RSM8512_Assignment1_1006759189_Bias/Variance_Tradeoff

Question 1 For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer. **(a)** The sample size n is extremely large, and the number of predictors p is small.

The flexible method is better: With a an extremely sample size and few predictors, a flexible model can accurately learn complex patterns without overfitting, as there is enough data to support the model complexity.

(b) The number of predictors p is extremely large, and the number of observations n is small.

The inflexible method is better: When the number of predictors is extremely large and outnumbers observations, flexible models overfit by trying to fit the noise in the data. Inflexible models are more stable and generalize better with limited data.

(c) The relationship between the predictors and response is highly non-linear.

The flexible model is better: When the relationship between the predictors and the response is highly non-linear,flexible models are able to adapt to these complex patterns, while inflexible models cannot capture the non-linearity effectively and tend to underfit in these situations.

(d) The variance of the error terms, is extremely high.

The inflexible model is better: When there is high error variance that means there is a lot of noise, flexible models are likely to fit this noise, leading to poor performance on unseen data.On the other hand, inflexible models ignore the noise, focusing on the overall trend, improving the model's ability to generalize.

Question 2 We now revisit the bias-variance decomposition. **(a)** Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.

```r
library(ggplot2)
```

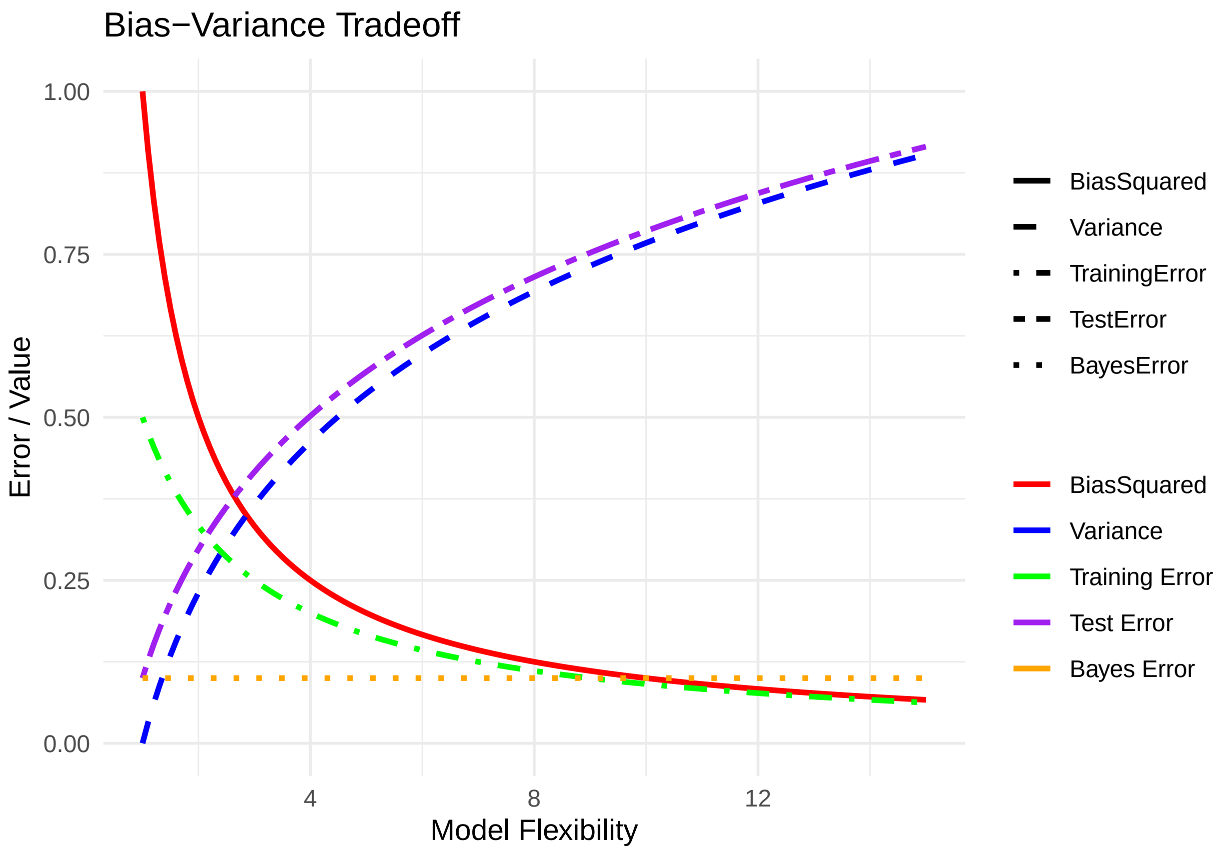
```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```r
flexibility <- seq(1, 15, by = 0.1)

bias_squared <- 1 / flexibility
variance <- log(flexibility) / 3
training_error <- 1 / (flexibility + 1)
test_error <- 0.2 / (flexibility + 1) + variance
bayes_error <- rep(0.1, length(flexibility))

error_data <- data.frame(
  Flexibility = flexibility,
  BiasSquared = bias_squared,
  Variance = variance,
  TrainingError = training_error,
  TestError = test_error,
  BayesError = bayes_error
)

library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.3
```

```
error_data_melted <- melt(error_data, id = "Flexibility")

ggplot(error_data_melted, aes(x = Flexibility, y = value, color = variable, linetype = variable)) +
  geom_line(size = 1) +
  labs(title = "Bias-Variance Tradeoff",
       x = "Model Flexibility",
       y = "Error / Value") +
  theme_minimal() +
  scale_color_manual(values = c("red", "blue", "green", "purple", "orange"),
                     labels = c("BiasSquared", "Variance", "Training Error", "Test Error", "Bayes Error
  scale_linetype_manual(values = c("solid", "dashed", "dotdash", "twodash", "dotted")) +
  theme(legend.title = element_blank()) +
  ylim(0, 1)
```

```
## 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.
```



