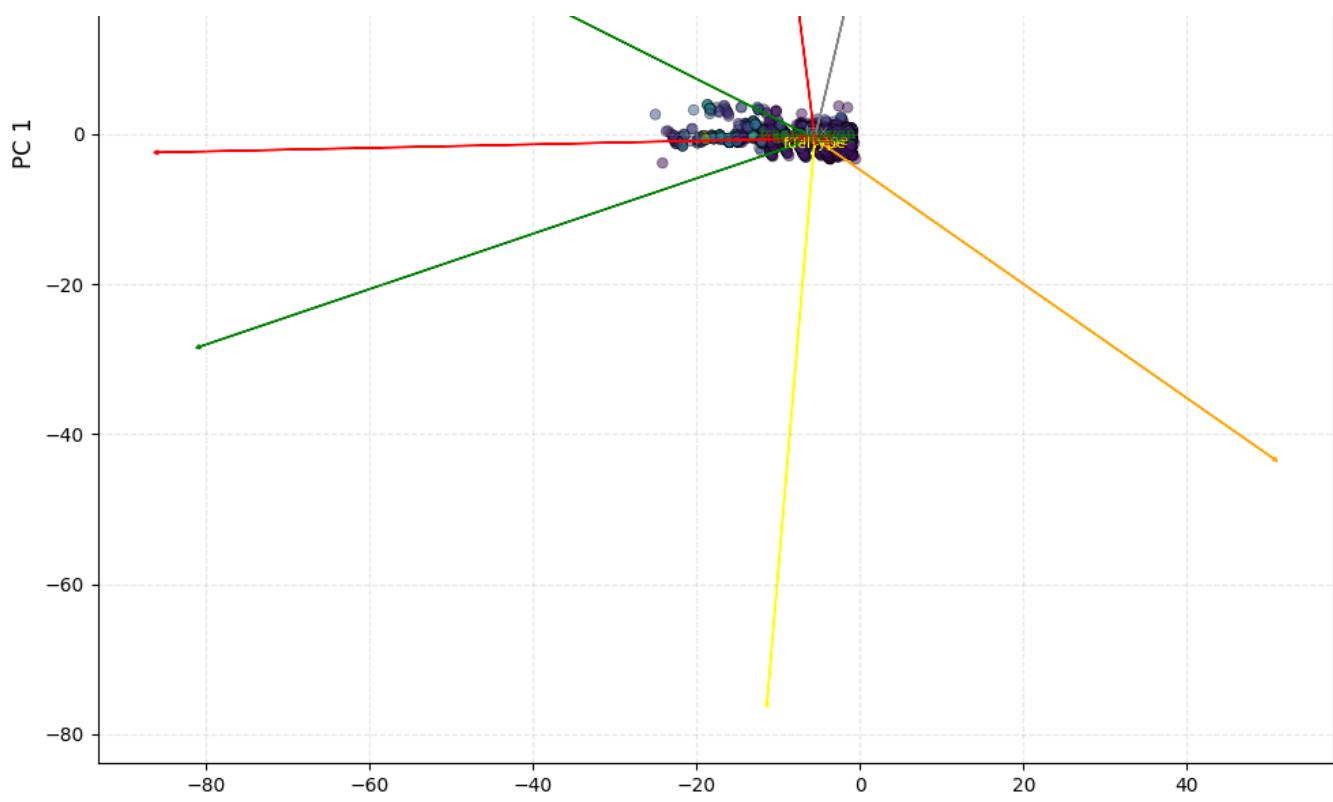


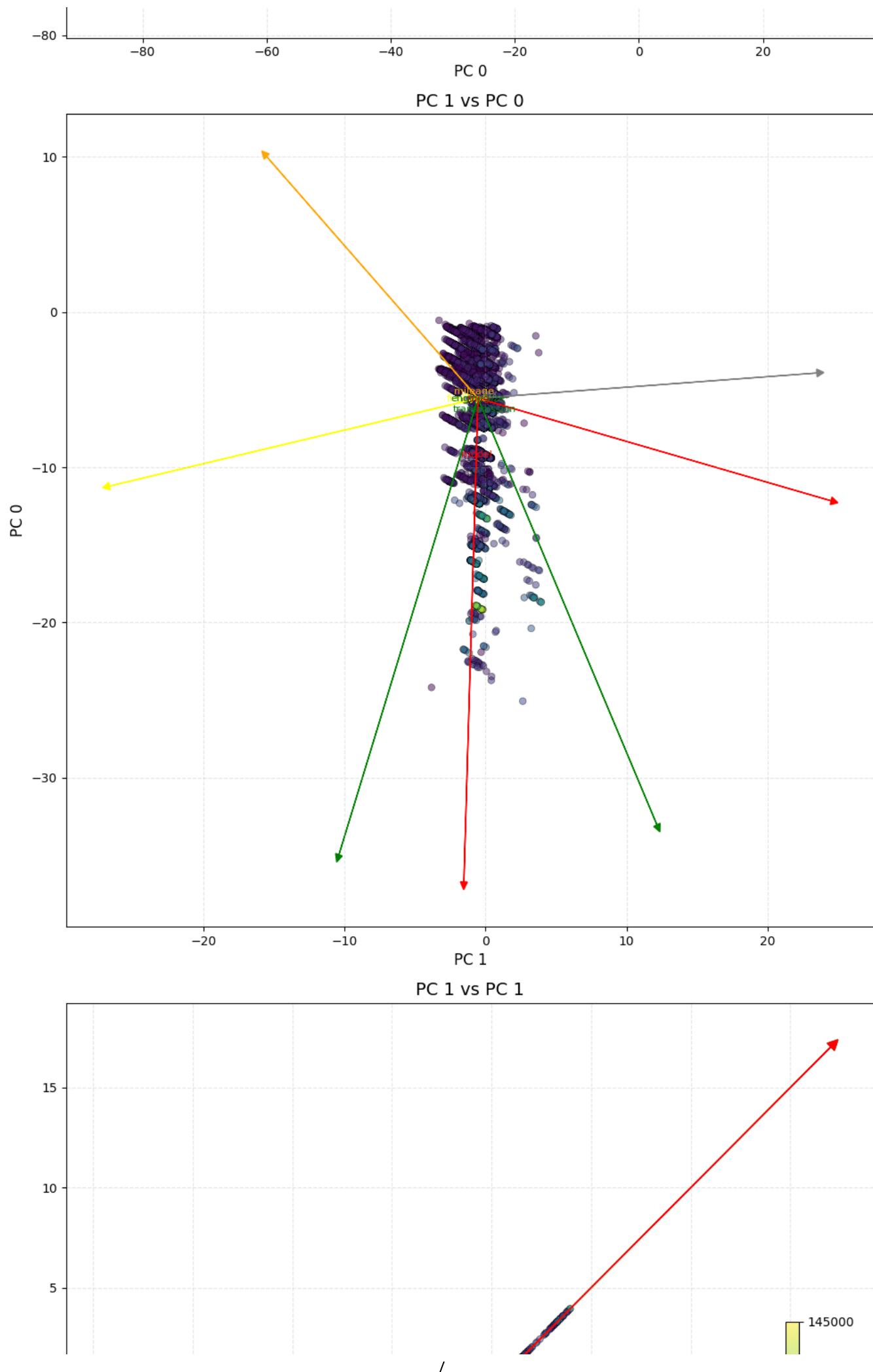
- Analyzed the directions of the vectors representing principal components on the pair plots. The orientation and length of these vectors illustrate the influence of each component on specific feature

