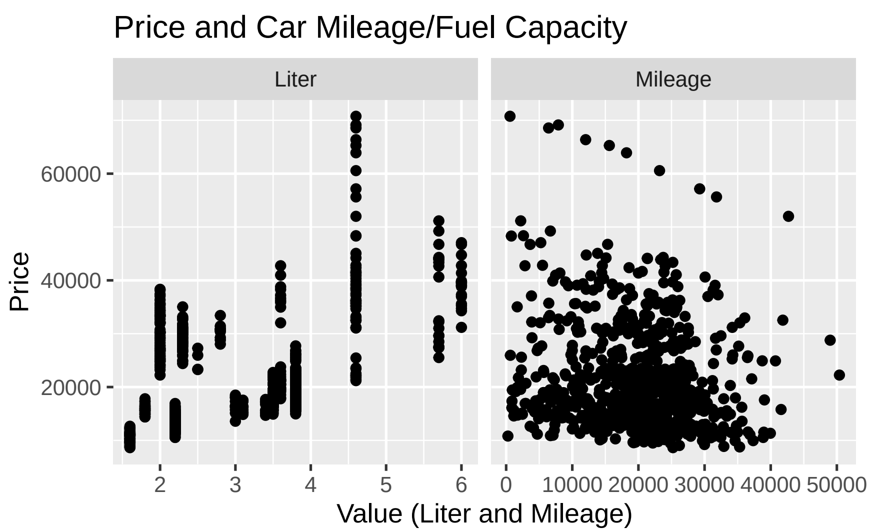
```
car_prices %>%
  pivot_longer(
    cols = Mileage | Liter,
    names_to = "name",
    values_to = "value") %>%
  ggplot() +
  geom_point(mapping = aes(x = value, y = Price)) +
  facet_wrap(~name, scales = "free_x") +
  labs(title = "Price and Car Mileage/Fuel Capacity",
       x = "Value (Liter and Mileage)",
       y = "Price")
```

Price and Car Mileage VS. Fuel Capacity




```
continuous_model <- lm(Price ~ Mileage + Liter, data = car_prices)
continuous_model %>%
  glance() %>%
  select(r.squared)
```

Continuous Model

```
## # A tibble: 1 x 1
##   r.squared
##       <dbl>
## 1     0.329
```

```
continuous_model %>%
  tidy()
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 9427.    1095.          8.61 3.90e-17
## 2 Mileage       -0.160    0.0349     -4.58 5.28e- 6
## 3 Liter       4968.     259.         19.2  7.33e-68
```

```
# predict model plane over values
lit <- unique(car_prices$Liter)
mil <- unique(car_prices$Mileage)
grid <- with(car_prices, expand.grid(lit, mil))
d <- setNames(data.frame(grid), c("Liter", "Mileage"))
vals <- predict(continuous_model, newdata = d)

# form surface matrix and give to plotly
m <- matrix(vals, nrow = length(unique(d$Liter)), ncol = length(unique(d$Mileage)))
p <- plot_ly() %>%
  add_markers(
    x = ~car_prices$Mileage,
    y = ~car_prices$Liter,
    z = ~car_prices$Price,
    marker = list(size = 1)
    ) %>%
  add_trace(
    x = ~mil, y = ~lit, z = ~m, type="surface",
    colorscale=list(c(0,1), c("yellow","yellow")),
    showscale = FALSE
    ) %>%
  layout(
    scene = list(
      xaxis = list(title = "mileage"),
      yaxis = list(title = "liters"),
      zaxis = list(title = "price")
    )
  )
if (!is_pdf) {p}
```

