Mileage and average speed analysis

Devam Patel

2023-03-31

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

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

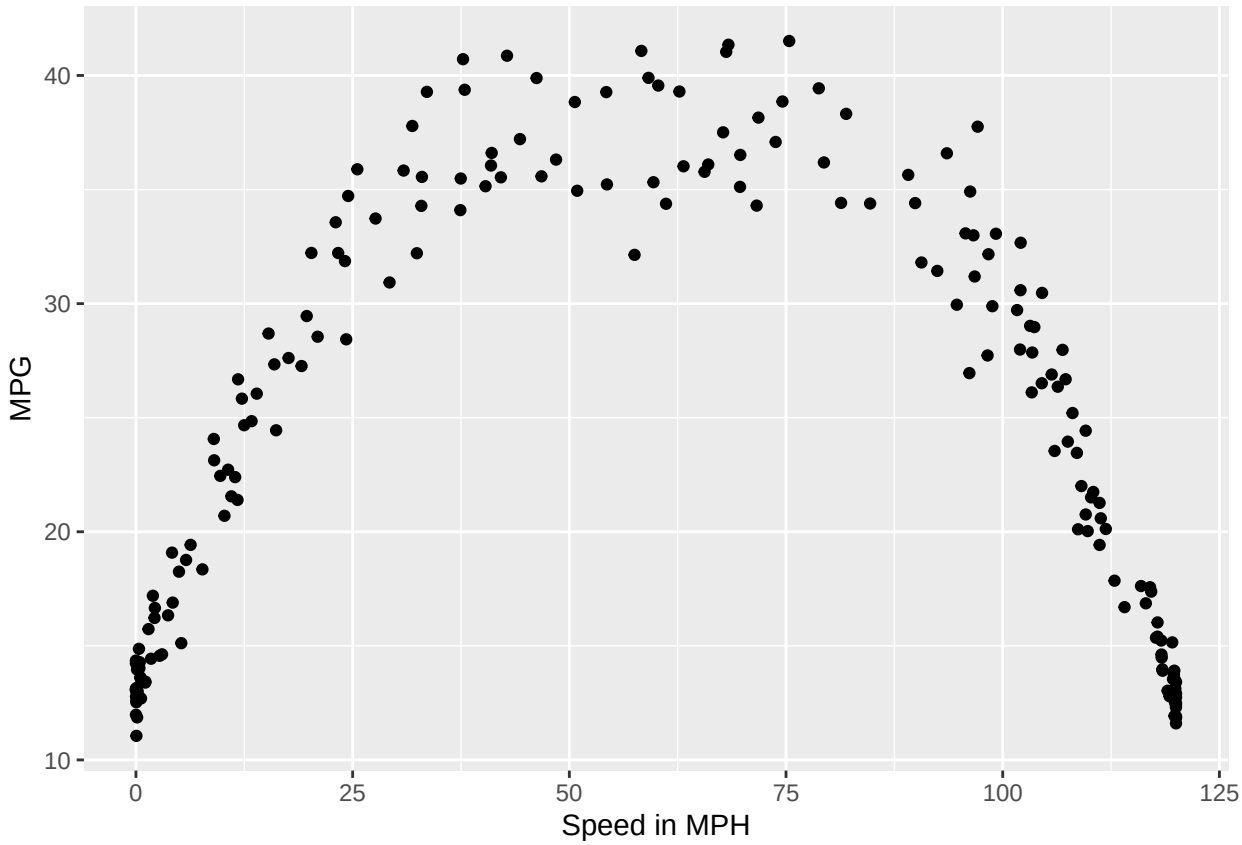
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(moderndive)
data = read.delim('mpg_data.txt')

head(data)
```

```
##   Speed_.mph.      MPG
## 1   79.373049 36.18784
## 2    1.080288 13.39034
## 3  111.317162 20.59088
## 4   54.263529 39.27212
## 5  106.904475 27.97295
## 6   13.940522 26.05444
```

1. Create a scatterplot of the data from the calibration runs, plotting the MPG on the vertical axis and speed on the horizontal axis (be sure to properly label your plot). Does there appear to be an association between the speed the bike is driven at and the MPG? If so, explain what the nature of the relationship seems to be.