Question 2 b) (b) Explain why each of the five curves has the shape displayed in part (a).

(Squared) Bias:

Decreases as flexibility increases. Less flexible (simpler) models make strong assumptions and fail to capture data patterns, leading to high bias. As the model becomes more flexible, it fits the data better, reducing this bias.

Variance:

Increases as flexibility increases. More flexible models are highly sensitive to the specific training data they are given. This causes them to vary greatly with small changes in the data, resulting in higher variance as the model tries to fit every detail (including noise).

Training Error:

Decreases rapidly with increasing flexibility. Since flexible models can adjust closely to the training data, they achieve very low error on that data. As flexibility increases, the model essentially memorizes the training data, leading to a sharp drop in training error.

Test Error:

Forms a U-shaped curve. Initially, test error decreases as increased flexibility reduces bias. However, after a certain point, the increase in variance outweighs the bias reduction, causing overfitting and an increase in test error on unseen data.

Bayes Error:

Remains constant. This error is due to the irreducible noise in the data itself, which cannot be reduced by any model, regardless of its flexibility. Hence, it stays flat across all levels of model flexibility.

Question 3 This exercise involves the Auto data set studied in the lab. Make sure that the missing values have been removed from the data. **(a)** Which of the predictors are quantitative, and which are qualitative?

```
# Question 3a)

auto_data <- read.table("C:/Users/dokan/Downloads/Assignment_1_RSM8512/Auto.data", header = TRUE, na.st


quantitative <- sapply(auto_data, is.numeric)
qualitative <- !quantitative

cat("Quantitative Predictors:", names(auto_data)[quantitative], "\n")
```

```
## Quantitative Predictors: mpg cylinders displacement horsepower weight acceleration year origin
```

```
cat("Qualitative Predictors:", names(auto_data)[qualitative], "\n")
```

```
## Qualitative Predictors: name
```

(b) What is the range of each quantitative predictor? You can answer this using the range() function.

```
#Question 3b)

quantitative_cols <- names(auto_data)[quantitative]
range_values <- sapply(auto_data[quantitative_cols], function(x) range(x, na.rm = TRUE))

cat("Range of each quantitative predictor: \n")
```

```
## Range of each quantitative predictor:
```

```
print(range_values)
```

```
##       mpg cylinders displacement horsepower weight acceleration year origin
## [1,]  9.0         3           68         46   1613          8.0   70      1
## [2,] 46.6         8          455        230   5140         24.8   82      3
```

(c) What is the mean and standard deviation of each quantitative predictor?

```
#Question 3c)

mean_values <- sapply(auto_data[quantitative_cols], function(x) mean(x, na.rm = TRUE))
sd_values <- sapply(auto_data[quantitative_cols], function(x) sd(x, na.rm = TRUE))

cat("Mean of each quantitative predictor: \n")
```

```
## Mean of each quantitative predictor:
```

```
print(mean_values)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration 
##    23.515869     5.458438   193.532746   104.469388  2970.261965    15.555668 
##         year       origin 
##    75.994962     1.574307 
```

```
cat("Standard deviation of each quantitative predictor: \n")
```

```
## Standard deviation of each quantitative predictor:
```

```
print(sd_values)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration 
##    7.8258039    1.7015770  104.3795833   38.4911599  847.9041195    2.7499953 
##         year       origin 
##    3.6900049    0.8025495 
```