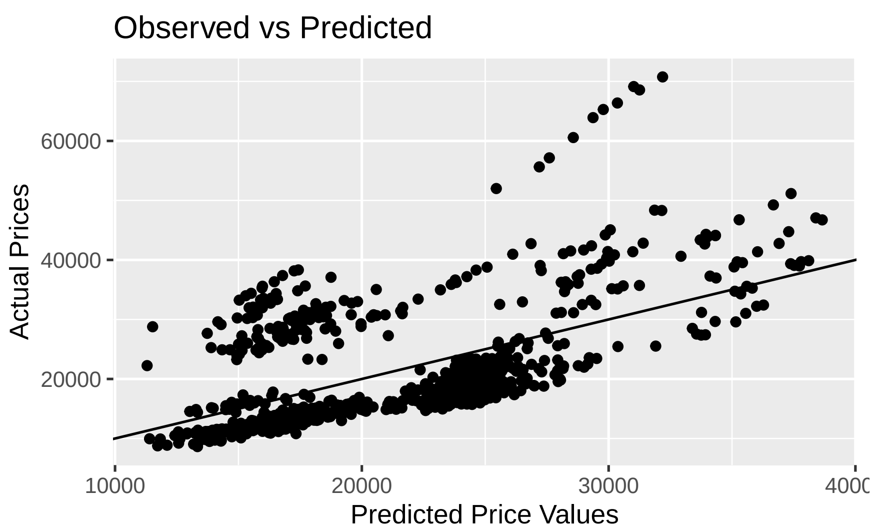
```
continuous_df <- augment(continuous_model, car_prices)
```

3D Model

```
continuous_df %>%
  ggplot() +
  geom_point(mapping = aes(x = .fitted, y = Price)) +
  geom_abline(slope = 1, intercept = 0) +
  labs(title = "Observed vs Predicted",
```

```
       x = "Predicted Price Values",
       y = "Actual Prices")
```

Observed vs. Predicted prices



Linearity: This "Observed vs Predicted" plot's linearity does not capture the actual relationship between the explanatory and response variables because many points deviate significantly from the line, especially as the predicted values increase.

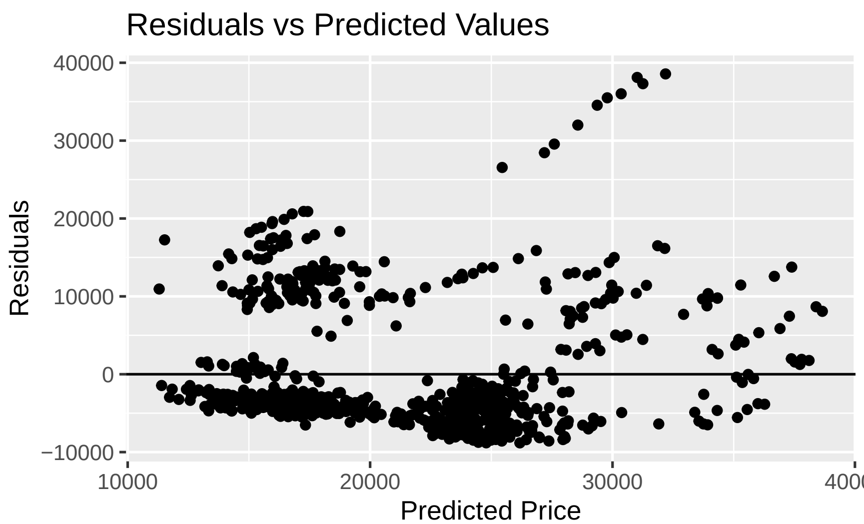
Nearly normal residuals: There is a curve in the spread of points, where the actual prices tend to be higher than predicted for certain ranges.

Constant Variation of Residuals: the spread of actual prices around the predicted line appears to increase as the predicted prices increases. this pattern violates the assumption of constant residual variation.