```
ggplot(data=data, aes(x=Speed_.mph., y=MPG)) + geom_point() +xlab('Speed in MPH') + ylab('MPG')
```

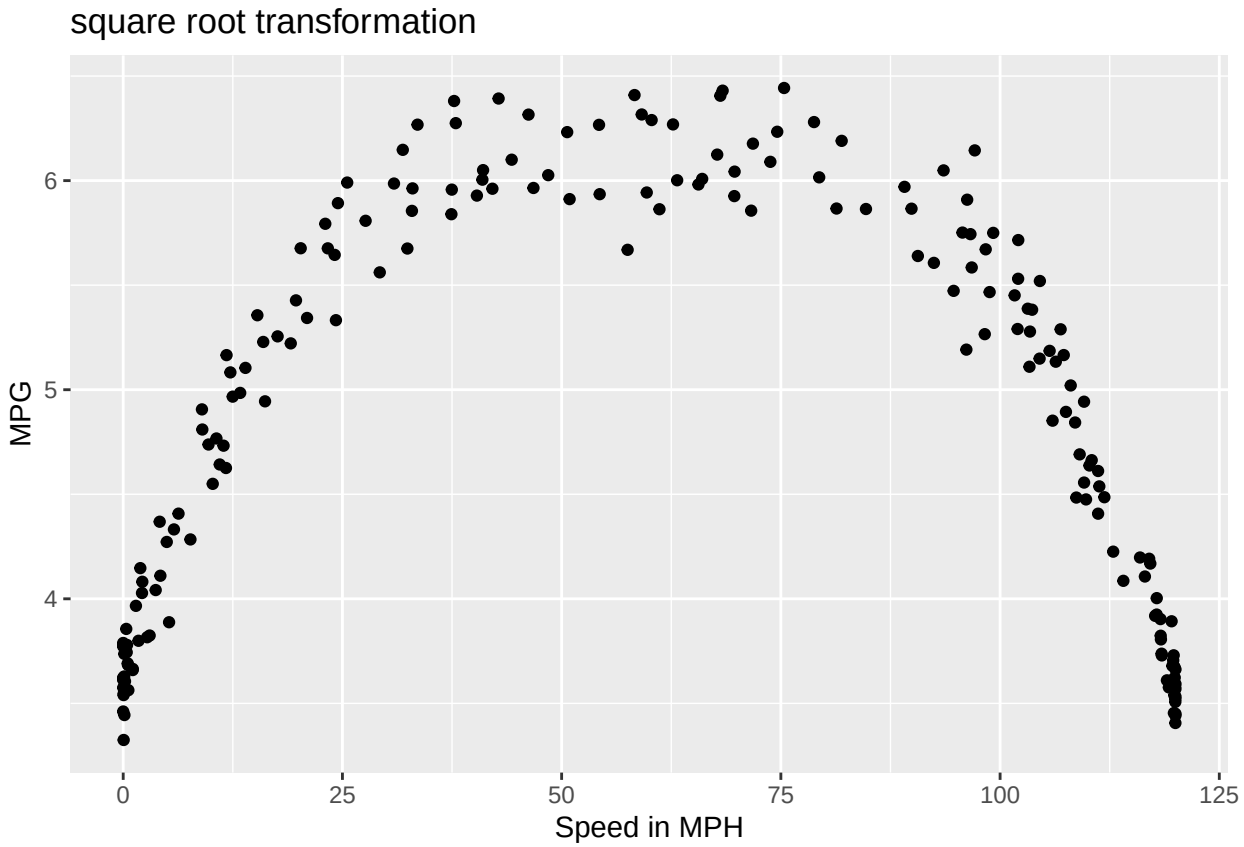


Answer 1. Based on the plot above, it is evident that there is a quadratic relationship between speed and miles per gallon. The plot is not linear and it is inappropriate to use SLM as linearity can not be corrected as the plot is not monotonic. More specifically, for speed < 60 MPH, the MPG increases as speed increases, while after 60 MPH, MPG decreases as speed increases.

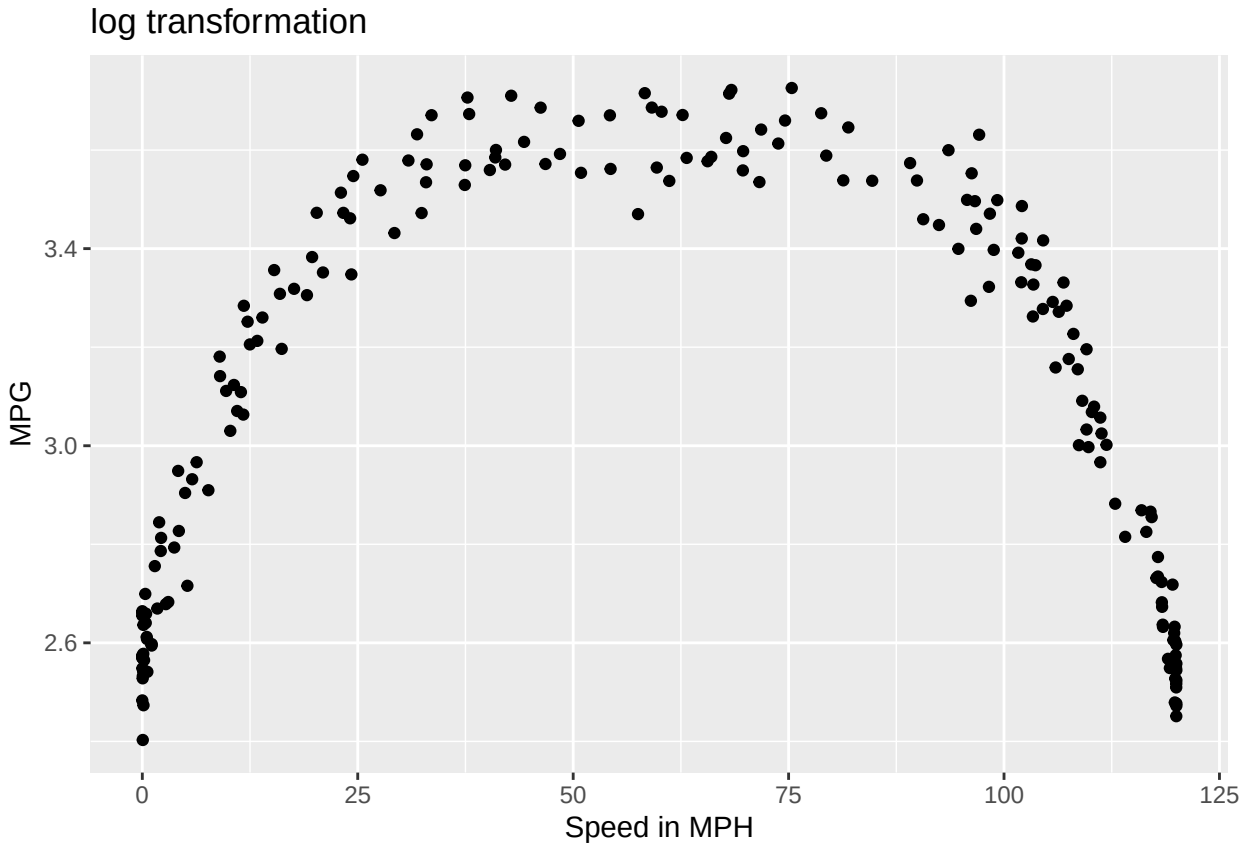
