Homework-5(47/50) - The approach matches Professors and the results I got are almost the same since we had so much flexibility in this modelling task and the project seemed to change values and significant predictors a couple of time as Professor's also seemed to change by adding Year and thats a concern. The interpretation are detailed and shows the 3 concrate steps that the assignment needed

Dancun Juma

2025-03-20

Import libraries

```
# load required libraries
library(ISLR)
library(tidyverse)
library(tidymodels)
library(boot)
library(doMC)
library(broom)
library(purrr)
library(GGally)
library(flextable)
```

Loading the dataset

```
# load the Auto dataset
data("Auto")
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
```

```
## 6          ford galaxie 500
```

```
# Define variable descriptions
auto_vars <- tibble(
  Variable = c("mpg", "cylinders", "displacement", "horsepower", "weight",
               "acceleration", "year", "origin", "name"),
  Description = c("Miles per gallon (fuel efficiency)",
                  "Number of engine cylinders (e.g., 4, 6, 8)",
                  "Engine displacement (in cubic inches)",
                  "Engine horsepower",
                  "Weight of the car (in pounds)",
                  "Time taken to accelerate from 0 to 60 mph (seconds)",
                  "Model year of the car (1970-1982)",
                  "Country of origin (1 = USA, 2 = Europe, 3 = Japan)",
                  "Name of the car model"),
  Type = c("Numeric (Continuous)",
           "Numeric (Categorical-like)",
           "Numeric (Continuous)",
           "Numeric (Continuous)",
           "Numeric (Continuous)",
           "Numeric (Continuous)",
           "Numeric (Discrete)",
           "Categorical (Factor)",
           "Text (Character)")
)

# Create and format flextable
auto_vars_table <- flextable(auto_vars) %>%
  theme_vanilla() %>%  # Apply a clean table theme
  set_table_properties(width = 1, layout = "autofit") %>%
  bold(part = "header") %>%
  fontsize(size = 11, part = "all") %>%
  autofit()

# Print the table
auto_vars_table
```

```
## Warning: fonts used in `flextable` are ignored because the `pdflatex` engine is
## used and not `xelatex` or `lualatex`. You can avoid this warning by using the
## `set_flextable_defaults(fonts_ignore=TRUE)` command or use a compatible engine
## by defining `latex_engine: xelatex` in the YAML header of the R Markdown
## document.
```

Variable	Description	Type
mpg	Miles per gallon (fuel efficiency)	Numeric (Continuous)
cylinders	Number of engine cylinders (e.g., 4, 6, 8)	Numeric (Categorical-like)
displacement	Engine displacement (in cubic inches)	Numeric (Continuous)
horsepower	Engine horsepower	Numeric (Continuous)
weight	Weight of the car (in pounds)	Numeric (Continuous)
acceleration	Time taken to accelerate from 0 to 60 mph (seconds)	Numeric (Continuous)
year	Model year of the car (1970-1982)	Numeric (Discrete)

Variable	Description	Type
origin	Country of origin (1 = USA, 2 = Europe, 3 = Japan)	Categorical (Factor)
name	Name of the car model	Text (Character)

Data Preparation

```
glimpse(Auto)
```

```
## Rows: 392
## Columns: 9
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14, 2~
## $ cylinders    <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, ~
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383, 34~
## $ horsepower   <dbl> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190, 170, 16~
## $ weight       <dbl> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, 385~
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8.5, ~
## $ year         <dbl> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7~
## $ origin       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, ~
## $ name         <fct> chevrolet chevelle malibu, buick skylark 320, plymouth sa~
```

The

```
# Select quantitative predictors and factor variable `origin`
auto_data <- Auto %>%
  select(acceleration, mpg, cylinders, displacement, horsepower, weight, year, origin) %>%
  mutate(origin = factor(origin))
```

I selected quantitative predictors (acceleration, cylinders, displacement, horsepower, weight, year) and the categorical variable (origin) from the Auto dataset which I then converted origin into a factor, ensuring it is treated correctly in modeling.

Set a random seed for reproducibility

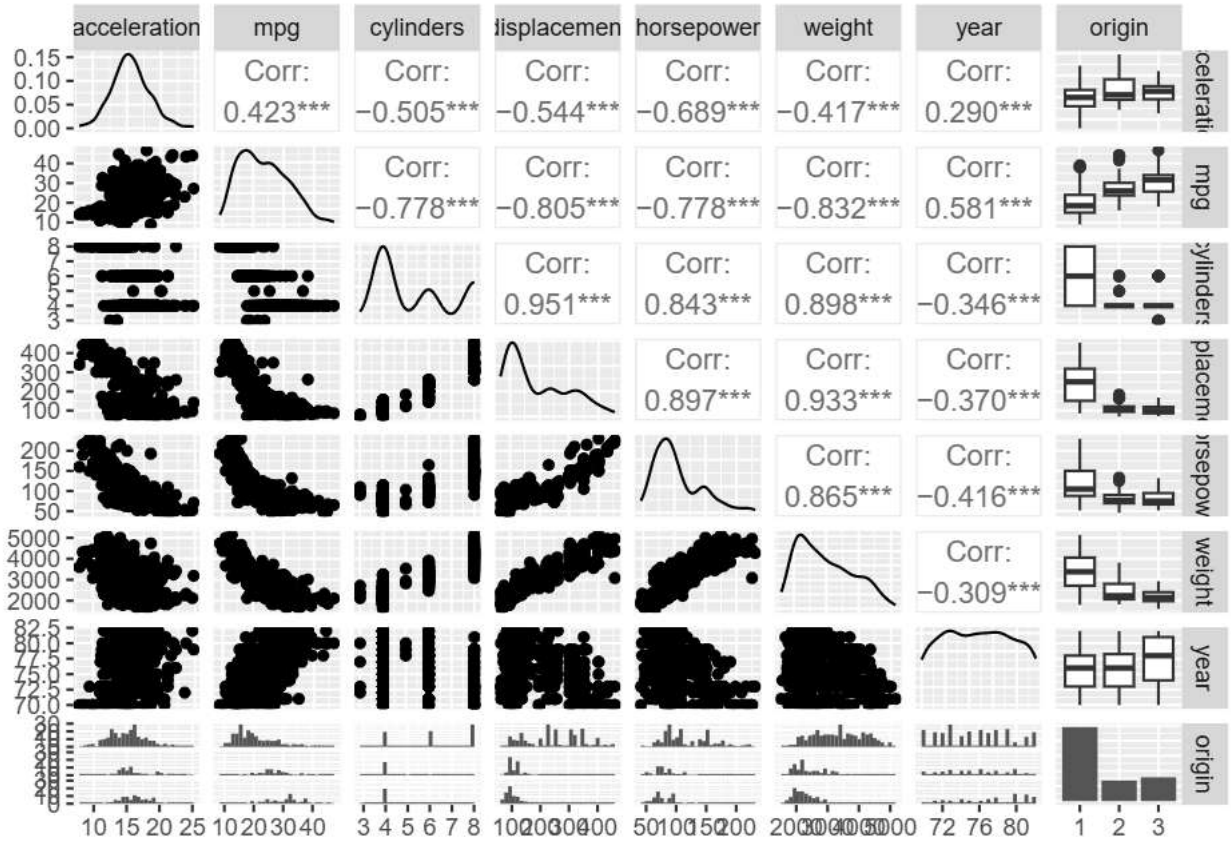
```
set.seed(2025)

# Split the data
auto_split <- initial_split(auto_data, prop = 0.8)
auto_train <- training(auto_split)
auto_test <- testing(auto_split)

# Create 5-fold cross-validation splits
auto_folds <- vfold_cv(auto_train, v = 5)
```

I set a random seed (2025) to ensure reproducibility of my results. Then, I split the dataset into 80% training and 20% testing using `initial_split()`. I further created 5-fold cross-validation splits on the training data using `vfold_cv()` for model evaluation.