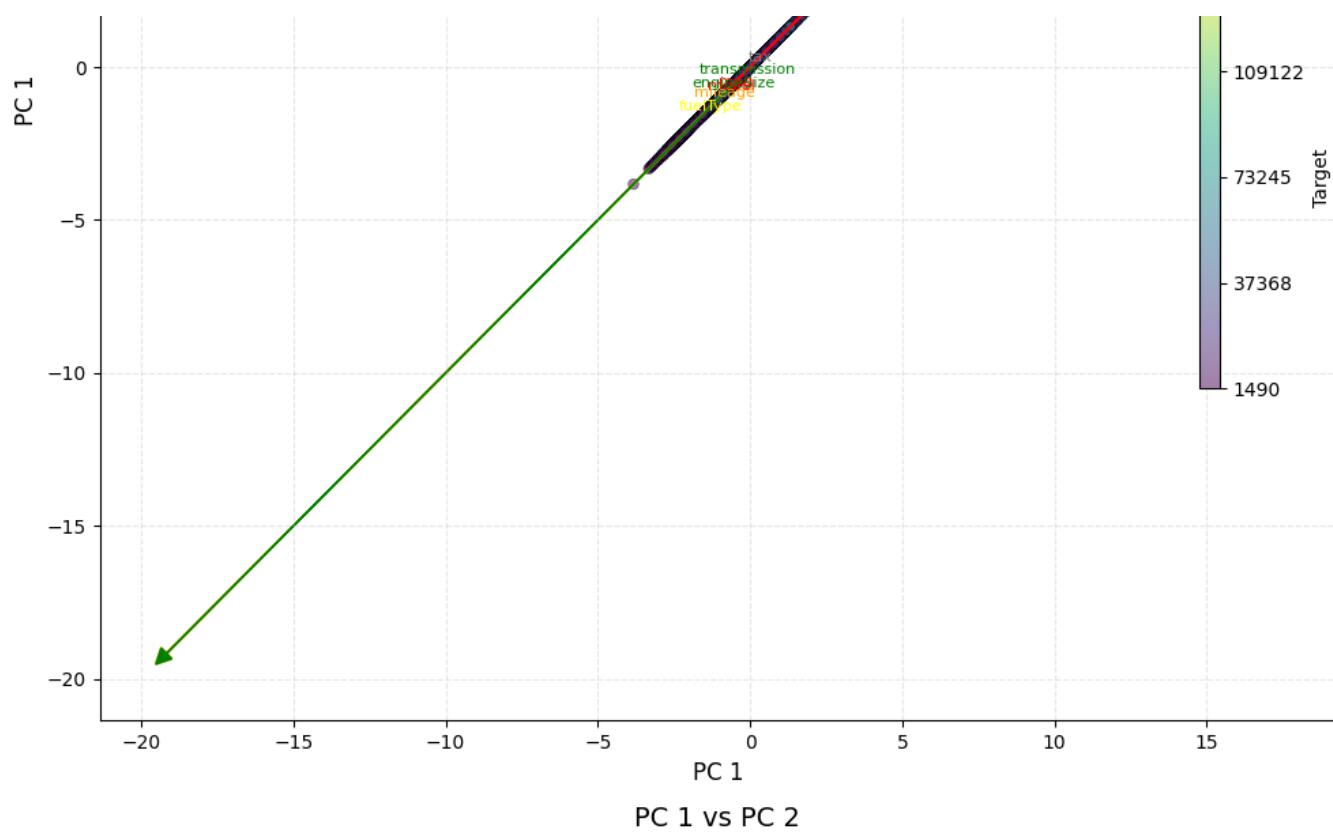
pairs, aiding in the identification of dominant patterns.