Question 2. The National Highway Traffic Safety Administration (NHTSA) requires all vehicles marketed in the US to provide ranges for what the mean MPG is at a variety of speeds. Treating MPG as the response variable and speed as the explanatory variable, are enough of the model assumptions satisfied in order to fit a polynomial model to this data towards the prior purpose? If not, explain what must be done to address the deviations from the needed model assumptions (if necessary).

Answer 2. As evident from the plot in part 1, we can not use simple linear model as most importantly the linearity assumption is violated because while the curve is simple, it is not monotonic. In addtion, the assumption of homoscedasticity or constant variance is also violated by the look of the plot as the data is clustered towards the tails of the graph. To address this, a form of transformation of the dataset is needed, which are explored below

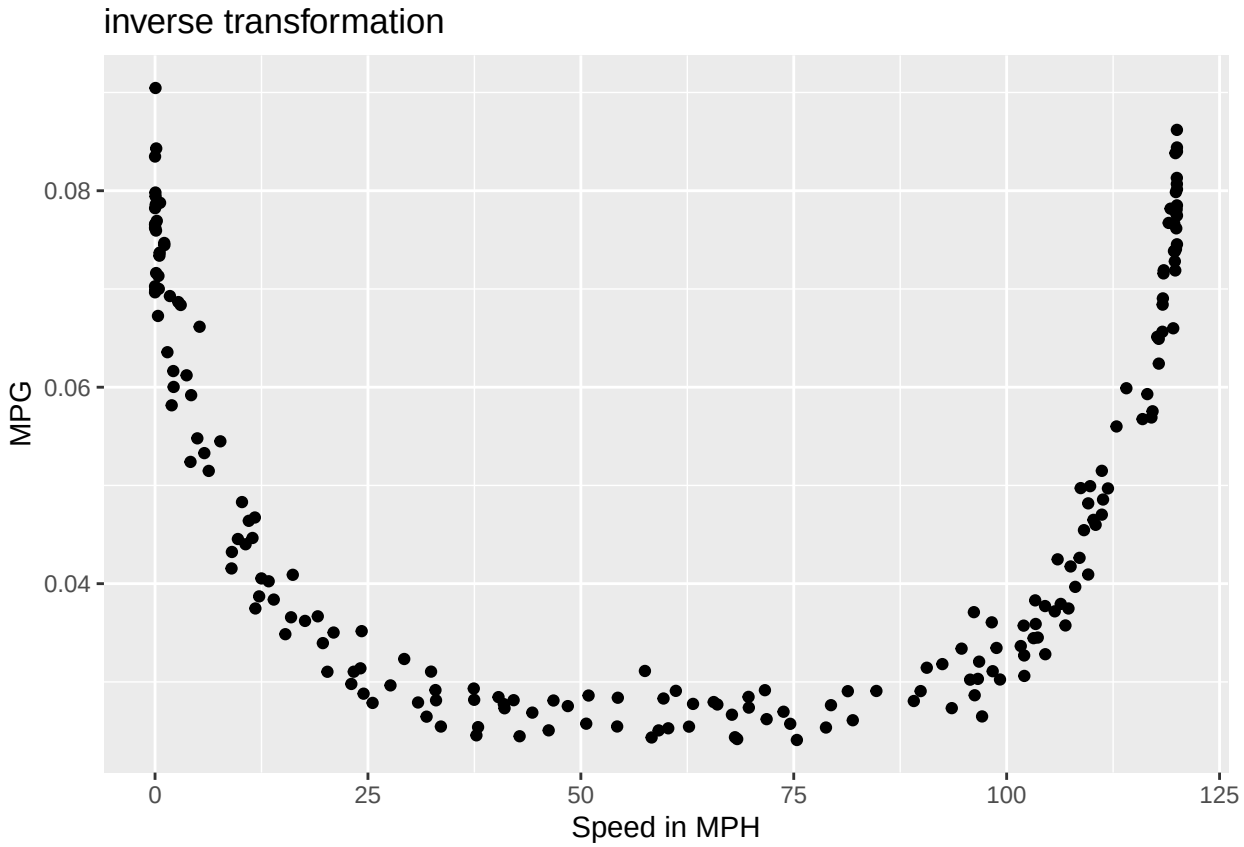
```
ggplot(data=data, aes(x=(Speed_.mph.), y=(MPG)^0.5)) + geom_point() +xlab('Speed in MPH') + ylab('MPG')+
```