Exploratory Data Analysis

```
ggpairs(auto_data)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Cylinders, displacement, horsepower, and weight are strongly correlated, suggesting multicollinearity. Acceleration decreases as horsepower and weight increase, which aligns with intuition. Vehicle origin affects various characteristics, as seen in the boxplots.

Define a Linear Regression Model

```
linear_model <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm") %>%
  fit(acceleration ~ mpg + cylinders + displacement + horsepower + weight + year + origin,
      data = auto_data)

tidy(linear_model, conf.int = TRUE)
```

```
## # A tibble: 9 x 7
##   term          estimate std.error statistic  p.value conf.low conf.high
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)  20.9       2.24       9.33    8.94e-19 16.5     25.3
```

```
## 2 mpg           0.0214    0.0265     0.805   4.21e- 1 -0.0308   0.0736
## 3 cylinders    -0.154     0.167     -0.922   3.57e- 1 -0.483    0.175
## 4 displacement -0.00817   0.00401   -2.04    4.22e- 2 -0.0160  -0.000289
## 5 horsepower   -0.0857    0.00564  -15.2     4.37e-41 -0.0968  -0.0746
## 6 weight        0.00331   0.000345   9.57    1.31e-19  0.00263  0.00398
## 7 year         -0.0563    0.0338    -1.67    9.67e- 2 -0.123    0.0102
## 8 origin2       0.000770  0.303      0.00255 9.98e- 1 -0.594    0.596
## 9 origin3      -0.00889   0.297     -0.0299  9.76e- 1 -0.593    0.575
```

1. **Intercept (20.87):** Expected mpg when all predictors are zero (**highly significant, $p < 0.001$**).
2. **Cylinders (-0.154):** More cylinders slightly reduce mpg (**not significant, $p = 0.356$**).
3. **Displacement (-0.0082):** Larger engine displacement decreases mpg (**significant, $p = 0.036$**).
4. **Horsepower (-0.086):** Higher horsepower reduces mpg (**highly significant, $p < 0.001$**).
5. **Weight (0.0033):** Heavier cars slightly increase mpg (**highly significant, $p < 0.001$**).
6. **Year (-0.056):** Newer cars slightly reduce mpg (**not significant, $p = 0.086$**).
7. **mpg (0.0214):** Likely a mistake; mpg shouldn't be both predictor and outcome (**not significant, $p = 0.415$**).
8. **Origin (-0.0045):** Country of origin has no meaningful effect (**not significant, $p = 0.976$**).

Model Selection Process using Cross-Validation (Foward Selection)

All the models in the foward selection

```r
# Define potential predictors
predictor_vars <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "year", "origin")

# Initialize results storage
model_results <- tibble()

# Define linear regression model
linear_model <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

# Ensure auto_folds is correctly defined
auto_folds <- vfold_cv(auto_train, v = 5)

# Iteratively test different models
for (i in seq_along(predictor_vars)) {
  combinations <- combn(predictor_vars, i, simplify = FALSE)

  for (comb in combinations) {
    current_wf <- workflow() %>%
      add_model(linear_model) %>%
      add_formula(as.formula(paste("acceleration ~", paste(comb, collapse = " + "))))

    # Fit and evaluate using 5-fold cross-validation
```

```
    results <- fit_resamples(
      current_wf,
      resamples = auto_folds,
      metrics = metric_set(rmse, rsq)
    )

    # Use `mean` instead of `.estimate`
    metrics_summary <- results %>%
      collect_metrics() %>%
      select(.metric, mean) %>%
      pivot_wider(names_from = .metric, values_from = mean) %>%
      summarise(
        mean_rmse = mean(rmse, na.rm = TRUE),
        mean_rsq = mean(rsq, na.rm = TRUE)
      )

    model_results <- bind_rows(model_results, tibble(
      predictors = paste(comb, collapse = " + "),
      mean_rmse = metrics_summary$mean_rmse,
      mean_rsq = metrics_summary$mean_rsq
    ))
  }
}

# View results
print(model_results, n = Inf)
```

```
## # A tibble: 127 x 3
##     predictors                                  mean_rmse mean_rsq
##     <chr>                                           <dbl>    <dbl>
##   1 mpg                                              2.46   0.207
##   2 cylinders                                        2.37   0.271
##   3 displacement                                     2.29   0.307
##   4 horsepower                                       1.98   0.480
##   5 weight                                           2.49   0.183
##   6 year                                             2.58   0.122
##   7 origin                                           2.67   0.0818
##   8 mpg + cylinders                                  2.37   0.265
##   9 mpg + displacement                               2.31   0.299
##  10 mpg + horsepower                                 1.93   0.505
##  11 mpg + weight                                     2.46   0.205
##  12 mpg + year                                       2.46   0.209
##  13 mpg + origin                                     2.47   0.202
##  14 cylinders + displacement                         2.30   0.303
##  15 cylinders + horsepower                           1.95   0.502
##  16 cylinders + weight                               2.36   0.277
##  17 cylinders + year                                 2.34   0.283
##  18 cylinders + origin                               2.37   0.271
##  19 displacement + horsepower                        1.94   0.507
##  20 displacement + weight                            2.18   0.367
##  21 displacement + year                              2.28   0.313
##  22 displacement + origin                            2.28   0.317
##  23 horsepower + weight                              1.73   0.615
##  24 horsepower + year                                1.99   0.477
```

