# HW6

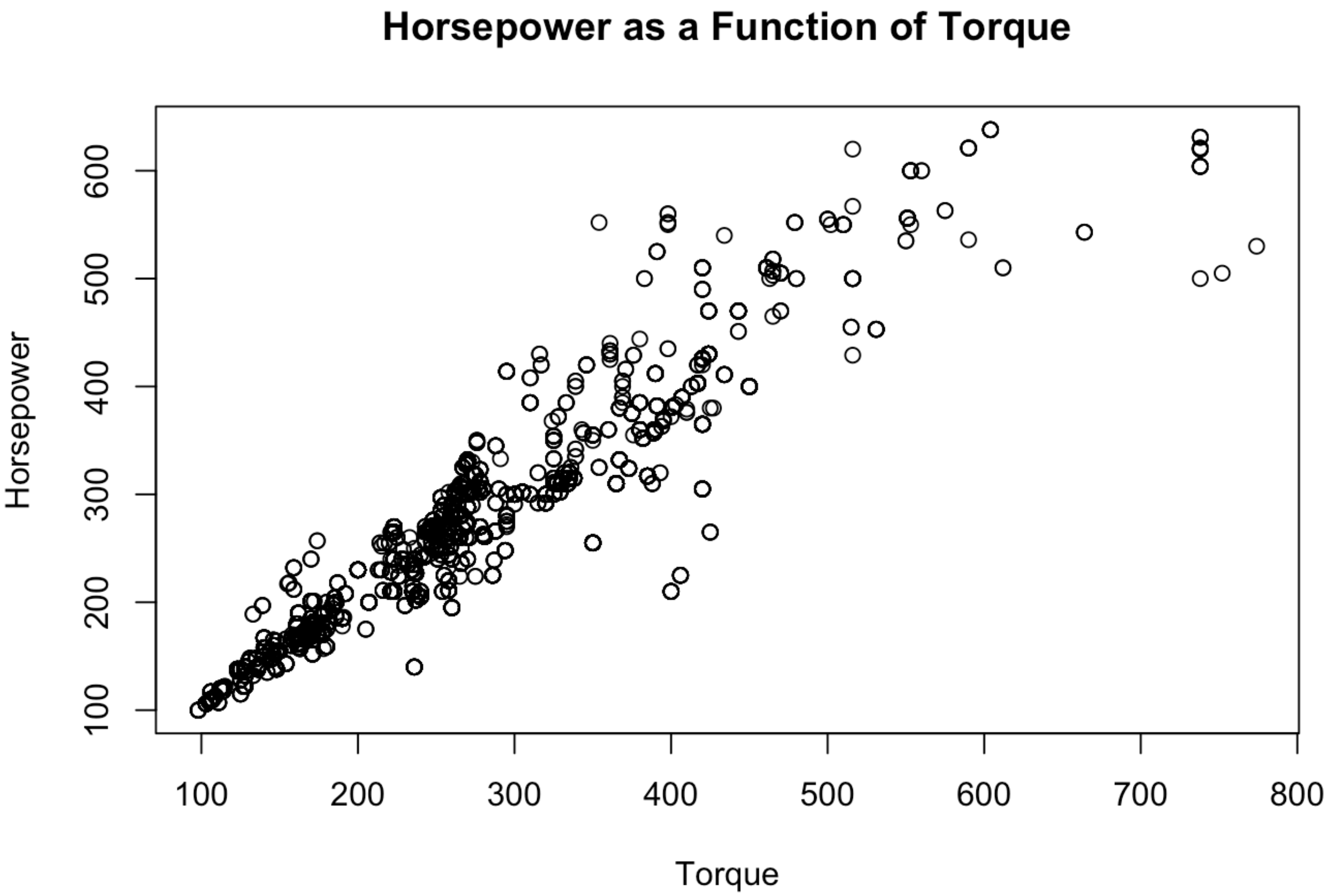
### 2024-06-01

# Problem 1

Horsepower as a function of torque.