```
ggplot(data=data, aes(x=(Speed_.mph.), y=log(MPG))) + geom_point() +xlab('Speed in MPH') + ylab('MPG') +
```




```
ggplot(data=data, aes(x=(Speed_.mph.), y=1/(MPG))) + geom_point() +xlab('Speed in MPH') + ylab('MPG') +
```



Log Transformation was applied to MPG

```
data$y_trans = log(data$MPG)
head(data)
```

```
##   Speed_.mph.      MPG  y_trans
## 1   79.373049 36.18784 3.588723
## 2    1.080288 13.39034 2.594534
## 3  111.317162 20.59088 3.024848
## 4   54.263529 39.27212 3.670515
## 5  106.904475 27.97295 3.331238
## 6   13.940522 26.05444 3.260188
```

Question 3. After addressing any issues in part 2, fit a polynomial model to the data. Clearly explain the process with which you went about arriving at the order of the polynomial model you fit (you will need to fit several polynomial models and compare them). Explicitly write out the estimated model equation for the polynomial model you decided upon (on the transformed scales if data transformations were needed).

```
model1 = lm(data = data, y_trans ~ Speed_.mph.)
summary(model1)
```

```
##
## Call:
## lm(formula = y_trans ~ Speed_.mph., data = data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.71069 -0.44803  0.07394  0.41232  0.60200
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.1135686  0.0496068  62.765   <2e-16 ***
## Speed_.mph. 0.0001379  0.0006485   0.213    0.832
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.42 on 198 degrees of freedom
## Multiple R-squared:  0.0002284,  Adjusted R-squared:  -0.004821
## F-statistic: 0.04524 on 1 and 198 DF,  p-value: 0.8318
```

```
model2 = lm(data = data, y_trans ~ Speed_.mph. + I(Speed_.mph.^2))
summary(model2)
```

```
##
## Call:
## lm(formula = y_trans ~ Speed_.mph. + I(Speed_.mph.^2), data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.303201 -0.089438 -0.005311  0.091005  0.281194
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.649e+00  1.721e-02  153.97   <2e-16 ***
## Speed_.mph.       3.745e-02  8.096e-04   46.26   <2e-16 ***
## I(Speed_.mph.^2) -3.113e-04  6.577e-06  -47.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1197 on 197 degrees of freedom
## Multiple R-squared:  0.9192, Adjusted R-squared:  0.9184
## F-statistic:  1121 on 2 and 197 DF,  p-value: < 2.2e-16
```

```
model3 = lm(data = data, y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3))
summary(model3)
```

```
##
## Call:
## lm(formula = y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3),
##     data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.301422 -0.089994 -0.008313  0.089468  0.265627
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.660e+00  1.920e-02 138.547  < 2e-16 ***
## Speed_.mph.       3.526e-02  1.864e-03  18.918  < 2e-16 ***
## I(Speed_.mph.^2) -2.611e-04  3.915e-05  -6.669 2.56e-10 ***
## I(Speed_.mph.^3) -2.804e-07  2.156e-07  -1.300    0.195
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1195 on 196 degrees of freedom
## Multiple R-squared:  0.9199, Adjusted R-squared:  0.9187
## F-statistic: 750.2 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
model4 = lm(data = data, y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3) + I(Speed_.mph.^4)
summary(model4)
```

```
##
## Call:
## lm(formula = y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3) +
##     I(Speed_.mph.^4), data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.183479 -0.046792 -0.002348  0.051251  0.169324
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.572e+00  1.212e-02  212.11   <2e-16 ***
## Speed_.mph.       6.831e-02  2.019e-03   33.83   <2e-16 ***
## I(Speed_.mph.^2) -1.689e-03  7.693e-05  -21.96   <2e-16 ***
## I(Speed_.mph.^3)  1.876e-05  9.873e-07   19.00   <2e-16 ***
## I(Speed_.mph.^4) -7.865e-08  4.045e-09  -19.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06989 on 195 degrees of freedom
## Multiple R-squared:  0.9727, Adjusted R-squared:  0.9722
## F-statistic:  1740 on 4 and 195 DF,  p-value: < 2.2e-16
```

Answer 3. First step for deciding on the model was to take a glance at the simple linear model to get an idea of the explanatory variables explains the response in a linear relationship. Clearly, the p-value was very high, which leads to trying a polynomial model with different degrees. First, I decided to use degree 2 as the plot changes direction only once. With such a model, the p-value was significant as it was < 0.05 but the adjusted R-squared was around 0.91. I continued to try with degrees 3 and 4, and noticed a massive jump as I evaluated the model with degree 4, with the adjusted r-square value increasing from 0.91 to 0.97. So I continued to try higher degrees that are even that would optimize the adjusted r-squared value.

```
model5 = lm(data = data, y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3) + I(Speed_.mph.^4)
summary(model5)
```