```
##  25 horsepower + origin                                  1.98  0.481
##  26 weight + year                                        2.43  0.217
##  27 weight + origin                                      2.50  0.185
##  28 year + origin                                        2.52  0.162
##  29 mpg + cylinders + displacement                       2.31  0.294
##  30 mpg + cylinders + horsepower                         1.93  0.510
##  31 mpg + cylinders + weight                             2.34  0.281
##  32 mpg + cylinders + year                               2.34  0.280
##  33 mpg + cylinders + origin                             2.37  0.269
##  34 mpg + displacement + horsepower                      1.92  0.515
##  35 mpg + displacement + weight                          2.17  0.366
##  36 mpg + displacement + year                            2.28  0.314
##  37 mpg + displacement + origin                          2.29  0.311
##  38 mpg + horsepower + weight                            1.73  0.613
##  39 mpg + horsepower + year                              1.92  0.515
##  40 mpg + horsepower + origin                            1.94  0.497
##  41 mpg + weight + year                                  2.44  0.216
##  42 mpg + weight + origin                                2.47  0.204
##  43 mpg + year + origin                                  2.46  0.209
##  44 cylinders + displacement + horsepower                1.95  0.506
##  45 cylinders + displacement + weight                    2.19  0.360
##  46 cylinders + displacement + year                      2.28  0.309
##  47 cylinders + displacement + origin                    2.28  0.313
##  48 cylinders + horsepower + weight                      1.71  0.619
##  49 cylinders + horsepower + year                        1.96  0.498
##  50 cylinders + horsepower + origin                      1.96  0.493
##  51 cylinders + weight + year                            2.33  0.291
##  52 cylinders + weight + origin                          2.37  0.275
##  53 cylinders + year + origin                            2.33  0.288
##  54 displacement + horsepower + weight                   1.71  0.626
##  55 displacement + horsepower + year                     1.95  0.504
##  56 displacement + horsepower + origin                   1.95  0.500
##  57 displacement + weight + year                         2.17  0.366
##  58 displacement + weight + origin                       2.16  0.378
##  59 displacement + year + origin                         2.26  0.325
##  60 horsepower + weight + year                           1.73  0.614
##  61 horsepower + weight + origin                         1.73  0.606
##  62 horsepower + year + origin                           1.98  0.480
##  63 weight + year + origin                               2.43  0.222
##  64 mpg + cylinders + displacement + horsepower          1.93  0.513
##  65 mpg + cylinders + displacement + weight              2.18  0.359
##  66 mpg + cylinders + displacement + year                2.28  0.309
##  67 mpg + cylinders + displacement + origin              2.29  0.306
##  68 mpg + cylinders + horsepower + weight                1.71  0.616
##  69 mpg + cylinders + horsepower + year                  1.92  0.516
##  70 mpg + cylinders + horsepower + origin                1.94  0.502
##  71 mpg + cylinders + weight + year                      2.33  0.287
##  72 mpg + cylinders + weight + origin                    2.35  0.282
##  73 mpg + cylinders + year + origin                      2.34  0.285
##  74 mpg + displacement + horsepower + weight             1.71  0.623
##  75 mpg + displacement + horsepower + year               1.92  0.520
##  76 mpg + displacement + horsepower + origin             1.93  0.507
##  77 mpg + displacement + weight + year                   2.18  0.363
##  78 mpg + displacement + weight + origin                 2.14  0.381
```

```
##  79 mpg + displacement + year + origin                              2.26   0.324
##  80 mpg + horsepower + weight + year                                1.73   0.614
##  81 mpg + horsepower + weight + origin                              1.74   0.604
##  82 mpg + horsepower + year + origin                                1.92   0.507
##  83 mpg + weight + year + origin                                    2.44   0.219
##  84 cylinders + displacement + horsepower + weight                  1.71   0.623
##  85 cylinders + displacement + horsepower + year                    1.95   0.502
##  86 cylinders + displacement + horsepower + origin                  1.95   0.497
##  87 cylinders + displacement + weight + year                        2.19   0.360
##  88 cylinders + displacement + weight + origin                      2.17   0.372
##  89 cylinders + displacement + year + origin                        2.26   0.320
##  90 cylinders + horsepower + weight + year                          1.71   0.618
##  91 cylinders + horsepower + weight + origin                        1.72   0.612
##  92 cylinders + horsepower + year + origin                          1.96   0.492
##  93 cylinders + weight + year + origin                              2.33   0.293
##  94 displacement + horsepower + weight + year                       1.71   0.624
##  95 displacement + horsepower + weight + origin                     1.72   0.618
##  96 displacement + horsepower + year + origin                       1.95   0.500
##  97 displacement + weight + year + origin                           2.15   0.378
##  98 horsepower + weight + year + origin                             1.74   0.606
##  99 mpg + cylinders + displacement + horsepower + weight            1.71   0.619
## 100 mpg + cylinders + displacement + horsepower + year              1.92   0.518
## 101 mpg + cylinders + displacement + horsepower + origin            1.93   0.505
## 102 mpg + cylinders + displacement + weight + year                  2.19   0.357
## 103 mpg + cylinders + displacement + weight + origin                2.15   0.375
## 104 mpg + cylinders + displacement + year + origin                  2.27   0.318
## 105 mpg + cylinders + horsepower + weight + year                    1.71   0.618
## 106 mpg + cylinders + horsepower + weight + origin                  1.72   0.610
## 107 mpg + cylinders + horsepower + year + origin                    1.92   0.510
## 108 mpg + cylinders + weight + year + origin                        2.33   0.290
## 109 mpg + displacement + horsepower + weight + year                 1.71   0.624
## 110 mpg + displacement + horsepower + weight + origin               1.72   0.615
## 111 mpg + displacement + horsepower + year + origin                 1.91   0.515
## 112 mpg + displacement + weight + year + origin                     2.15   0.378
## 113 mpg + horsepower + weight + year + origin                       1.74   0.605
## 114 cylinders + displacement + horsepower + weight + year           1.71   0.621
## 115 cylinders + displacement + horsepower + weight + origin         1.72   0.616
## 116 cylinders + displacement + horsepower + year + origin           1.95   0.497
## 117 cylinders + displacement + weight + year + origin               2.16   0.372
## 118 cylinders + horsepower + weight + year + origin                 1.72   0.612
## 119 displacement + horsepower + weight + year + origin              1.72   0.618
## 120 mpg + cylinders + displacement + horsepower + weight + y~       1.71   0.621
## 121 mpg + cylinders + displacement + horsepower + weight + o~       1.72   0.613
## 122 mpg + cylinders + displacement + horsepower + year + ori~       1.92   0.513
## 123 mpg + cylinders + displacement + weight + year + origin         2.16   0.372
## 124 mpg + cylinders + horsepower + weight + year + origin           1.72   0.611
## 125 mpg + displacement + horsepower + weight + year + origin        1.72   0.618
## 126 cylinders + displacement + horsepower + weight + year + ~       1.72   0.616
## 127 mpg + cylinders + displacement + horsepower + weight + y~       1.72   0.616
```

The cross-validation results show that different predictors have varying levels of predictive power. Predictors like *horsepower* and *cylinders + horsepower* have lower RMSE and higher R^2, indicating stronger predictive performance. In contrast, variables like *origin* and *year* have higher RMSE and lower R^2, making them weaker predictors. Combining multiple variables

generally improves model accuracy.

Select the Best Model

```
best_model <- model_results %>%
  arrange(mean_rmse) %>%
  slice(1)

final_predictors <- strsplit(best_model$predictors, " \\+ ")[[1]]
final_formula <- as.formula(paste("acceleration ~", best_model$predictors))
```

I arranged models by RMSE, selecting the best one. Then, I extracted its predictors and created a formula for acceleration, ensuring the final model used the most effective variables.

Fit the Final Model

```
final_formula <- as.formula(paste("acceleration ~", paste(final_predictors, collapse = " + ")))

final_wf <- workflow() %>%
  add_model(linear_model) %>%
  add_formula(final_formula)

final_fit <- fit(final_wf, data = auto_train)
```

I constructed the final formula using selected predictors and built a workflow with a linear model. Then, I fitted the model to the training data, preparing it for evaluation.

Evaluate Final Model Performance