Independent Observations: The plot doesn't provide enough evidence for or against this assumption.

```
continuous_df %>%
  ggplot() +
  geom_point(mapping = aes(x = .fitted, y = .resid)) +
  geom_hline(yintercept = 0) +
  labs(title = "Residuals vs Predicted Values",
       x = "Predicted Price",
       y = "Residuals")
```

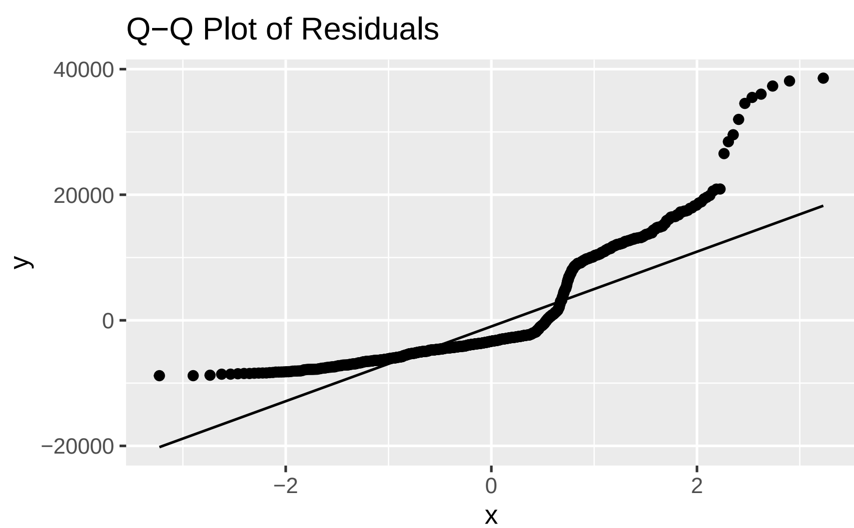
Residuals vs. Predicted values



The spread of residuals increase as the predicted price values increase. For lower predicted prices, the residuals are clustered around the reference line at y = 0, while for higher predicted prices, the residuals show greater spread and variability. Overall, it suggests violation of the constant variability.

```
continuous_df %>%
  ggplot() +
  geom_qq(aes(sample = .resid)) +
  geom_qq_line(aes(sample = .resid)) +
  labs(title = "Q-Q Plot of Residuals")
```

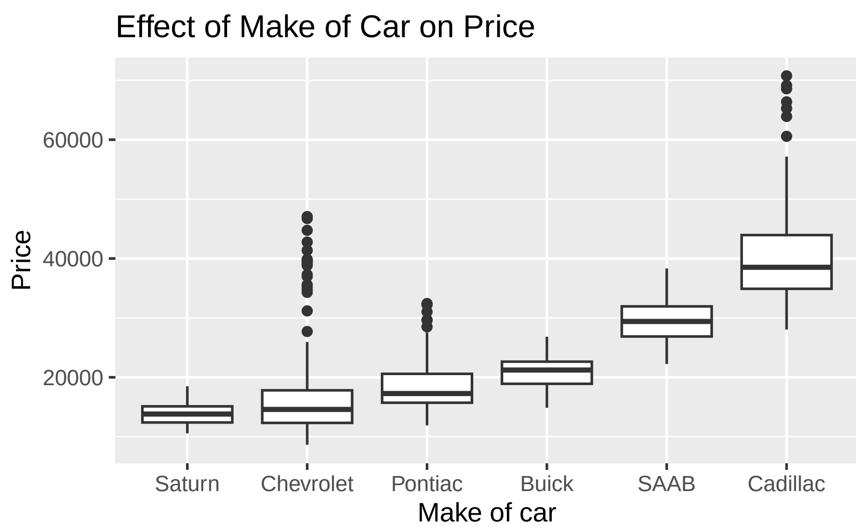
Q-Q Plot of Residuals



If the residuals were nearly normal, the points would closely follow the reference line. In this plot, there are clear deviations from the line, especially in the tails. Residuals have a non-normal distribution.

```r
car_prices %>%
  ggplot() +
  geom_boxplot(aes(x = reorder(Make, Price, FUN=median), y = Price)) +
  labs(x = "Make of car", title = "Effect of Make of Car on Price")
```