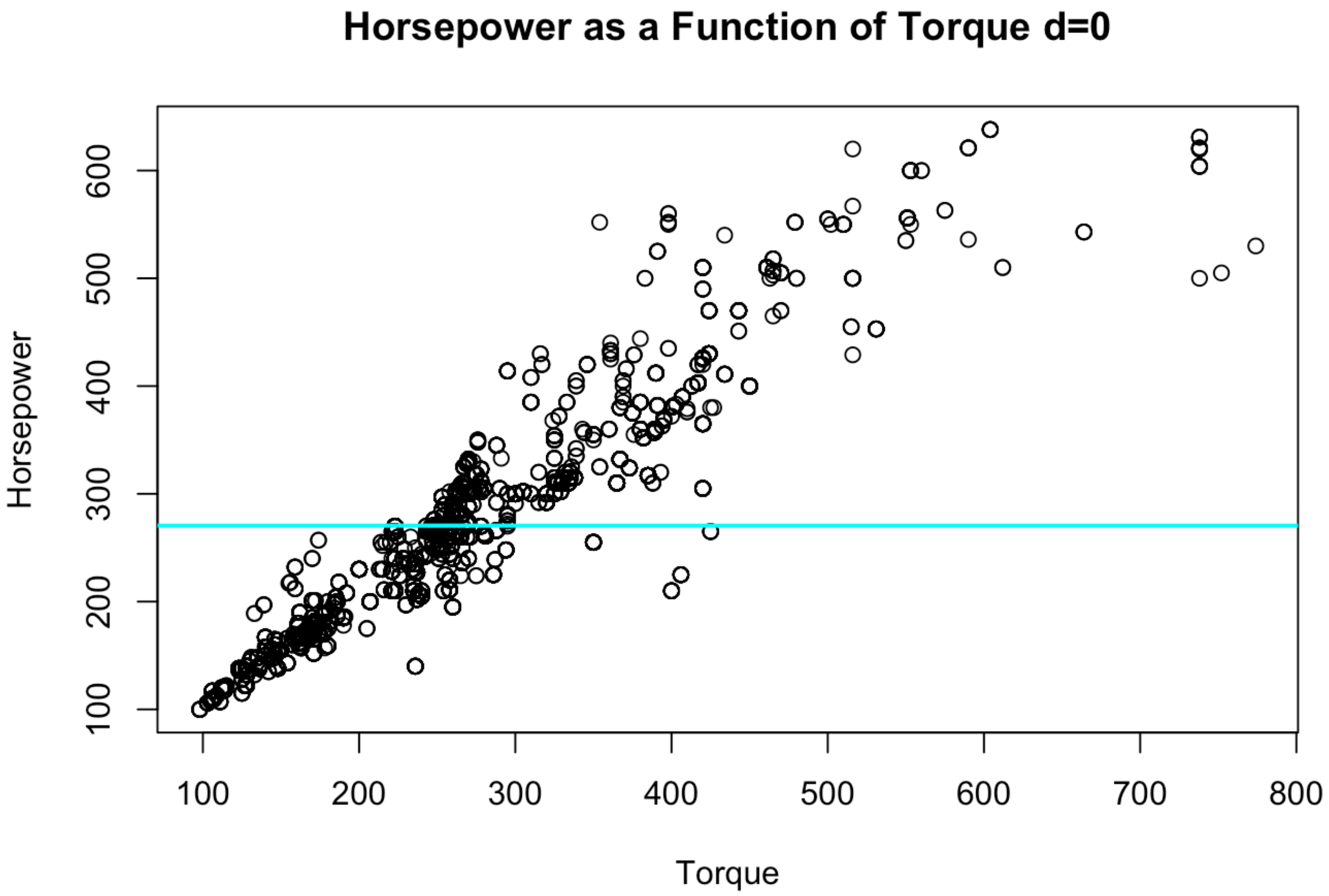
A.

```
dataset <- read.csv("cars.csv")
horsepower <- dataset$Engine.Information.Engine.Statistics.Horsepower
torque <- dataset$Engine.Information.Engine.Statistics.Torque
plot(torque, horsepower,
     main="Horsepower as a Function of Torque",
     xlab="Torque",
     ylab="Horsepower")
```