(d) Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

```
#Question 3d)

auto_data_subset <- auto_data[-(10:85), ]

subset_range <- sapply(auto_data_subset[quantitative_cols], function(x) range(x, na.rm = TRUE))
subset_mean <- sapply(auto_data_subset[quantitative_cols], function(x) mean(x, na.rm = TRUE))
subset_sd <- sapply(auto_data_subset[quantitative_cols], function(x) sd(x, na.rm = TRUE))

cat("Range of each predictor in subset: \n")
```

```
## Range of each predictor in subset:
```

```
print(subset_range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year origin
## [1,] 11.0         3           68         46   1649          8.5   70      1
## [2,] 46.6         8          455        230   4997         24.8   82      3
```

```
cat("Mean of each predictor in subset: \n")
```

```
## Mean of each predictor in subset:
```

```
print(subset_mean)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration 
##    24.438629     5.370717   187.049844   100.955836  2933.962617    15.723053 
##         year       origin 
##    77.152648     1.598131
```

```
cat("Standard deviation of each predictor in subset: \n")
```

```
## Standard deviation of each predictor in subset:
```

```
print(subset_sd)
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration 
##    7.9081842    1.6534857   99.6353853   35.8955668  810.6429384    2.6805138 
##         year       origin 
##    3.1112298    0.8161627
```

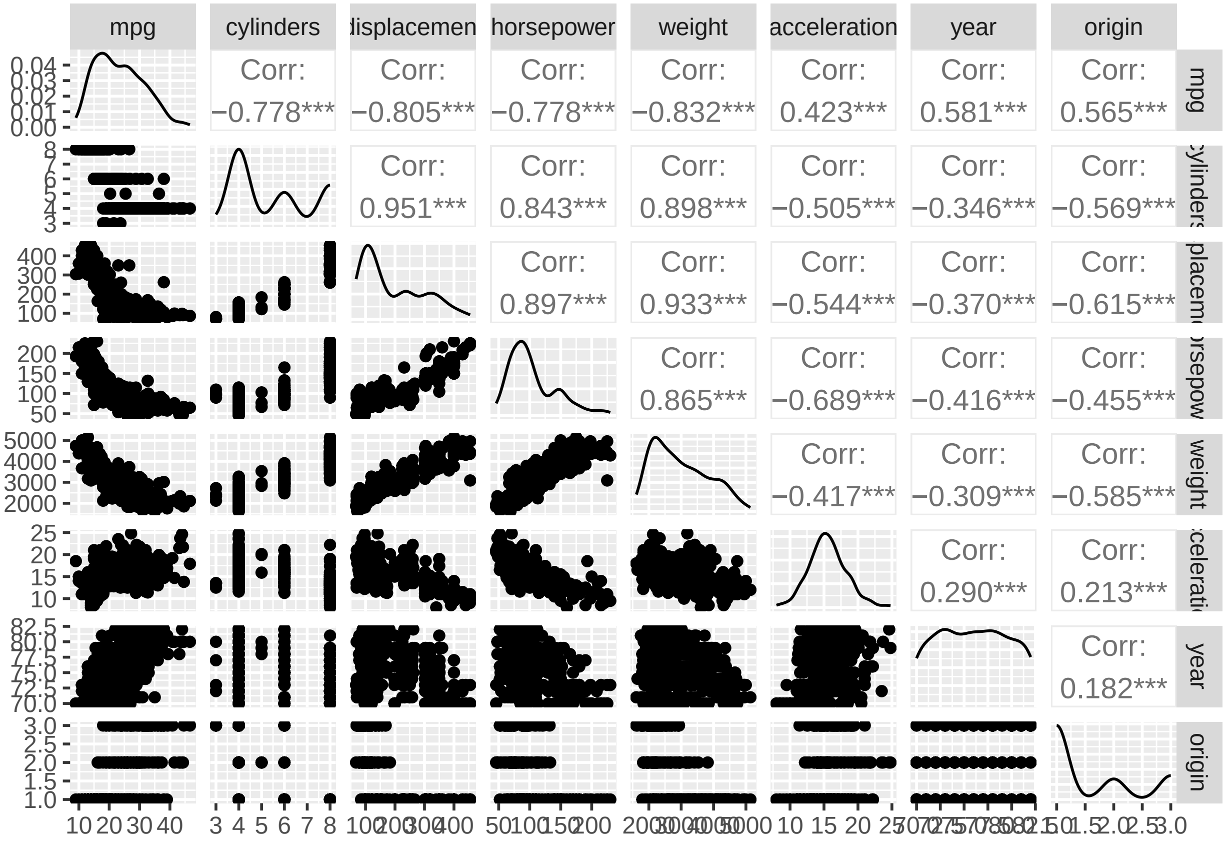
(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

```
#Question 3e)

library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.1.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from   
##   +.gg   ggplot2
```

