Data cleaned:

```r
library(dplyr)
library(ggplot2)

# Remove rows with a year greater than 2024
ford <- ford %>%
  filter(year <= 2024)
```

EDA:

```r
# Histogram of Prices
ggplot(ford, aes(x = price)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  ggtitle("Distribution of Prices") +
  xlab("Price") +
  ylab("Frequency")

# Histogram of Mileage
ggplot(ford, aes(x = mileage)) +
  geom_histogram(bins = 30, fill = "green", color = "black") +
  ggtitle("Distribution of Mileage") +
  xlab("Mileage") +
  ylab("Frequency")

# Histogram of Vehicle Age
ford <- ford %>% mutate(age = 2024 - year)  # Calculating age of the vehicles
ggplot(ford, aes(x = age)) +
  geom_histogram(bins = 30, fill = "red", color = "black") +
  ggtitle("Distribution of Vehicle Age") +
  xlab("Age (years)") +
  ylab("Frequency")

# Scatter Plot: Price vs. Mileage
ggplot(ford, aes(x = mileage, y = price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  ggtitle("Price vs. Mileage") +
  xlab("Mileage") +
  ylab("Price")

# Scatter Plot: Price vs. Age
ggplot(ford, aes(x = age, y = price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  ggtitle("Price vs. Age") +
```

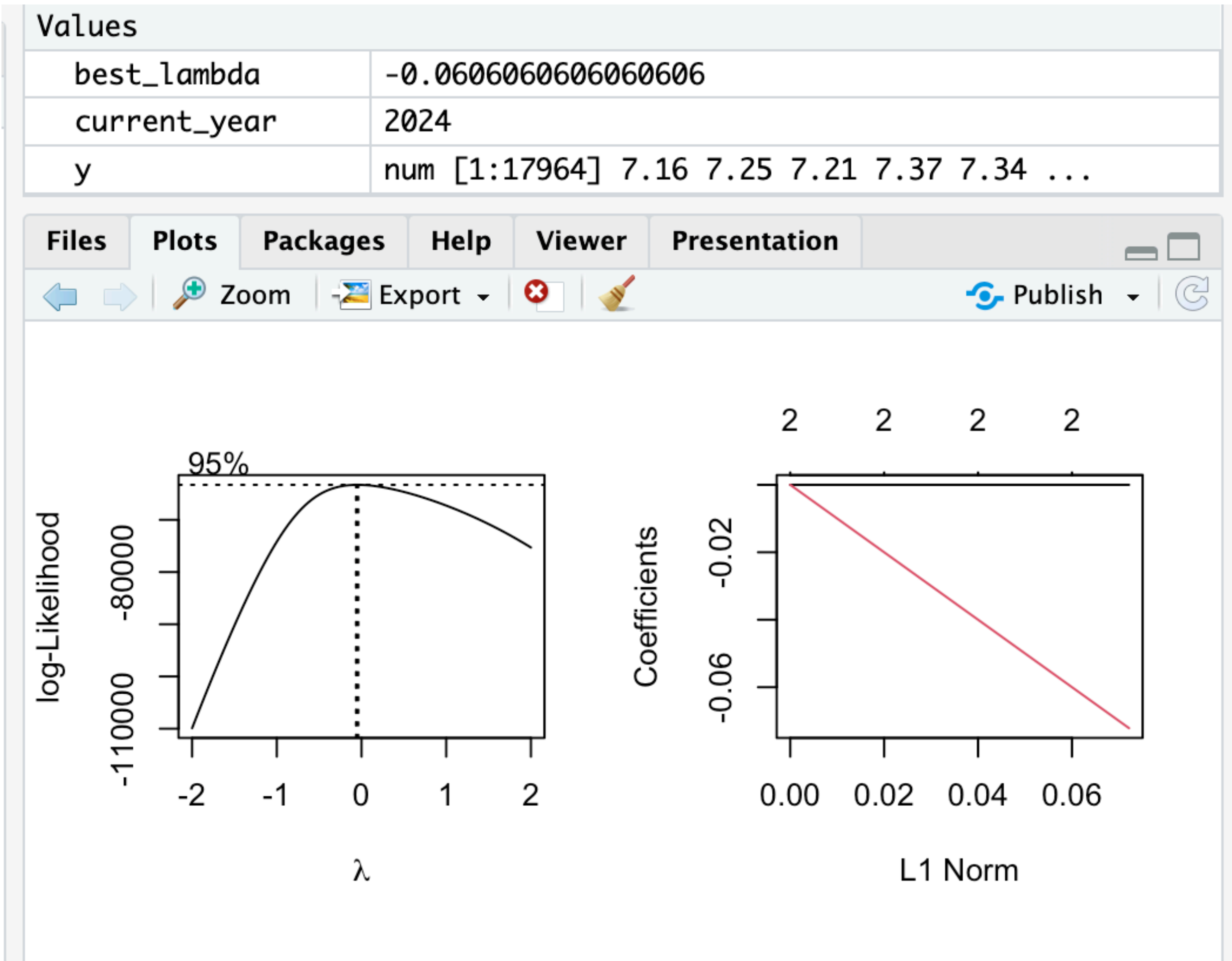
```
  xlab("Age (years)") +
  ylab("Price")
```