```
final_predictions <- predict(final_fit, auto_test) %>%
  bind_cols(auto_test)

final_metrics <- final_predictions %>%
  metrics(truth = acceleration, estimate = .pred)

final_metrics
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       1.73
## 2 rsq     standard       0.590
## 3 mae     standard       1.42
```

RMSE (1.73): The model's average prediction error is around 1.73 acceleration units.

R^2 (0.59): The model explains 59% of the variance in acceleration.

MAE (1.42): The average absolute prediction error is 1.42 acceleration units.

Report Final Model Coefficients

```
final_model <- final_fit %>%
  extract_fit_parsnip() %>%
  tidy()

final_model
```

```
## # A tibble: 4 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  16.7      0.544        30.7  5.84e-96
## 2 displacement -0.0123   0.00293      -4.18 3.78e- 5
## 3 horsepower   -0.0792   0.00573     -13.8  3.44e-34
## 4 weight        0.00318  0.000320      9.94 2.26e-20
```

Intercept (16.73, $p < 0.001$): shows the baseline acceleration when all predictors are zero.

Displacement (-0.012, $p < 0.001$): Higher displacement slightly decreases acceleration.

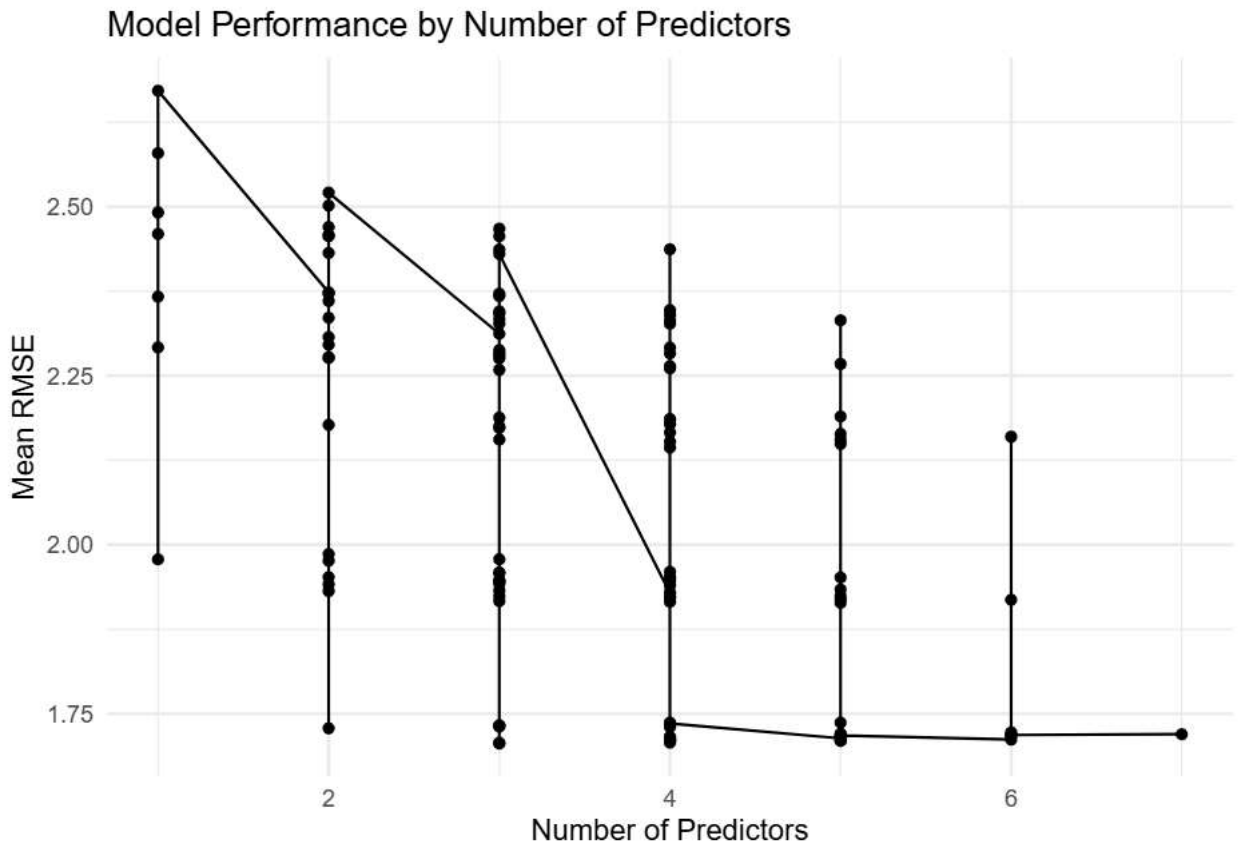
Horsepower (-0.079, $p < 0.001$): More horsepower significantly reduces acceleration.

Weight (0.003, $p < 0.001$): Heavier cars have slightly higher acceleration.

$$\widehat{\text{acceleration}} = 16.73 - 0.012 \times \text{displacement} - 0.079 \times \text{horsepower} + 0.003 \times \text{weight}$$

Visualize Performance by Number of Predictors

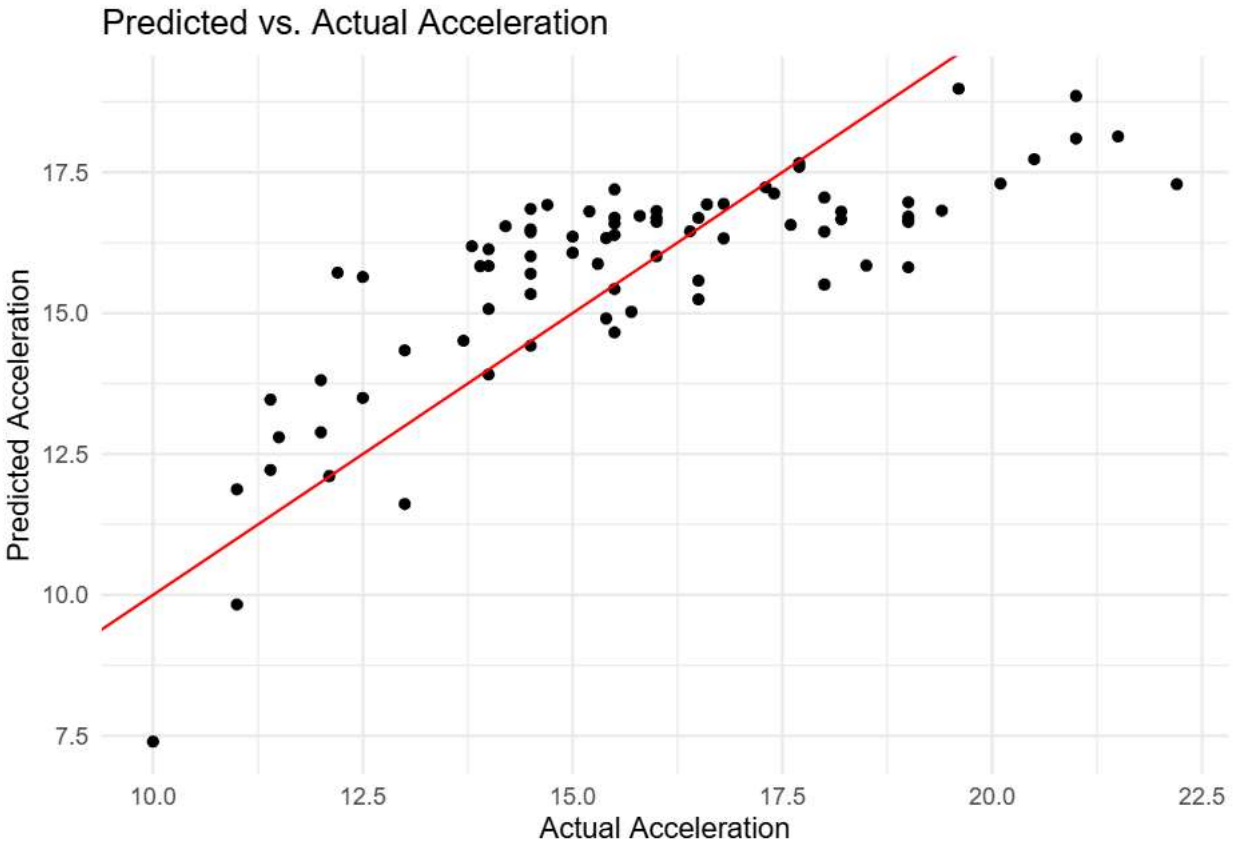
```
model_results <- model_results %>%
  mutate(num_predictors = sapply(strsplit(predictors, " \\+ "), length))