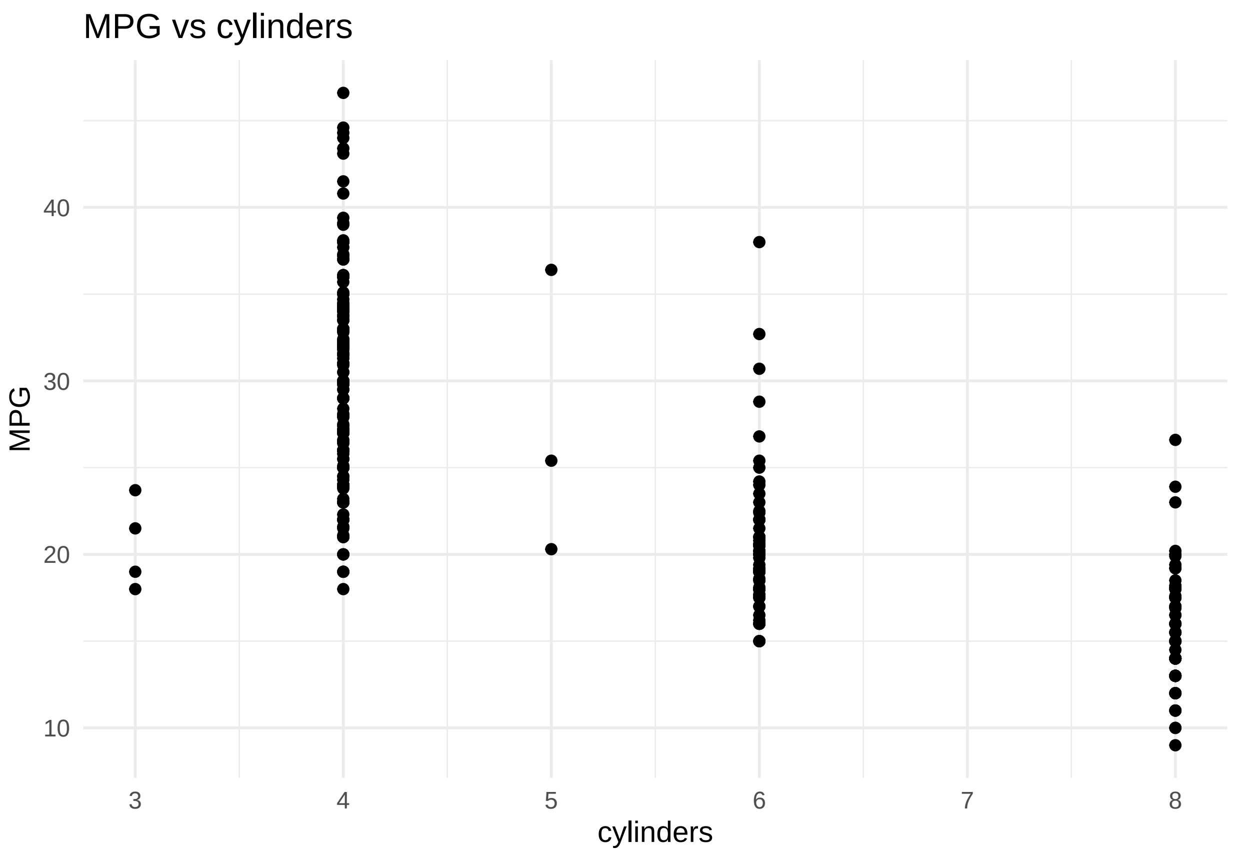
```
auto_data_clean <- na.omit(auto_data)

# Create scatterplot matrix for quantitative predictors
ggpairs(auto_data_clean[quantitative_cols])
```