```
##
## Call:
## lm(formula = y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3) +
##     I(Speed_.mph.^4) + I(Speed_.mph.^5) + I(Speed_.mph.^6), data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.182343 -0.047475 -0.003195  0.050774  0.170197
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.573e+00  1.340e-02 191.990  < 2e-16 ***
## Speed_.mph.       6.787e-02  5.039e-03  13.469  < 2e-16 ***
## I(Speed_.mph.^2) -1.669e-03  4.463e-04  -3.739 0.000243 ***
## I(Speed_.mph.^3)  1.870e-05  1.529e-05   1.223 0.222793
## I(Speed_.mph.^4) -8.634e-08  2.425e-07  -0.356 0.722165
## I(Speed_.mph.^5)  1.076e-10  1.791e-09   0.060 0.952152
## I(Speed_.mph.^6) -4.115e-13  4.988e-12  -0.082 0.934337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07024 on 193 degrees of freedom
## Multiple R-squared:  0.9728, Adjusted R-squared:  0.9719
## F-statistic:  1148 on 6 and 193 DF,  p-value: < 2.2e-16
```

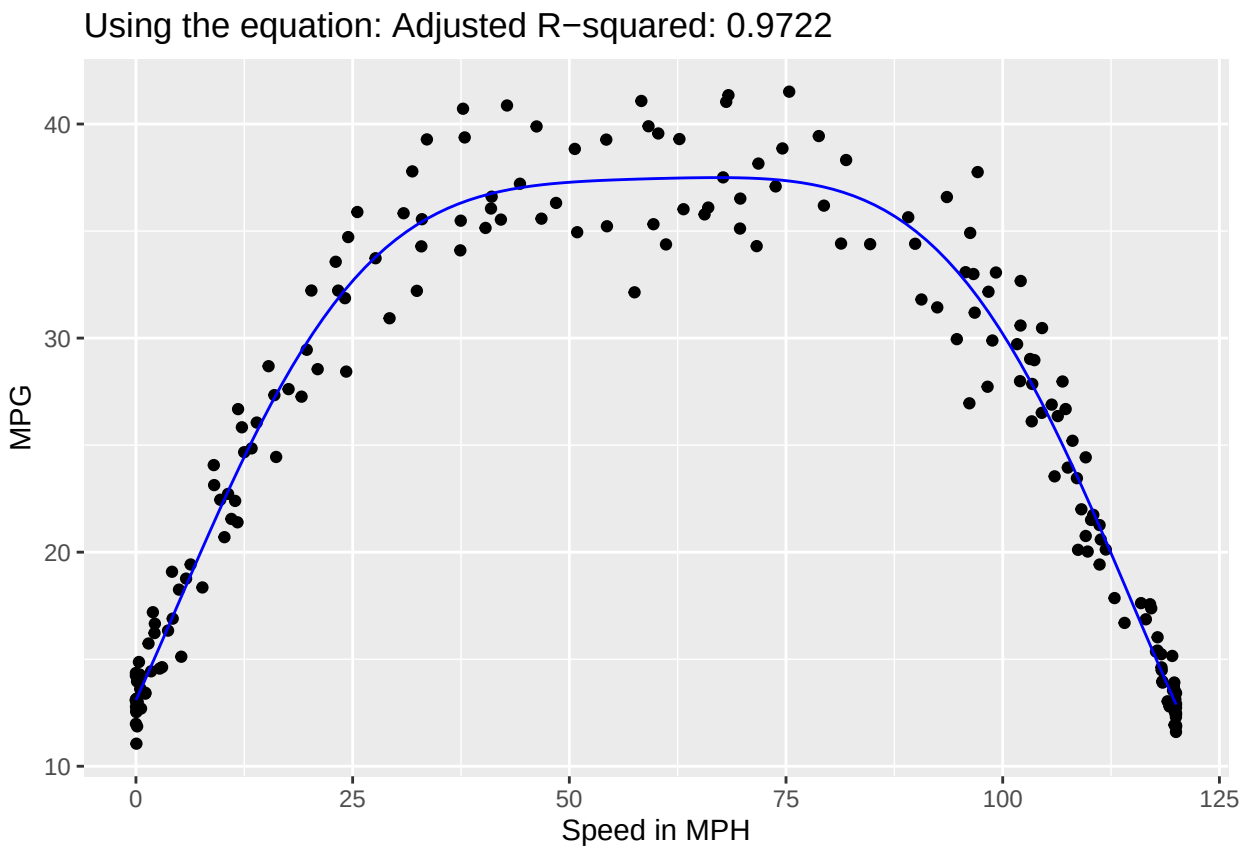
```
model6 = lm(data = data, y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3) + I(Speed_.mph.^4)
summary(model6)
```