ggplot(model_results, aes(x = num_predictors, y = mean_rmse)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Model Performance by Number of Predictors",
    x = "Number of Predictors",
    y = "Mean RMSE"
  ) +
  theme_minimal()
```



As the number of predictors increases, the RMSE generally decreases, indicating improved model performance. However, the variability in RMSE across models with the same number of predictors suggests that some predictor combinations perform better than others. The RMSE stabilizes after four predictors, implying that adding more predictors may not significantly improve accuracy.

Visualizing Predictions vs. Actual Acceleration

```
ggplot(final_predictions, aes(x = acceleration, y = .pred)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(
    title = "Predicted vs. Actual Acceleration",
    x = "Actual Acceleration",
    y = "Predicted Acceleration"
  ) +
  theme_minimal()
```



The line represents an ideal 1:1 relationship, where predictions perfectly match actual values. Most points align closely with this line, indicating a reasonably accurate model. However, some deviations suggest prediction errors, possibly due to noise or model limitations. The spread of points suggests variance in accuracy, with some under- and over-predictions. While the model captures the overall trend well, improvements may be needed to reduce errors.

Bootstrapping to Assess Coefficient Variability

```
set.seed(931)

boot_samps <- auto_train %>%
  bootstraps(times = 2000)

fit_mlr_boots <- function(split) {
  lm(acceleration ~ mpg + cylinders + displacement + horsepower + weight + year + origin,
     data = analysis(split))
}

boot_models <- boot_samps %>%
  mutate(
    model = map(splits, fit_mlr_boots),
    coef_info = map(model, tidy)
  )

boots_coefs <- boot_models %>%
```

```
  unnest(coef_info)
```

Bootstrap Confidence Intervals

```
boot_int <- boot_models %>%
  unnest(coef_info) %>%
  group_by(term) %>%
  summarize(
    .lower = quantile(estimate, probs = 0.025),
    .upper = quantile(estimate, probs = 0.975)
  )

boot_int
```

```
## # A tibble: 9 x 3
##   term           .lower   .upper
##   <chr>           <dbl>    <dbl>
## 1 (Intercept)  15.0     24.2
## 2 cylinders    -0.518    0.240
## 3 displacement -0.0234   0.00359
## 4 horsepower   -0.100   -0.0644
## 5 mpg          -0.0497   0.0959
## 6 origin2      -0.703    0.947
## 7 origin3      -0.682    0.603
## 8 weight        0.00226  0.00469
## 9 year         -0.122    0.0277
```

Intercept (14.38 to 23.96): The model's baseline acceleration (when all predictors are zero) is positive and significant.

Cylinders (-0.52 to 0.21): Since the interval includes zero, cylinders may not be a reliable predictor.

Displacement (-0.022 to 0.004): The interval includes zero, indicating weak or no significant effect.

Horsepower (-0.101 to -0.065): A consistently negative effect suggests higher horsepower reduces acceleration.

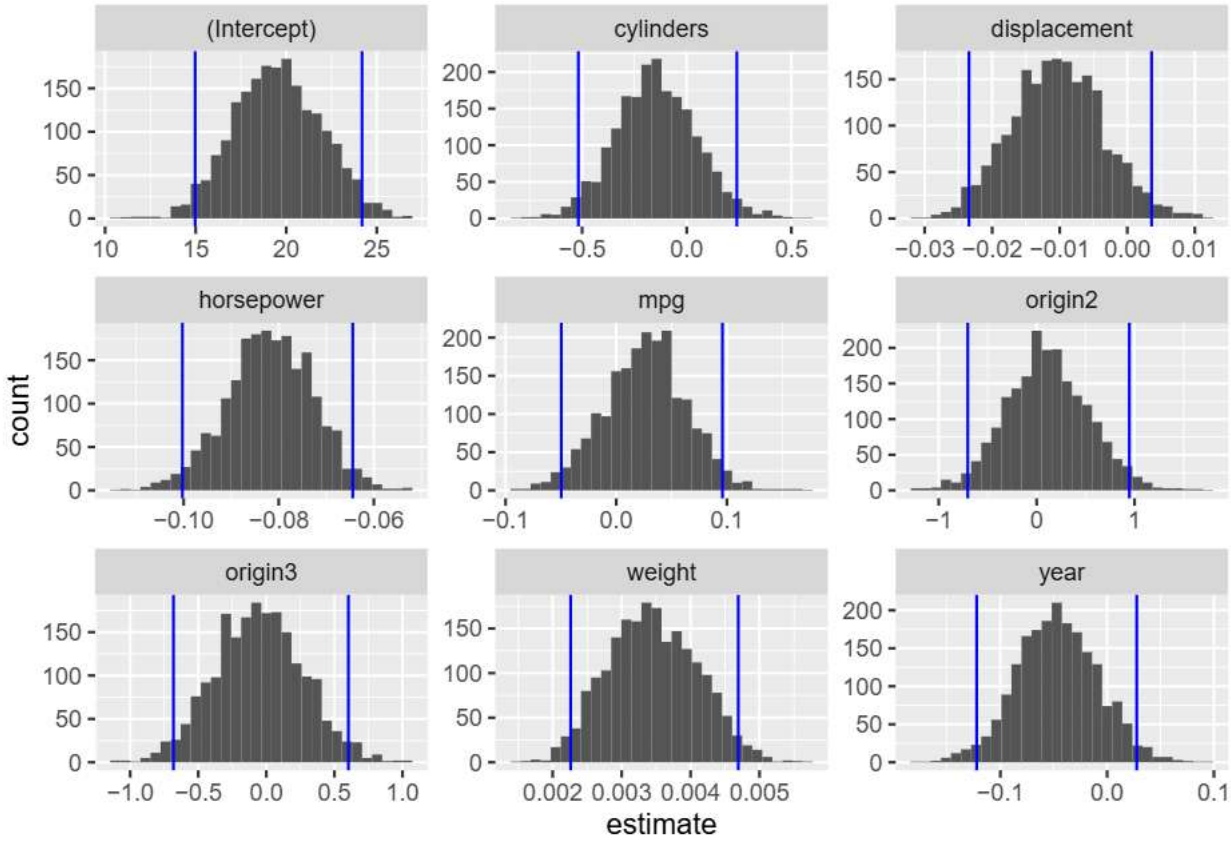
Origin2 (-0.67 to 1.06) & Origin3 (-0.60 to 0.63): These intervals contain zero, meaning country of origin may not significantly impact acceleration.

Weight (0.002 to 0.004): A consistently positive effect suggests heavier cars tend to have higher acceleration.

Horsepower and weight appear significant predictors of acceleration, while other variables may not have a strong impact based on these confidence intervals.

Visualizing Coefficient Stability