Add age column:

```
current_year <- 2024
ford_clean$age <- current_year - ford_clean$year

library(MASS)
# Determine best lambda for Box-Cox Transformation
bc <- boxcox(price ~ mileage + age, data = ford_clean, lambda = seq(-2, 2, by = 0.1))
best_lambda <- bc$x[which.max(bc$y)]
# Transform the price
ford_clean$price_transformed <- (ford_clean$price^best_lambda - 1) / best_lambda

library(glmnet)
> x <- model.matrix(~ mileage + age, data = ford_clean)
> y <- ford_clean$price_transformed
>
> # Fit Ridge regression model
> ridge_model <- glmnet(x, y, alpha = 0, lambda = 10^seq(4, -2, length = 100))
> plot(ridge_model)
```



WLS

```
weights <- 1 / ford_clean$age
> wls_model <- lm(price_transformed ~ mileage + age, data = ford_clean, weights = weights)
> summary(wls_model)
```

```
Call:
lm(formula = price_transformed ~ mileage + age, data = ford_clean,
    weights = weights)

Weighted Residuals:
      Min        1Q    Median        3Q       Max
-0.233417 -0.035271 -0.005627  0.031583  0.304440

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.728e+00  4.847e-03 1594.39   <2e-16 ***
mileage     -1.685e-06  8.645e-08  -19.50   <2e-16 ***
age         -7.847e-02  8.681e-04  -90.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05466 on 17961 degrees of freedom
Multiple R-squared:  0.5539,	Adjusted R-squared:  0.5538
F-statistic: 1.115e+04 on 2 and 17961 DF,  p-value: < 2.2e-16
```

```
>
> # Fit a robust regression model using rlm
> robust_model <- rlm(price ~ mileage + age, data = ford_clean)
> summary(robust_model)

Call: rlm(formula = price ~ mileage + age, data = ford_clean)
Residuals:
    Min      1Q  Median      3Q     Max
-7605.0 -2030.6  -476.8  2213.2 41302.5

Coefficients:
            Value      Std. Error t value
(Intercept) 21498.2916    97.3243   220.8933
mileage        -0.0283     0.0017   -16.2577
age         -1249.1449    16.7198   -74.7103

Residual standard error: 3088 on 17961 degrees of freedom
>
>
```

Fit a robust regression model
> robust_model <- rlm(price ~ mileage + age, data = ford_clean)
> summary(robust_model)

Hypothesis testing:
Finding best lambda for ridge:

library(glmnet)
cv_ridge <- cv.glmnet(x, y, alpha = 0)
best_lambda_ridge <- cv_ridge$lambda.min
ridge_model <- glmnet(x, y, alpha = 0, lambda = best_lambda_ridge)

#Ridge regression results
coef(ridge_model, s = "0.018090048613303")

```
>
> # Summarize the Ridge regression results
> coef(ridge_model, s = "0.018090048613303")  # Choose lambda that minimizes error
4 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept)  7.671154e+00
(Intercept)  .
mileage     -2.207898e-06
age         -6.872264e-02
>
```

# Summary of WLS model
summary(wls_model)

```
> # Summary of WLS model
> summary(wls_model)

Call:
lm(formula = price_transformed ~ mileage + age, data = ford_clean,
    weights = weights)

Weighted Residuals:
      Min        1Q    Median        3Q       Max
-0.233417 -0.035271 -0.005627  0.031583  0.304440

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.728e+00  4.847e-03 1594.39   <2e-16 ***
mileage     -1.685e-06  8.645e-08  -19.50   <2e-16 ***
age         -7.847e-02  8.681e-04  -90.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05466 on 17961 degrees of freedom
Multiple R-squared:  0.5539,    Adjusted R-squared:  0.5538
F-statistic: 1.115e+04 on 2 and 17961 DF,  p-value: < 2.2e-16


>
```

Summary of robust model

summary(robust_model)

```
>
> # Summary of robust model
> summary(robust_model)

Call: rlm(formula = price ~ mileage + age, data = ford_clean)
Residuals:
    Min      1Q  Median      3Q     Max
-7605.0 -2030.6  -476.8  2213.2 41302.5

Coefficients:
            Value      Std. Error t value
(Intercept) 21498.2916    97.3243   220.8933
mileage        -0.0283     0.0017   -16.2577
age         -1249.1449    16.7198   -74.7103

Residual standard error: 3088 on 17961 degrees of freedom
>
>
```

Breusch-Pagan test for heteroscedasticity
> bptest(wls_model)

```
	studentized Breusch-Pagan test

data:  wls_model
BP = 505.81, df = 2, p-value < 2.2e-16
```

> # Shapiro-Wilk test for normality of residuals
> shapiro.test(residuals(wls_model))

> shapiro.test(sample(residuals(wls_model),5000))

```
	Shapiro-Wilk normality test

data:  sample(residuals(wls_model), 5000)
W = 0.98665, p-value < 2.2e-16
```

```
# Diagnostic plots
> par(mfrow = c(2, 2))
> plot(wls_model)
```

10 fold cross validation

```
train.control <- trainControl(method="cv", number = 10)
> cv.model <- train(price ~ age + mileage, data = ford_clean,
+                 method = "lm",
+                 trControl = train.control)
> print(cv.model)
```

```
Linear Regression

17964 samples
    2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 16168, 16168, 16167, 16167, 16168, 16166, ...
Resampling results:

  RMSE      Rsquared   MAE
  3591.364  0.4266639  2681.428
```

Cross validation on box cox transformed price

```
> train.control <- trainControl(method="cv", number = 10)
> cv.model <- train(price_transformed ~ age + mileage, data = ford_clean,
+                 method = "lm",
+                 trControl = train.control)
> print(cv.model)
```

```
> print(cv.model)
Linear Regression

17964 samples
    2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 16166, 16169, 16168, 16167, 16167, 16167, ...
Resampling results:

  RMSE       Rsquared  MAE
  0.1435874  0.617659  0.1109057
```