#### d. Identification of Key Relationships:

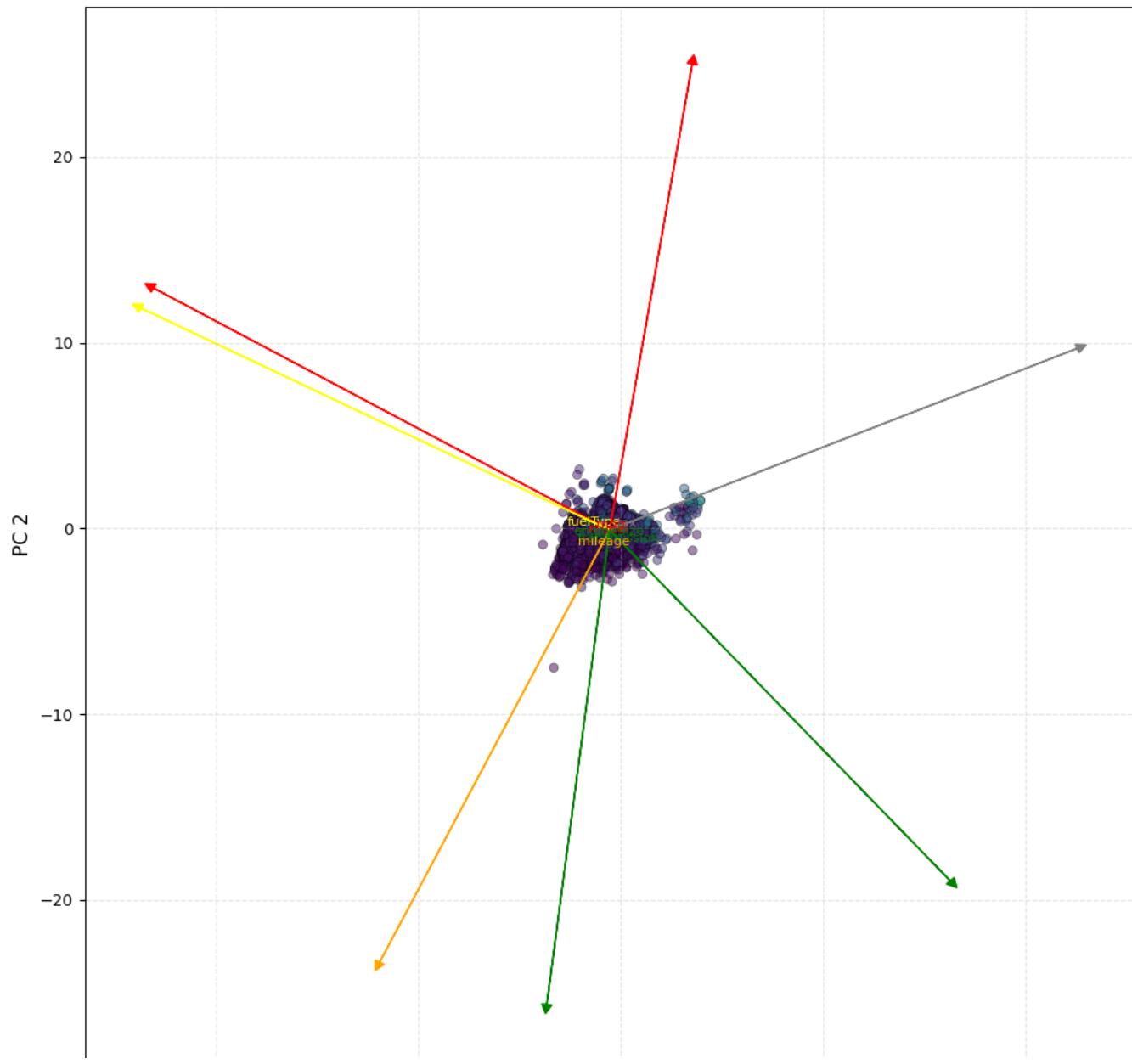


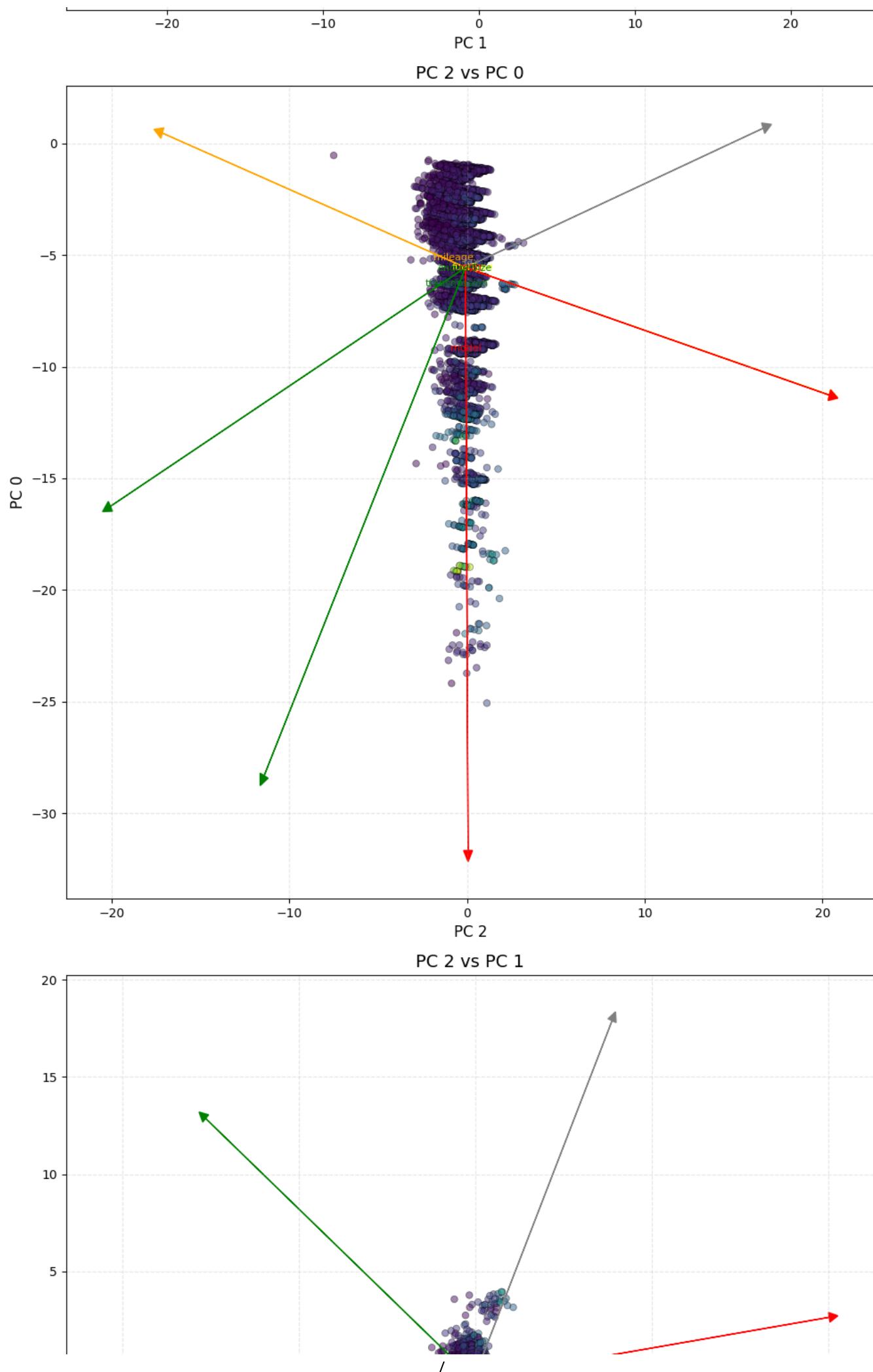


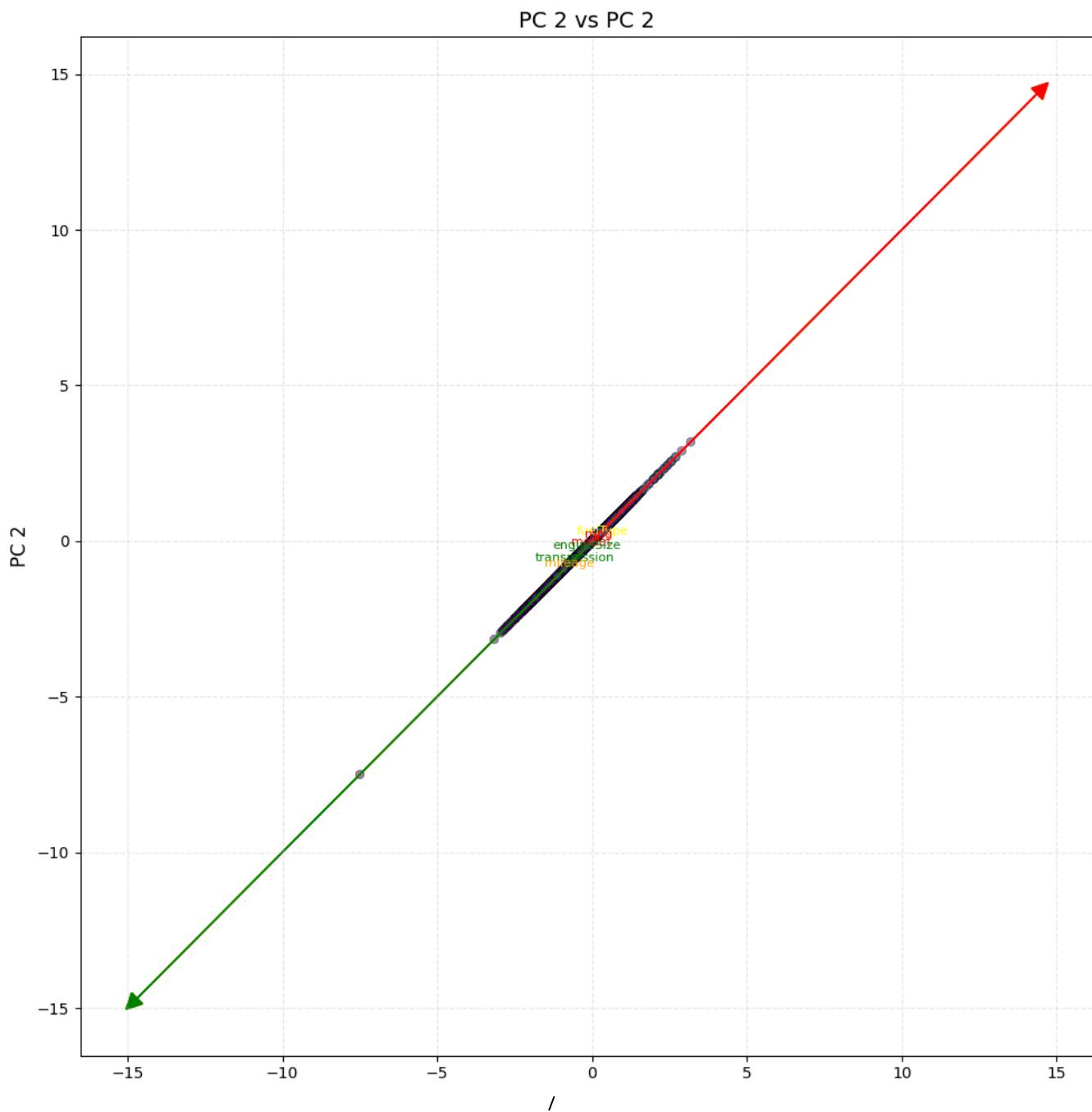