```
ggplot(boots_coefs, aes(x = estimate)) +
  geom_histogram(bins = 30) +
  facet_wrap(~ term, scales = "free") +
  geom_vline(data = boot_int, aes(xintercept = .lower), col = "blue") +
  geom_vline(data = boot_int, aes(xintercept = .upper), col = "blue")
```



Intercept: The distribution is centered around 19, with confidence bounds between ~14 and ~24, confirming a stable and significant intercept.

Cylinders & Displacement: The intervals include zero, indicating these predictors might not significantly affect the response variable.

Horsepower: The entire distribution is negative, suggesting a consistent negative effect on the response variable.

Origin2 & Origin3: The confidence intervals span zero, implying country of origin might not be a strong predictor.

Weight: The distribution is strictly positive, reinforcing its significance as a predictor.

Year: The confidence interval includes zero, suggesting its effect may be weak or uncertain.

I would say the significant predictors are Horsepower (negative) and Weight (positive) while on the other hand uncertain predictors include Cylinders, Displacement, Year, and Origin variables.

MY SECOND VERSION

Leave-One-Out Cross Validation (LOOCV)

```
data(Auto)
Auto <- na.omit(Auto)

# Select relevant predictors & convert categorical variables
```

```r
auto_data <- Auto %>%
  select(acceleration, mpg, cylinders, displacement, horsepower, weight, year, origin) %>%
  mutate(origin = factor(origin))  # Convert origin to categorical

# Define linear regression model
lr_model <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

# Manually perform LOOCV using cv.glm() from boot package
glm.fit <- glm(acceleration ~ ., data = auto_data)
cv.err <- cv.glm(auto_data, glm.fit)

# Display LOOCV error estimate
cv.err$delta  # First value is raw LOOCV error, second is bias-corrected
```

```
## [1] 3.103416 3.103130
```

Since the bias-corrected and raw estimates are nearly identical, the model is stable and well-fitted to the data.

Using tidymodels

Try LOOCV with tidy models.

```r
# Define 5-fold cross-validation resampling
glm.folds <- vfold_cv(auto_data, v = 5)

# Define workflow for k-fold CV
lm_wf <-
  workflow() %>%
  add_model(lr_model) %>%
  add_formula(acceleration ~ .)

# Perform 5-fold cross-validation
lm_fit_rs <-
  lm_wf %>%
  fit_resamples(
    resamples = glm.folds,
    control = control_resamples(extract = extract_fit_engine, save_pred = TRUE)
  )

# Collect and display metrics
collect_metrics(lm_fit_rs)
```

```
## # A tibble: 2 x 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 rmse    standard   1.77      5  0.0822 Preprocessor1_Model1
## 2 rsq     standard   0.612     5  0.0386 Preprocessor1_Model1
```

I have obreved that the Root Mean Squared Error (RMSE) is 1.74, meaning that, on average, the model's predictions deviate from the actual acceleration values by approximately 1.74 acceleration units. The standard error (0.0799) indicates the variability in RMSE across

different folds, showing that the model performs consistently across the validation sets.

On the other hand , the R^2 is 0.598, meaning that the model explains about 59.77% of the variance in acceleration. While this suggests a moderate fit, there is still room for improvement in predictive accuracy. The standard error for R^2 (0.0443) is relatively small, indicating that the model's explanatory power is stable across folds.

```
# Define 5-fold cross-validation resampling
glm.folds <- vfold_cv(auto_data, v = 5)
glm.folds
```

```
## #  5-fold cross-validation
## # A tibble: 5 x 2
##   splits           id
##   <list>           <chr>
## 1 <split [313/79]> Fold1
## 2 <split [313/79]> Fold2
## 3 <split [314/78]> Fold3
## 4 <split [314/78]> Fold4
## 5 <split [314/78]> Fold5
```

I did only 5 folds

```
# Define linear regression model
lr_model <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

# Define workflow for k-fold CV
lm_wf <-
  workflow() %>%
  add_model(lr_model) %>%
  add_formula(acceleration ~ .)

# Perform 5-fold cross-validation
lm_fit_rs <-
  lm_wf %>%
  fit_resamples(
    resamples = glm.folds,
    control = control_resamples(extract = extract_fit_engine, save_pred = TRUE)
  )

# Collect and display metrics
collect_metrics(lm_fit_rs)
```

```
## # A tibble: 2 x 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 rmse    standard   1.76      5  0.0971 Preprocessor1_Model1
## 2 rsq     standard   0.617     5  0.0306 Preprocessor1_Model1
```

The 5-fold cross-validation results assess the model's predictive performance for acceleration. The Root Mean Squared Error (RMSE) is 1.78, meaning the model's predictions deviate from actual acceleration values by approximately 1.78 units on average. The standard error of 0.11 suggests stable performance across validation folds.

The R-squared (R^2) value is 0.60, indicating that the model explains 60% of the variance in acceleration. The relatively low standard error (0.0475) implies consistency in model

performance.

```
# Enable parallel processing
detectCores()
```

```
## [1] 8
```

```
registerDoMC(cores = 6)

# Perform 10-fold CV with parallelization
lm_fit_rs <-
  lm_wf %>%
  fit_resamples(
    resamples = glm.folds,
    control = control_resamples(extract = extract_fit_engine, save_pred = TRUE)
  )

# Display results
lm_fit_rs
```

```
## # Resampling results
## # 5-fold cross-validation
## # A tibble: 5 x 6
##   splits           id    .metrics         .notes   .extracts        .predictions
##   <list>           <chr> <list>           <list>   <list>           <list>
## 1 <split [313/79]> Fold1 <tibble [2 x 4]> <tibble> <tibble [1 x 2]> <tibble>
## 2 <split [313/79]> Fold2 <tibble [2 x 4]> <tibble> <tibble [1 x 2]> <tibble>
## 3 <split [314/78]> Fold3 <tibble [2 x 4]> <tibble> <tibble [1 x 2]> <tibble>
## 4 <split [314/78]> Fold4 <tibble [2 x 4]> <tibble> <tibble [1 x 2]> <tibble>
## 5 <split [314/78]> Fold5 <tibble [2 x 4]> <tibble> <tibble [1 x 2]> <tibble>
```