```
##
## Call:
## lm(formula = y_trans ~ Speed_.mph. + I(Speed_.mph.^2) + I(Speed_.mph.^3) +
##     I(Speed_.mph.^4) + I(Speed_.mph.^5) + I(Speed_.mph.^6) +
##     I(Speed_.mph.^7) + I(Speed_.mph.^8), data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.183826 -0.048321 -0.001676  0.051792  0.169531
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.568e+00  1.444e-02 177.833  < 2e-16 ***
## Speed_.mph.       7.506e-02  1.032e-02   7.275 8.73e-12 ***
## I(Speed_.mph.^2) -2.933e-03  1.610e-03  -1.823   0.0699 .
## I(Speed_.mph.^3)  1.007e-04  1.007e-04   1.000   0.3184
## I(Speed_.mph.^4) -2.687e-06  3.182e-06  -0.844   0.3995
## I(Speed_.mph.^5)  4.496e-08  5.556e-08   0.809   0.4194
## I(Speed_.mph.^6) -4.299e-10  5.426e-10  -0.792   0.4292
## I(Speed_.mph.^7)  2.146e-12  2.775e-12   0.773   0.4404
## I(Speed_.mph.^8) -4.364e-15  5.786e-15  -0.754   0.4516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07048 on 191 degrees of freedom
## Multiple R-squared:  0.9728, Adjusted R-squared:  0.9717
## F-statistic: 855.4 on 8 and 191 DF,  p-value: < 2.2e-16
```

(Cont.) Answer 3. After exploring models with higher degrees, as seen above, I think the most optimal is degree 4 as the change in adjusted r-squared value plateaus and even decreases when we try higher degrees such as 6 or 8. Therefore, the estimated model equation is MPG = e^(2.572e+00 + (6.831e-02)*X + (-1.689e-03)*X^2 + (1.876e-05)*X^3 + (-7.865e-08)*X^4), X = Speed in MPH.

Question 4. On a scatter plot depicting the MPG on the vertical axis and speed on the horizontal axis (on their original,untransformed measurement scales), overlay the estimated model on the plot (in the event you transformed any of your variables, this may necessitate back transforming the polynomial model that was constructed on the transformed data).

```
poly_eqn <- function(X) {
  return( (exp(2.572e+00  + (6.831e-02)*X + (-1.689e-03)*X^2 + (1.876e-05)*X^3 + (-7.865e-08)*X^4 ))  )
}




ggplot(data=data, aes(x=Speed_.mph., y=MPG)) + geom_point() +xlab('Speed in MPH') + ylab('MPG')+
  stat_function(fun = poly_eqn, color='blue')+
  ggtitle('Using the equation: Adjusted R-squared: 0.9722')
```



Question 5. From the model constructed in part 3, can one conclude that there is a statistically significant relationship between MPG and the speed? Explain what procedure you used to determine so and why you arrived at your conclusion

We can conclude that there is a statistically significant relationship between MPG and speed based on the model summaries above. According to the model summaries, the p-value is lass than 2.2e-16, which is less than the threshold of 0.05 and, thus, significant. In addition, a high R-squared value 0f 0.9722 indicates that the model fits the data well and that the speed variable has a strong relationship with the response variable, MPG.

Question 6. Calculate the coefficient of determination for the model on the original measurement scale (if transformations were applied to the data, calculations of the various sums of squares requires back transforming the fitted values from the polynomial model on the transformed data to get the fitted values and residuals on the original scale)

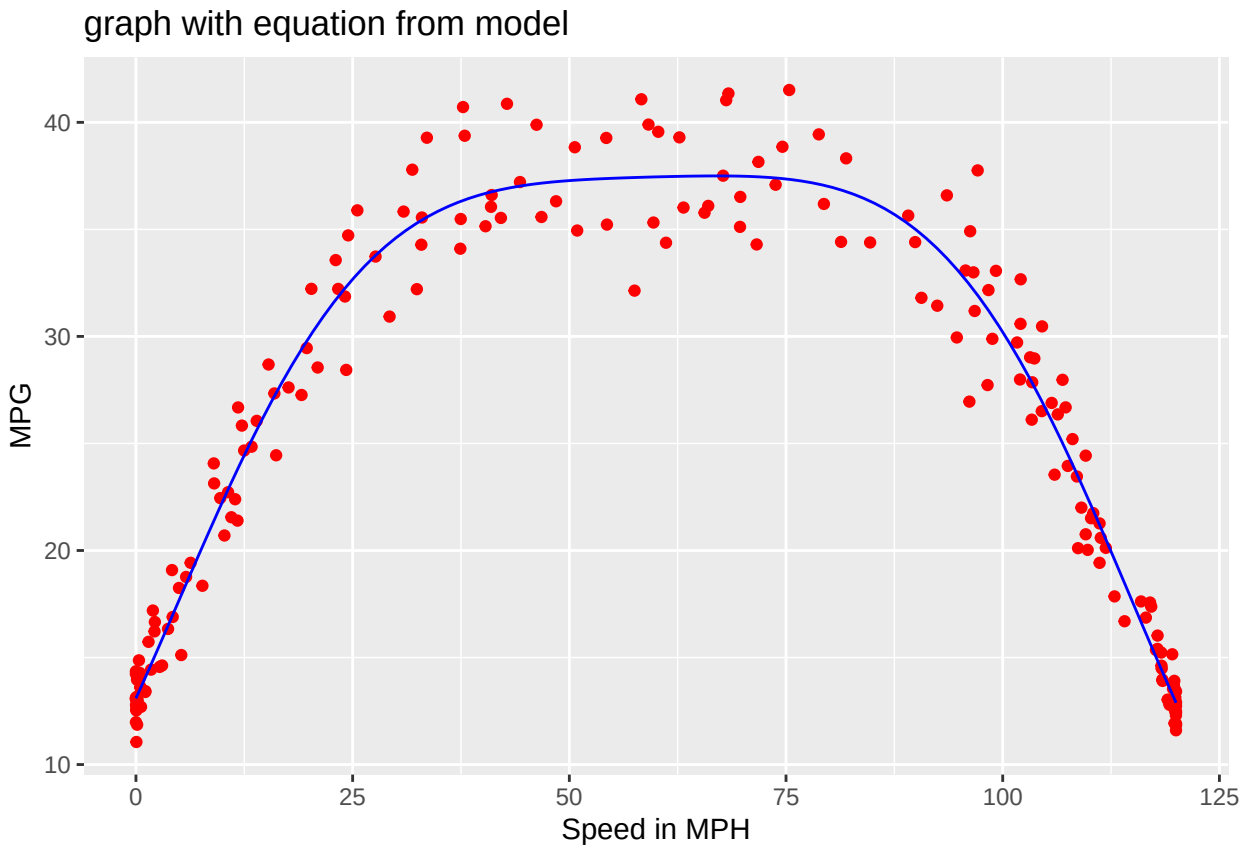
```
fitted = predict(model4)