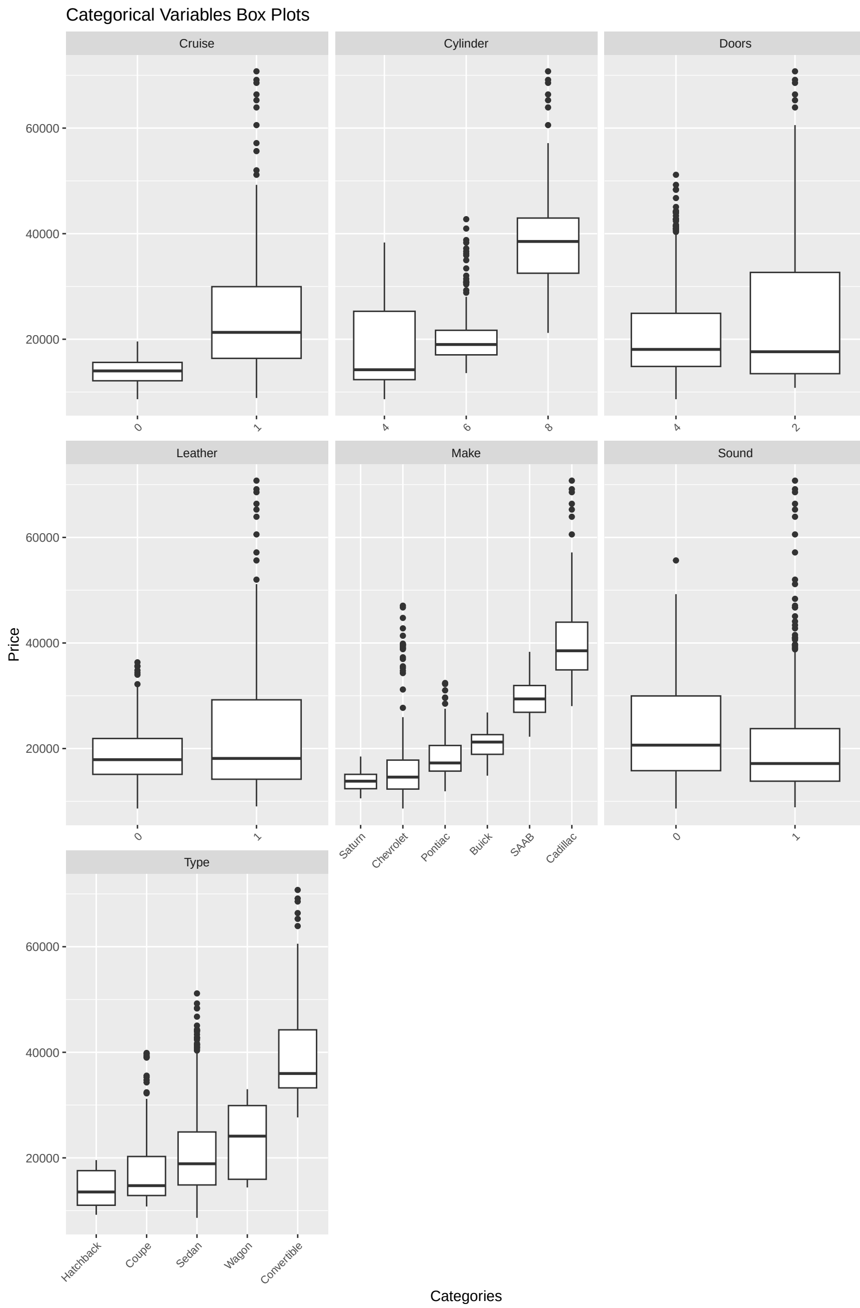
BoxPlot



Saturn has the lowest median price among the car brands shown in this plot. Cadillac has the greatest interquartile range, indicating a wider spread in prices withing the middle 50% of its data. Chevrolet, Cadillac, and Pontiac have outliers. They are indicated by the cirles outside the IQR range.

```r
car_prices %>%
  pivot_longer(
    cols = c(Make, Type, Cylinder, Doors, Cruise, Sound, Leather),
    names_to = "name",
    values_to = "value",
    values_transform = list(value = as.factor)
  ) %>%
  ggplot() +
  geom_boxplot(aes(x = reorder(value, Price, FUN=median), y = Price)) +
  facet_wrap(~name, scales = "free_x") +
  labs(title = "Categorical Variables Box Plots",
       x = "Categories",
       y = "Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8))
```