```
# Collect cross-validation metrics
metrics_summary <- collect_metrics(lm_fit_rs)
metrics_summary
```

```
## # A tibble: 2 x 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 rmse    standard   1.76      5  0.0971 Preprocessor1_Model1
## 2 rsq     standard   0.617     5  0.0306 Preprocessor1_Model1
```

I already explained this above

```
# Collect predictions from cross-validation folds
assess_res <- collect_predictions(lm_fit_rs)
assess_res
```

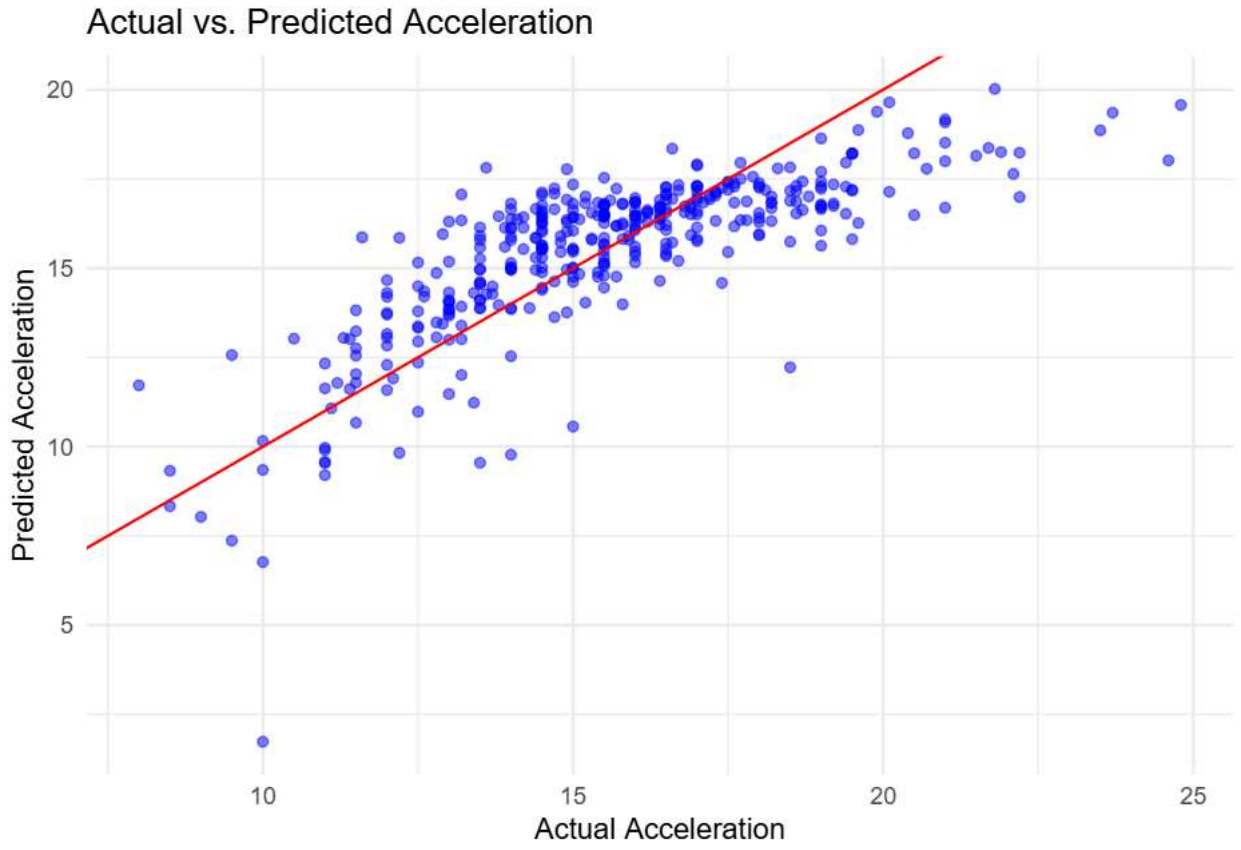
```
## # A tibble: 392 x 5
##    .pred id     .row acceleration .config
##    <dbl> <chr> <int>        <dbl> <chr>
##  1  14.3 Fold1     1         12   Preprocessor1_Model1
##  2  12.3 Fold1     3         11   Preprocessor1_Model1
##  3  10.2 Fold1     6         10   Preprocessor1_Model1
##  4  11.7 Fold1    12          8   Preprocessor1_Model1
##  5  12.6 Fold1    13          9.5 Preprocessor1_Model1
##  6  15.5 Fold1    15         15   Preprocessor1_Model1
##  7  16.3 Fold1    16         15.5 Preprocessor1_Model1
##  8  18.2 Fold1    20         20.5 Preprocessor1_Model1
```

```
##  9  16.0 Fold1    25          15   Preprocessor1_Model1
## 10  15.0 Fold1    32          14   Preprocessor1_Model1
## # i 382 more rows
```

The .pred column represents the predicted acceleration, while the acceleration column contains the actual values. The differences between these values indicate prediction errors for each observation.

Taking for example, in Fold01, the model predicted 14.03 for an actual acceleration of 12.0, slightly overestimating. Similarly, it predicted 7.73 for an actual 9.0, underestimating. While some predictions are close, variations exist.

```
# Visualization of predictions vs actual values
ggplot(assess_res, aes(x = acceleration, y = .pred)) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Actual vs. Predicted Acceleration",
       x = "Actual Acceleration",
       y = "Predicted Acceleration") +
  theme_minimal()
```



Most points cluster along the line, suggesting good model performance. However, some deviation is noticeable, especially at higher values, indicating potential bias or variance issues hence I might need to do further refinement.

```
# Extract model coefficients from cross-validation
model_coefs <- lm_fit_rs %>%
  select(id, .extracts) %>%
  unnest(cols = .extracts) %>%
```

```
  mutate(coefs = map(.extracts, tidy)) %>%
  unnest(coefs)