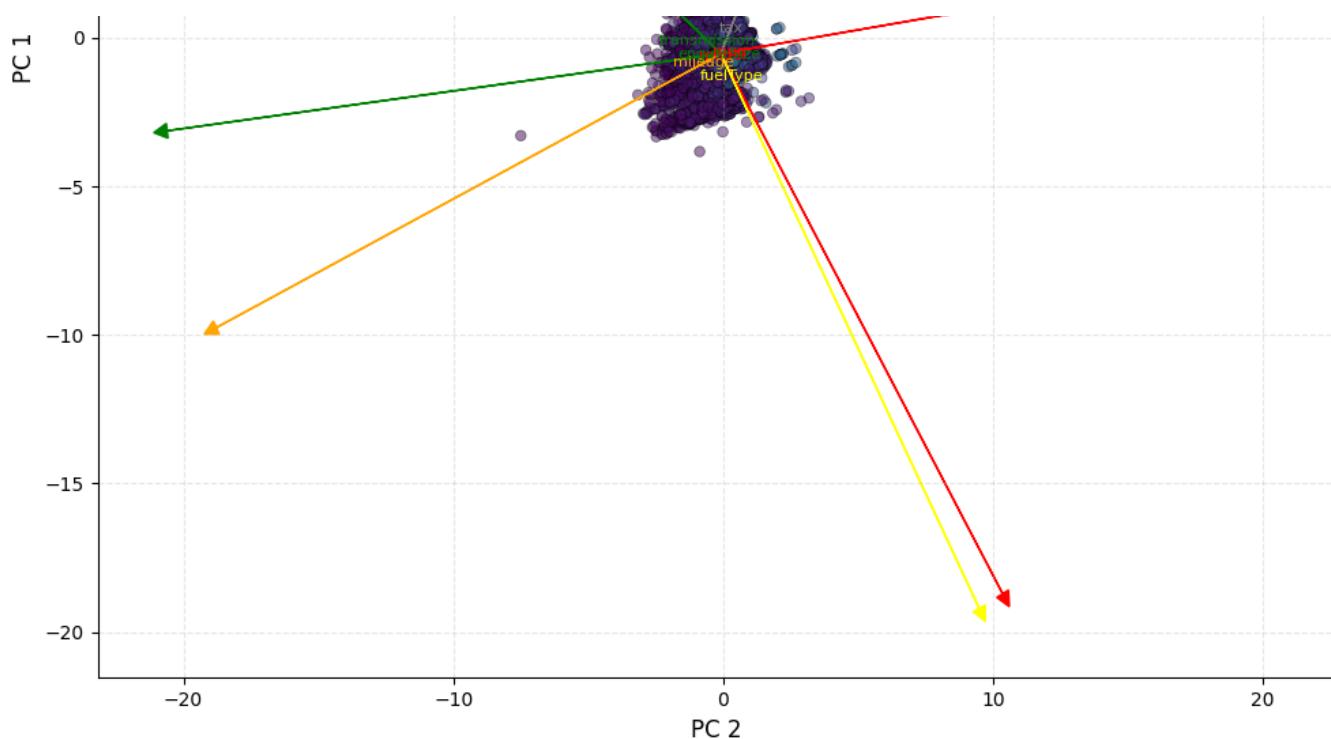




PC 1 vs PC 2







## PC 2

- Examined the pair plots to identify key relationships between features and how they align with the principal components. This step is crucial for understanding the impact of each principal component on different aspects of the data distribution.