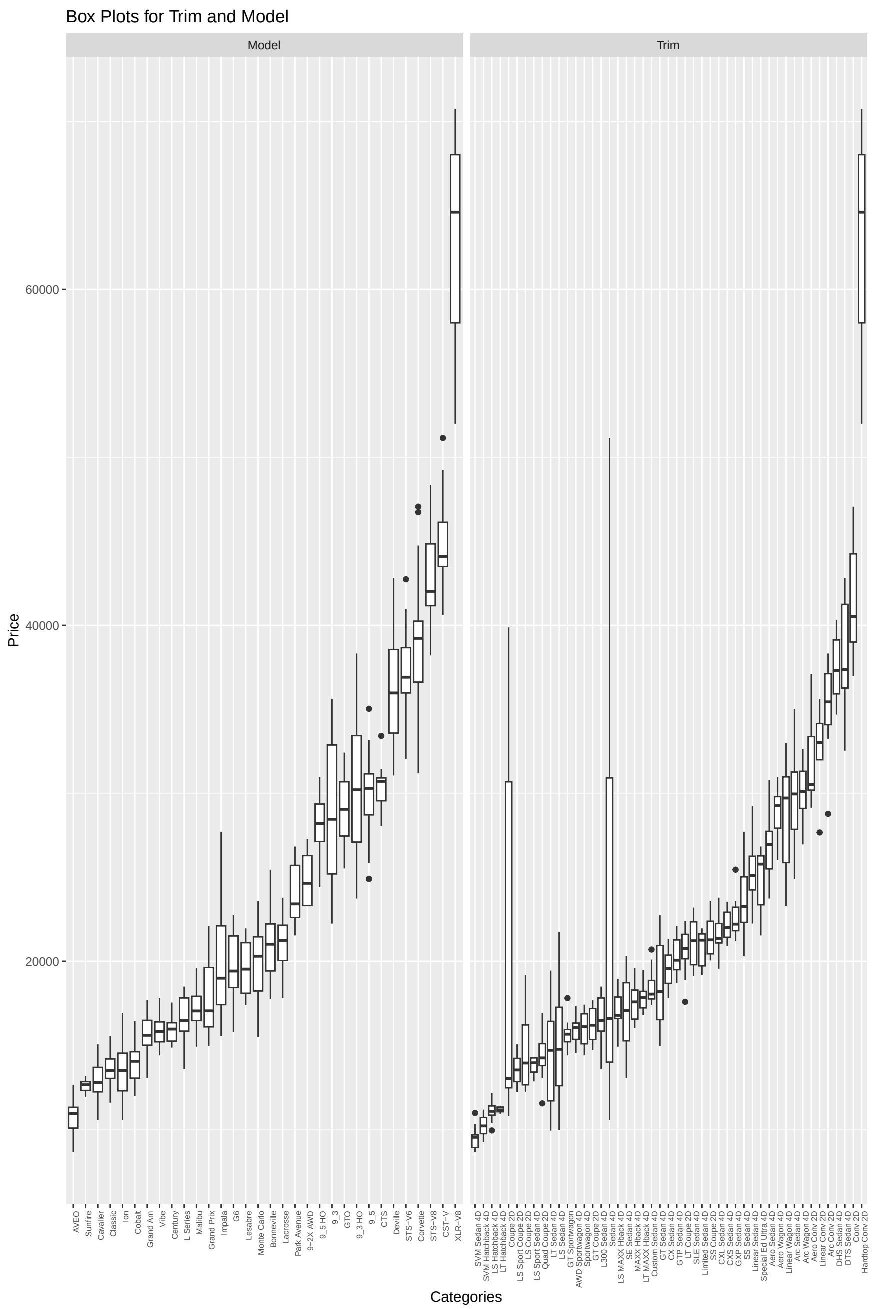
Categorical Variables Box Plots




```
car_prices %>%
  pivot_longer(
    cols = c(Trim, Model),
    names_to = "name",
    values_to = "value",
    values_transform = list(value = as.factor)
  ) %>%
  ggplot() +
  geom_boxplot(aes(x = reorder(value, Price, FUN=median), y = Price)) +
```

```
  facet_wrap(~name, scales = "free_x") +
  labs(title = "Box Plots for Trim and Model",
       x = "Categories",
       y = "Price") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 6))
```