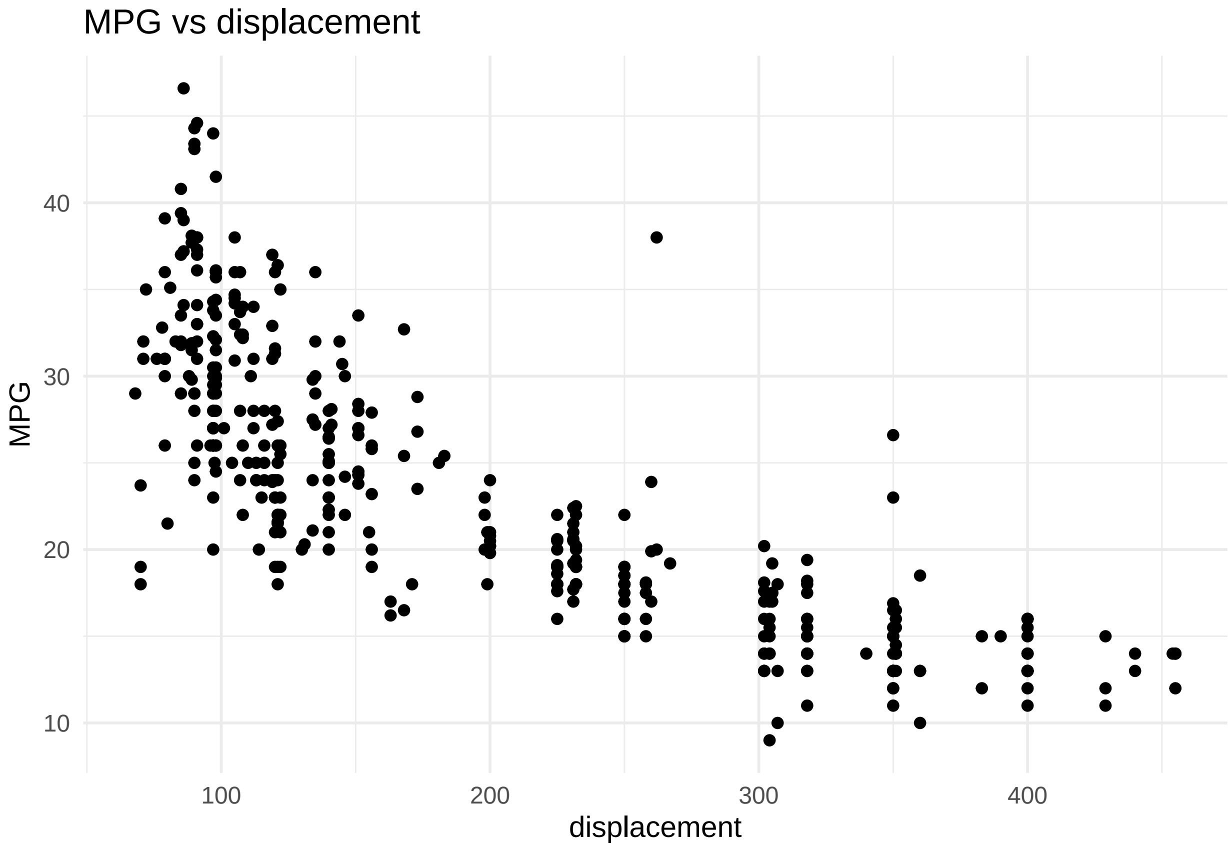


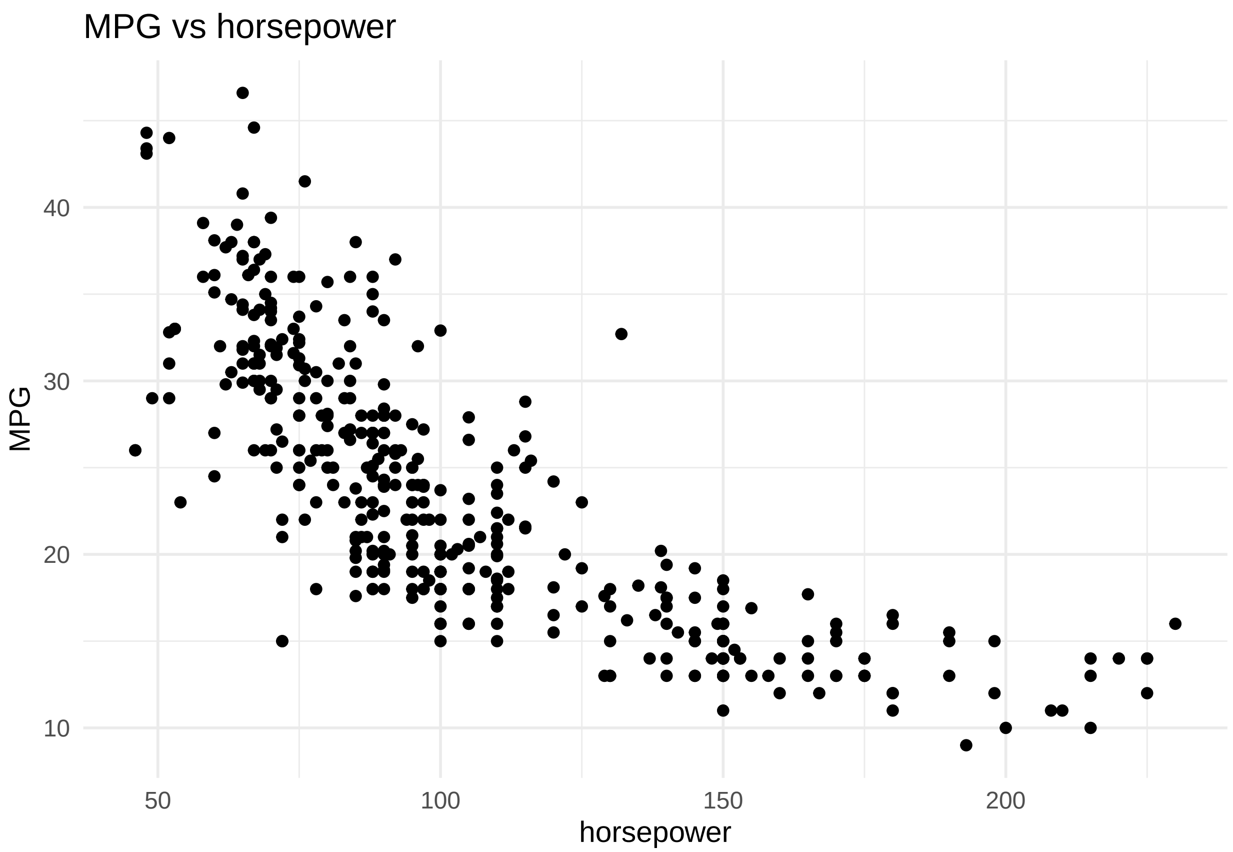
(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