fitted_orig = exp(fitted)

y = data$MPG

SStot <- sum((y - mean(y))^2)
SSres <- sum((y - fitted_orig)^2)
Rsquared <- 1 - SSres/SStot

Rsquared
```

```
## [1] 0.9639837
```

Answer 6. Based on the calculations above, the coefficient of determination for the model, or the R-squared value, on the original scale is calculated to 0.9639.

Question 7. According to the model constructed in part 3, at what speed is the engine most fuel efficient (i.e. what speed does it have the highest MPG on average). Explain how you arrived at this value (this can certainly be ascertained analytically, but providing a numerically approximated value is also acceptable as well).

```
ggplot(data=data, aes(x=Speed_.mph., y=MPG)) + geom_point(col= 'red' ) +xlab('Speed in MPH') + ylab('MP
```




```
a <- 50
b <- 75
```

```
f <- function(X) exp(2.572e+00  + (6.831e-02)*X + (-1.689e-03)*X^2 + (1.876e-05)*X^3 + (-7.865e-08)*X^4

result <- optimize(f, interval = c(a, b), maximum = TRUE)
result$objective
```

```
## [1] 37.49762
```

```
result$maximum
```

```
## [1] 67.06625
```

Answer 7. Based on the graph above, we could approximate that the engine is the most fuel efficient between 50 and 75. Using those values as the interval, we can optimize the function and find the absolute maxima of the function from the model. After optimizing, engine is most fuel efficient when the speed is 67.06625 MPH with the MPG of 37.497, based on the equation of the model.

Question 8. On a scatter plot depicting the MPG on the vertical axis and speed on the horizontal axis (on their original, untransformed measurement scales), overlay 90% confidence bands for the mean MPG as functions of the speed (in the event you transformed any of your variables, this will necessitate back transforming the 90% confidence bands for the polynomial model that were constructed on the transformed data).

```
plot(data$Speed_.mph., data$MPG, col = 'blue', xlab = "speed in MPH", ylab = 'MPG')
ci <- predict(model4, interval= 'confidence', level=0.90)
ci <- cbind(data$Speed_.mph., exp(ci))
```

```
ci <- ci[order(ci[,1]),]
points(ci[,1], ci[,2], type='l', col='red')
points(ci[,1], ci[,3], type='l', lty=2 , col='green')
points(ci[,1], ci[,4], type='l', lty=2 ,col='green')
```