Box Plots for Trim and Model vs. Price



```r
cars_factor_df <- car_prices %>%
  mutate(Cylinder = as.factor(Cylinder))
```

```r
mixed_model <- lm(Price ~ Mileage + Liter + Cylinder + Make + Type, data = cars_factor_df)
mixed_model %>%
  tidy()
```

Predicitive Modeling

```
## # A tibble: 14 x 5
##    term            estimate std.error statistic   p.value
##    <chr>              <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     18850.    892.          21.1  7.33e- 79
##  2 Mileage            -0.186   0.0106     -17.5  3.70e- 58
##  3 Liter            5697.    343.          16.6  1.94e- 53
##  4 Cylinder6       -3313.    620.          -5.34 1.20e-  7
##  5 Cylinder8       -3673.   1246.          -2.95 3.30e-  3
##  6 MakeCadillac    14504.    518.          28.0  2.19e-120
##  7 MakeChevrolet   -2271.    356.          -6.38 3.03e- 10
##  8 MakePontiac     -2355.    364.          -6.47 1.69e- 10
##  9 MakeSAAB         9905.    450.          22.0  4.73e- 84
## 10 MakeSaturn      -2090.    471.          -4.44 1.03e-  5
## 11 TypeCoupe      -11639.    465.         -25.0  2.34e-102
## 12 TypeHatchback -11726.    545.         -21.5  4.35e- 81
## 13 TypeSedan     -11786.    411.         -28.7  1.81e-124
## 14 TypeWagon      -8157.    501.         -16.3  1.14e- 51
```

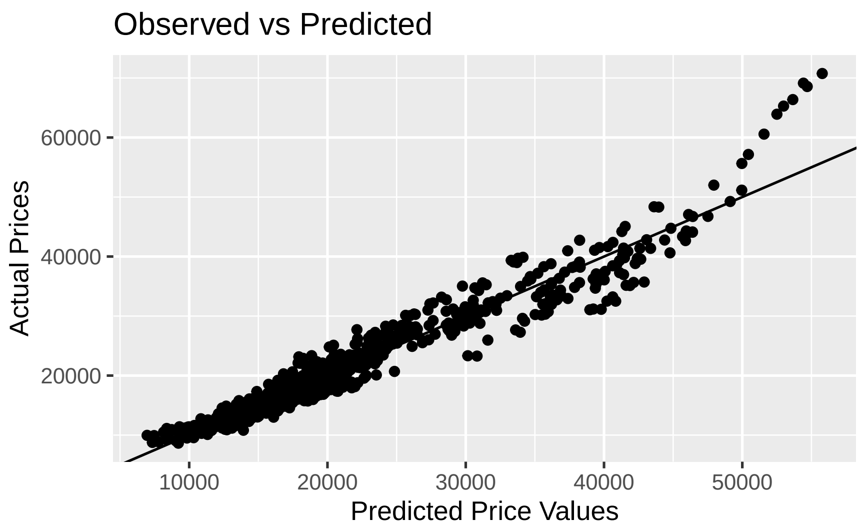
```r
glance(mixed_model) %>%
  select(r.squared)
```

```
## # A tibble: 1 x 1
##   r.squared
##       <dbl>
## 1     0.939
```

```r
mixed_df <- augment(mixed_model, cars_factor_df)
```