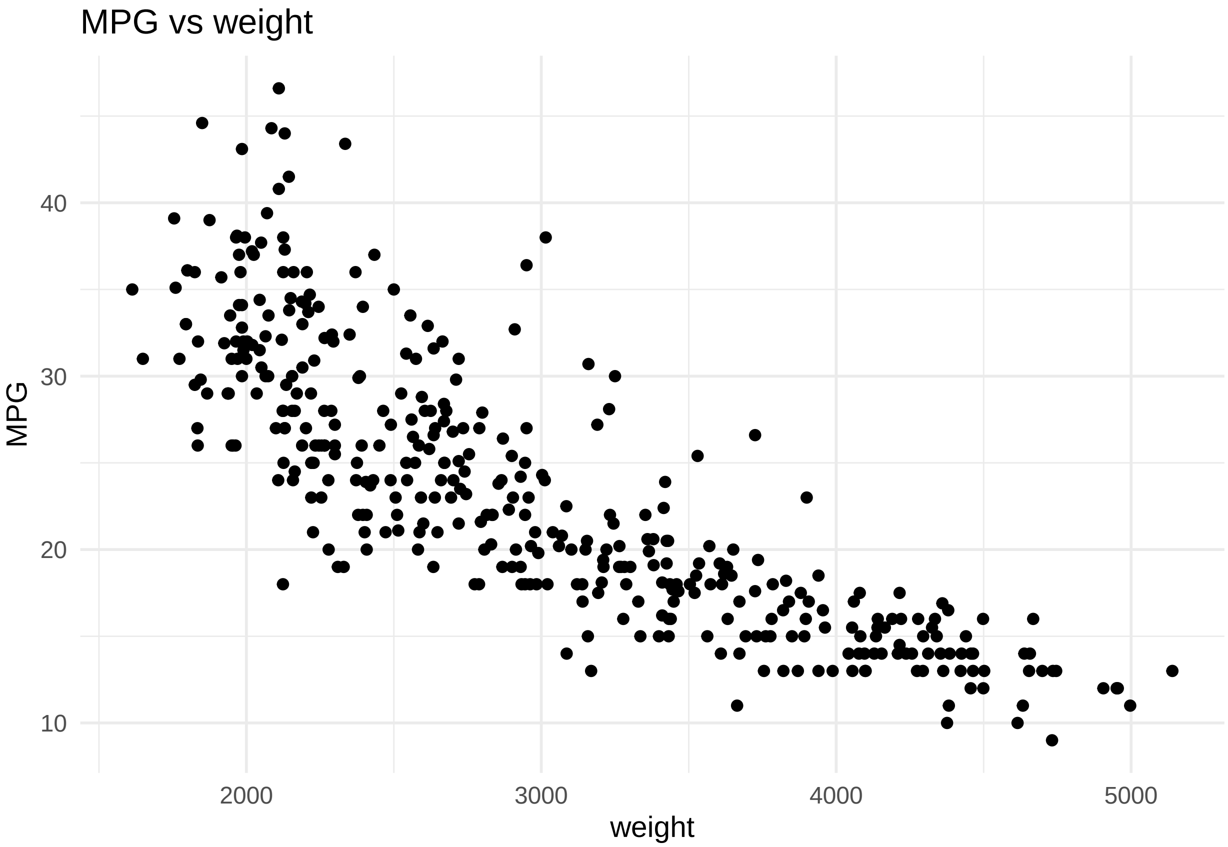
```r
#Question 3f)
# Create individual scatterplots to explore the relationship between mpg and other predictors
library(ggplot2)

for (var in quantitative_cols[quantitative_cols != "mpg"]) {
  print(
    ggplot(auto_data_clean, aes_string(x = var, y = "mpg")) +
      geom_point() +
      labs(title = paste("MPG vs", var), x = var, y = "MPG") +
      theme_minimal()
  )
}
```