### e. Insights for Feature Importance:

- Derived insights into the importance of features by observing how they align with the directions of principal components. Features aligned with dominant components contribute more significantly to the overall variability in the dataset.

## 8. Testing on Linear Regression

### a. Model Training with Original Data

#### Model Definition:

- Defined functions for linear regression model training, prediction, Mean Squared Error (MSE), and Mean Absolute Error (MAE). These functions facilitate the implementation of a gradient descent-based linear regression model.

#### Training on Original Data:

- Applied the linear regression model to the original data without dimensionality reduction.
- Utilized the defined functions to train the model, make predictions, and calculate the Root Mean Squared Error (RMSE) for performance evaluation.

#### Result Interpretation:

- Assessed the model's performance on the original data by analyzing the computed RMSE. This metric quantifies the accuracy of the model's predictions and provides insights into how well it generalizes to unseen data.

### b. Model Training with PCA-transformed Data

#### Transformation with PCA:

- Utilized Principal Component Analysis (PCA) to transform the original data into a reduced-dimensional space.

#### Training on Transformed Data:

- Applied the linear regression model to the PCA-transformed data.
- Employed the previously defined functions for model training, prediction, and RMSE calculation on the transformed features.

#### Comparison of Results:

- Compared the RMSE obtained from the model trained on the original data with that from the model trained on the PCA-transformed data.
- Analyzed any differences in performance and runtime, providing insights into the trade-offs between dimensionality reduction and model accuracy.

## 8. Conclusion and Analysis

### Analysis Summary

#### Tasks Performed

##### 1. Data Preprocessing:

- Converted categorical columns to numerical using mapping.
- Removed unnecessary 'year' column.

##### 2. Eigenvalue and Eigenvector Calculation:

- Calculated eigenvalues and eigenvectors using the covariance matrix.
- Sorted the eigenvalues and corresponding eigenvectors in descending order.

##### 3. Principal Component Analysis (PCA):

- Chose a specific number of principal components ( $k=3$ ).
- Extracted the selected eigenvectors and eigenvalues.

##### 4. Explained Variance:

- Calculated the explained variance for each component.
- Computed the cumulative explained variance.
- Determined the number of components explaining at least 90% of the variance.

##### 5. Sequential Variance Increase:

- Calculated and visualized the variance captured by each component.

##### 6. Visualization using Pair Plots:

- Created pair plots for visualizing relationships between features.
- Projected principal components onto pair plots as vectors.

##### 7. Testing on Linear Regression:

- Defined functions for linear regression model training, prediction, MSE, and MAE.
- Trained linear regression models on both the original and PCA-transformed data.
- Compared the performance of the models through RMSE values.

### Key Findings

#### 1. Dimensionality Reduction:

- Most of the variance (~91%) was captured using just 3 principal components, significantly reducing the dimensionality from 7 to 3.

## 2. Model Performance:

- Linear regression on the transformed data resulted in a higher RMSE (8652.07 vs. 5817.96) compared to the original data.
- However, the model on the transformed data converged much faster (4.17 seconds vs. 14.47 seconds), showcasing a trade-off between accuracy and computational efficiency.

## Implications and Considerations

- The choice of the number of principal components depends on the balance between dimensionality reduction and maintaining an acceptable level of model accuracy.
- Understanding the trade-offs between model performance and computational efficiency is crucial, especially in large datasets.
- The visualizations provide valuable insights into feature relationships and the impact of dimensionality reduction on data representation.

## Recommendations

- Consider the specific requirements of the task at hand when deciding the number of principal components.
- Evaluate the computational constraints and runtime considerations, especially in real-time or resource-intensive applications.

## Conclusion

The analysis demonstrated the efficacy of Principal Component Analysis (PCA) in reducing dimensionality while capturing a significant portion of the data's variance. The trade-offs between model performance and computational efficiency highlight the importance of thoughtful dimensionality reduction strategies in various data science applications.