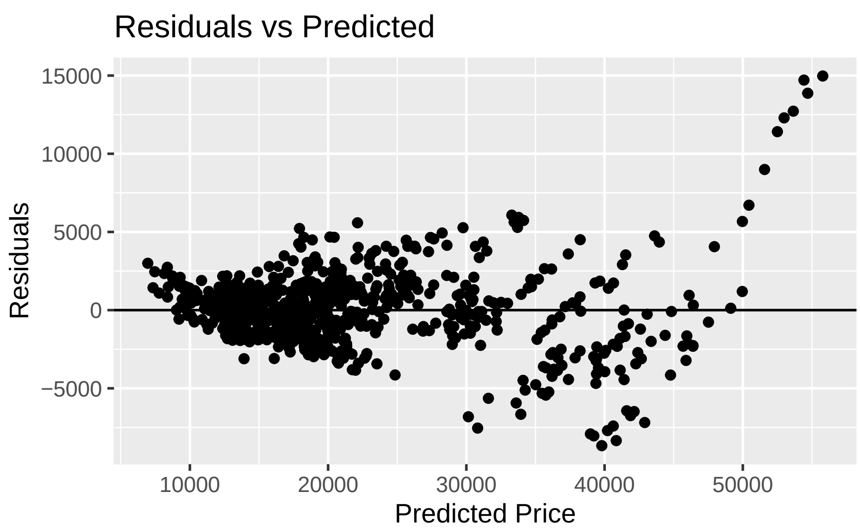
```r
mixed_df %>%
  ggplot() +
  geom_point(mapping = aes(x = .fitted, y = Price)) +
  geom_abline(slope = 1, intercept = 0) +
  labs(title = "Observed vs Predicted",
       x = "Predicted Price Values",
       y = "Actual Prices")
```

Observed vs. Predicted prices



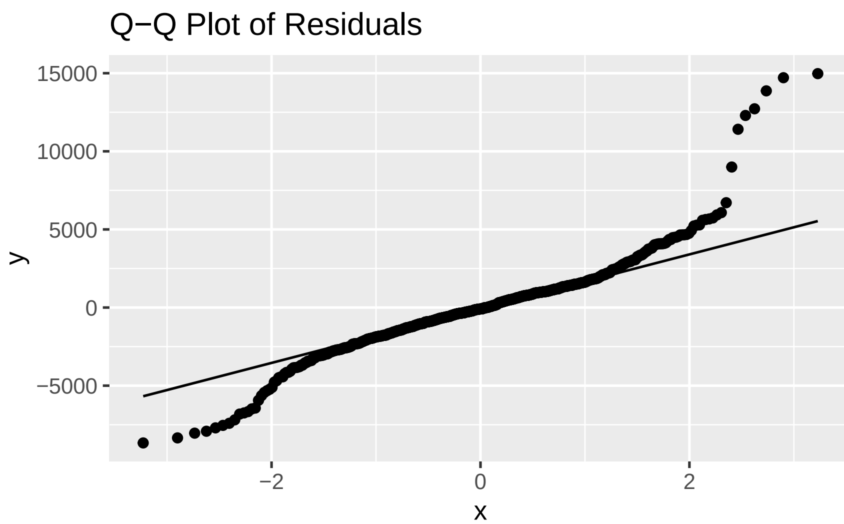

```
mixed_df %>%
  ggplot() +
  geom_point(mapping = aes(x = .fitted, y = .resid)) +
  geom_hline(yintercept = 0) +
  labs(title = "Residuals vs Predicted",
       x = "Predicted Price",
       y = "Residuals")
```

Residuals vs. Predicted prices




```
mixed_df %>%
  ggplot() +
```

```
geom_qq(aes(sample = .resid)) +
geom_qq_line(aes(sample = .resid)) +
labs(title = "Q-Q Plot of Residuals")
```



Conclusion Overall, the mixed categories model is a major improvement over the simpler model, as it better fits linearity, and constant variability (as seen in the R^2 values). Although, mixed_df model still has some minor issues with non-normality in the residuals, particularly at high prices. I would use mixed_df model for myself if I were picking a car because it's more reliable than the continuous_df model, because of R - squared values 94% vs. 33% which gives a more accurate prediction of car prices.