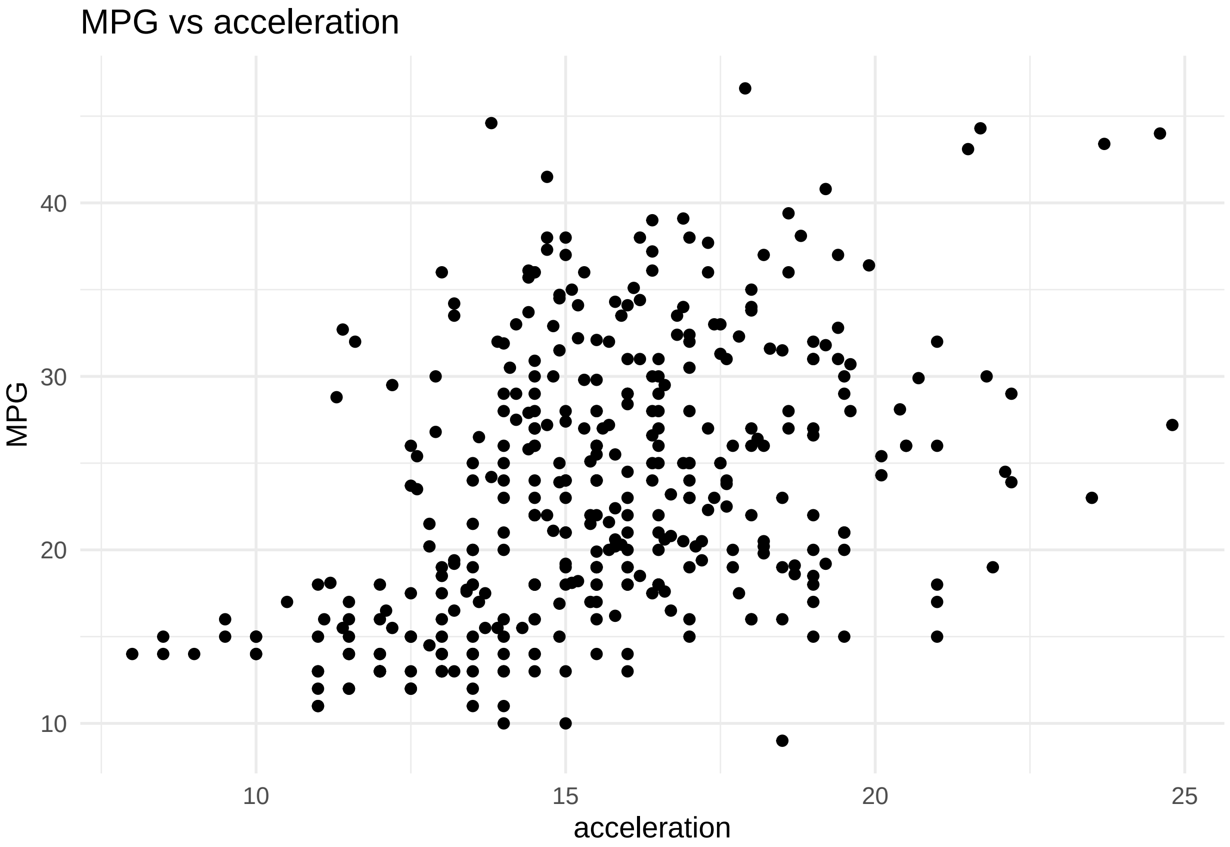
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