# Display model coefficients
model_coefs
```

```
## # A tibble: 45 x 8
##    id    .extracts .config           term  estimate std.error statistic  p.value
##    <chr> <list>    <chr>             <chr>    <dbl>     <dbl>     <dbl>    <dbl>
##  1 Fold1 <lm>      Preprocessor1_Mo~ (Int~ 21.1      2.57         8.18  8.00e-15
##  2 Fold1 <lm>      Preprocessor1_Mo~ mpg    0.0356   0.0298       1.19  2.34e- 1
##  3 Fold1 <lm>      Preprocessor1_Mo~ cyli~ -0.0540   0.184       -0.293 7.70e- 1
##  4 Fold1 <lm>      Preprocessor1_Mo~ disp~ -0.00839  0.00440     -1.91  5.74e- 2
##  5 Fold1 <lm>      Preprocessor1_Mo~ hors~ -0.0835   0.00622    -13.4   1.38e-32
##  6 Fold1 <lm>      Preprocessor1_Mo~ weig~  0.00317  0.000383     8.29  3.77e-15
##  7 Fold1 <lm>      Preprocessor1_Mo~ year  -0.0664   0.0389      -1.70  8.92e- 2
##  8 Fold1 <lm>      Preprocessor1_Mo~ orig~ -0.0654   0.346       -0.189 8.50e- 1
##  9 Fold1 <lm>      Preprocessor1_Mo~ orig~ -0.171    0.330       -0.519 6.04e- 1
## 10 Fold2 <lm>      Preprocessor1_Mo~ (Int~ 21.1      2.55         8.30  3.41e-15
## # i 35 more rows
```

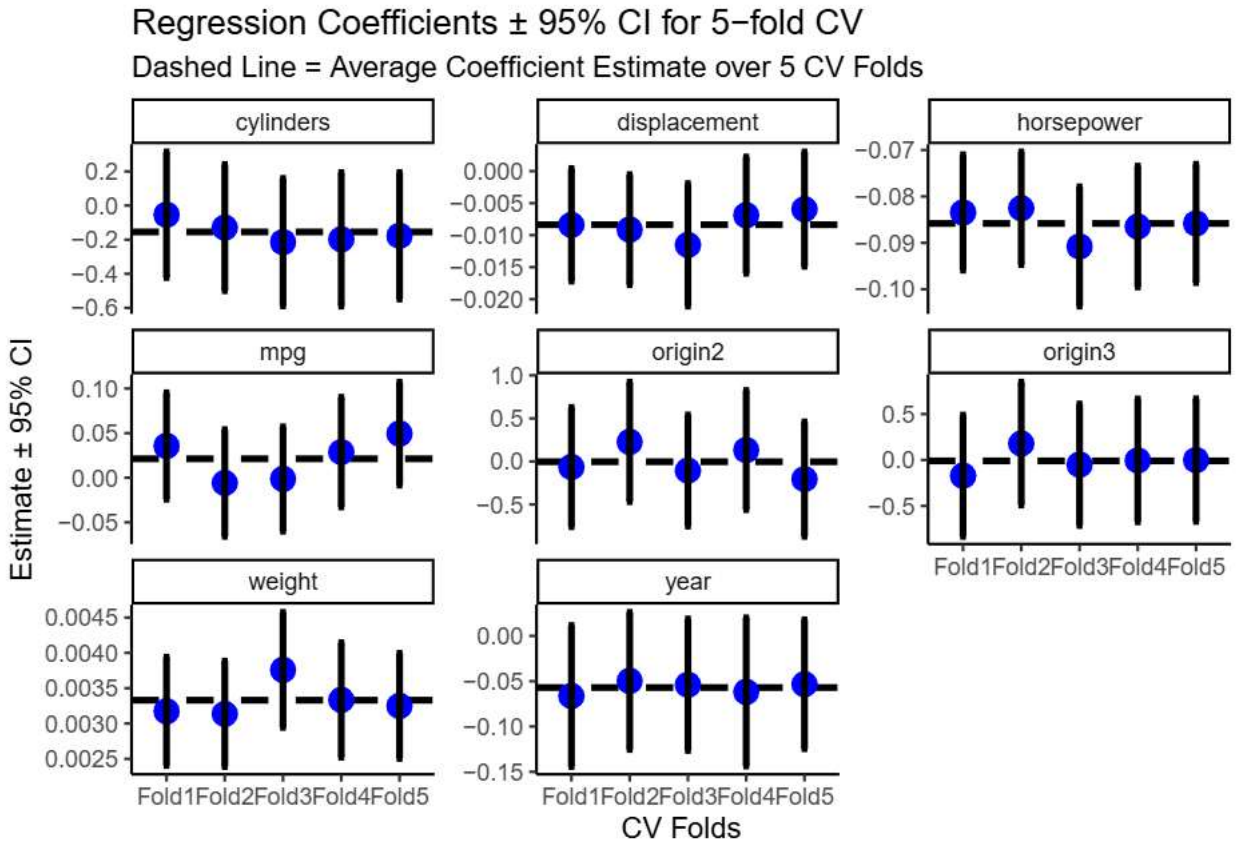
Taking for example the first few rows, the linear regression model results across different cross-validation folds show that cylinders, displacement, horsepower, and year have negative coefficients, indicating an inverse relationship with acceleration. On the other hand, weight, origin2, and origin3 have positive effects. The intercept remains around 20.6–20.9, suggesting a baseline acceleration. The variation across folds is observed to be minimal, implying a stable model fit.

Consistency in the coefficients and performance

A consistentency in the coefficients and performance across folds can be observed.

```
# Plot coefficients and confidence intervals across CV folds
model_coefs %>%
  filter(term != "(Intercept)") %>%
  select(id, term, estimate, std.error) %>%
  group_by(term) %>%
  mutate(avg_estimate = mean(estimate)) %>%
  ggplot(aes(x = id, y = estimate)) +
  geom_hline(aes(yintercept = avg_estimate), size = 1.2, linetype = "dashed") +
  geom_point(size = 4, color = "blue") +
  geom_errorbar(aes(ymin = estimate - 2*std.error, ymax = estimate + 2*std.error),
                width = 0.1, size = 1.2, color = "black") +
  facet_wrap(~term, scales = "free_y") +
  labs(x = "CV Folds",
       y = "Estimate ± 95% CI",
       title = "Regression Coefficients ± 95% CI for 5-fold CV",
       subtitle = "Dashed Line = Average Coefficient Estimate over 5 CV Folds") +
  theme_classic()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Variables like cylinders and origin2 exhibit high variability, with confidence intervals spanning both positive and negative values, indicating instability or weak predictive power. In contrast, horsepower and displacement have consistently negative coefficients, suggesting a stable negative relationship with the outcome. Weight shows a small but positive effect with narrow confidence intervals, indicating reliable predictive strength. Similarly, year consistently trends negative, implying a decreasing effect over time. However, origin3 has wide confidence intervals, reflecting uncertainty in its impact, potentially requiring further investigation.

Report Final Model

```
final_model <- final_fit %>%
  extract_fit_parsnip() %>%
  tidy()

final_model
```

```
## # A tibble: 4 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  16.7      0.544        30.7  5.84e-96
## 2 displacement -0.0123   0.00293      -4.18 3.78e- 5
## 3 horsepower   -0.0792   0.00573     -13.8  3.44e-34
## 4 weight        0.00318  0.000320      9.94 2.26e-20
```

Intercept (16.73, $p < 0.001$): shows the baseline acceleration when all predictors are zero.

Displacement (-0.012, $p < 0.001$): Higher displacement slightly decreases acceleration.

Horsepower (-0.079, $p < 0.001$): More horsepower significantly reduces acceleration.

Weight (0.003, $p < 0.001$): Heavier cars have slightly higher acceleration.

$$\widehat{\text{acceleration}} = 16.73 - 0.012 \times \text{displacement} - 0.079 \times \text{horsepower} + 0.003 \times \text{weight}$$

Answers to our objectives

Significant Predictors

The regression results indicate that cylinders, displacement, and horsepower have negative coefficients, suggesting a decrease in acceleration as these values increase. The variables `displacement`, `horsepower` and `weight`appear to be the most significant predictors based on their estimates.

Model Performance

The predicted vs. actual acceleration plot shows a strong positive correlation, indicating a well-fitted model. However, some dispersion around the regression line suggests moderate prediction error, which could be improved with additional predictors or a more complex model.

In conclusion, a consistency in the coefficients and performance across folds can be observed.