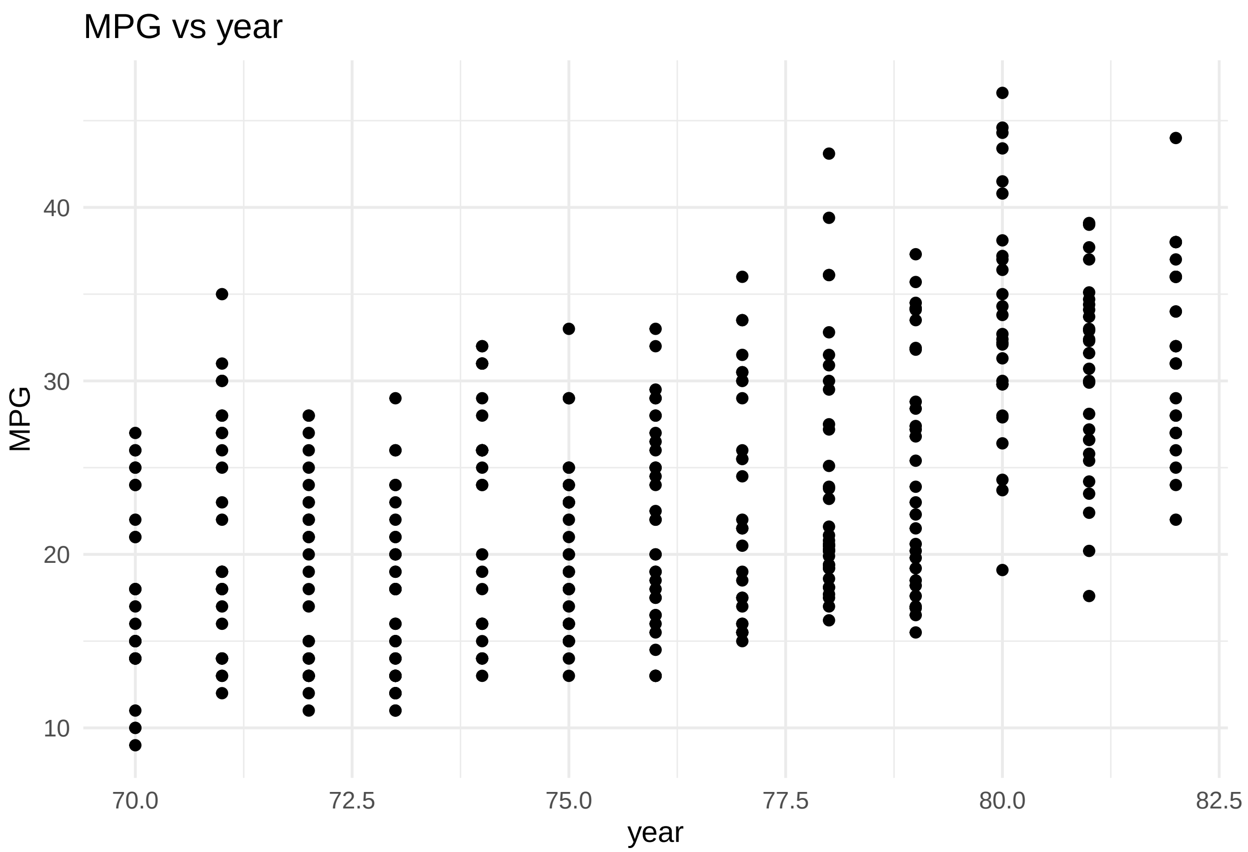

MPG vs cylinders
MPG
40
30
20
10
3
4
5
6
7
8
cylinders

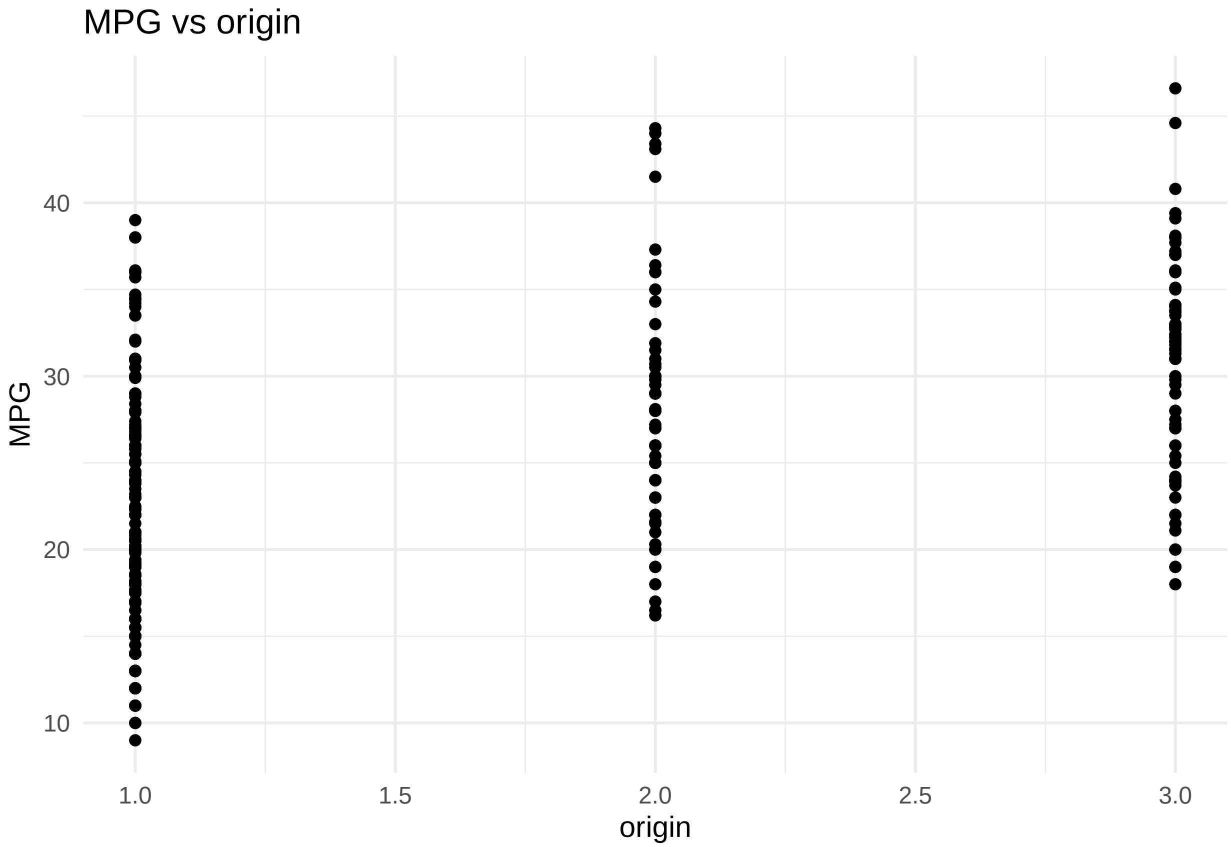

MPG vs displacement
MPG
40
30
20
10
100
200
300
400
displacement


MPG vs horsepower
MPG
40
30
20
10
50
100
150
200
horsepower


MPG vs weight
MPG
40
30
20
10
2000
3000
4000
5000
weight


MPG vs acceleration
MPG
40
30
20
10
10
15
20
25
acceleration


MPG vs year
MPG
40
30
20
10
year
70.0
72.5
75.0
77.5
80.0
82.5


MPG vs origin
MPG
40
30
20
10
1.0
1.5
2.0
2.5
3.0
origin