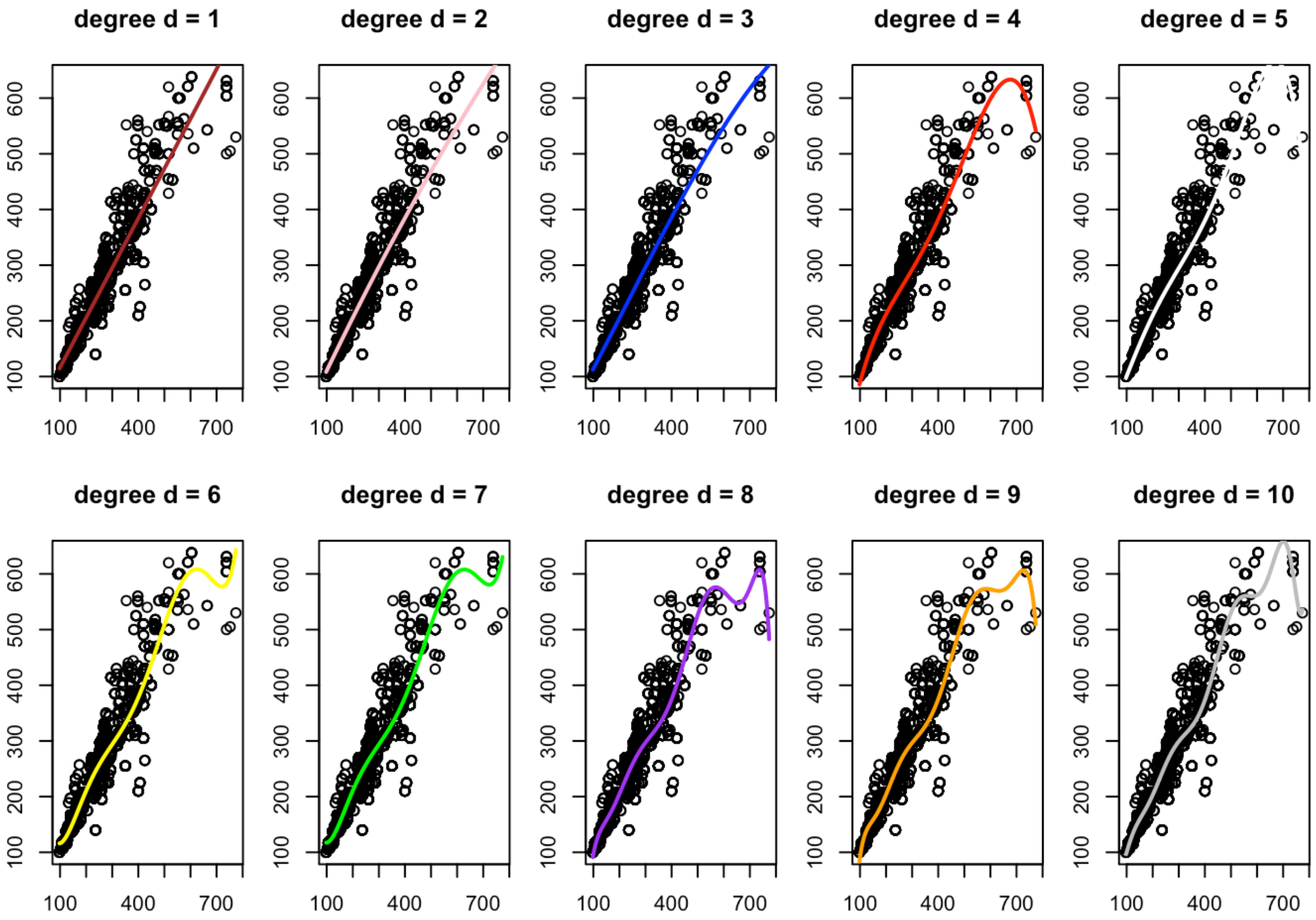


B:

```
cc <- c("Brown", "Pink", "Blue", "Red", "White", "Yellow",
           "Green", "Purple", "Orange", "Gray", "Cyan")
x.grid = seq(min(torque), max(torque))
# for d = 0
ave <- mean(horsepower)
plot(torque, horsepower,
     main="Horsepower as a Function of Torque d=0",
     xlab="Torque",
     ylab="Horsepower")
abline(h=ave, col = "Cyan", lwd = 2)
```



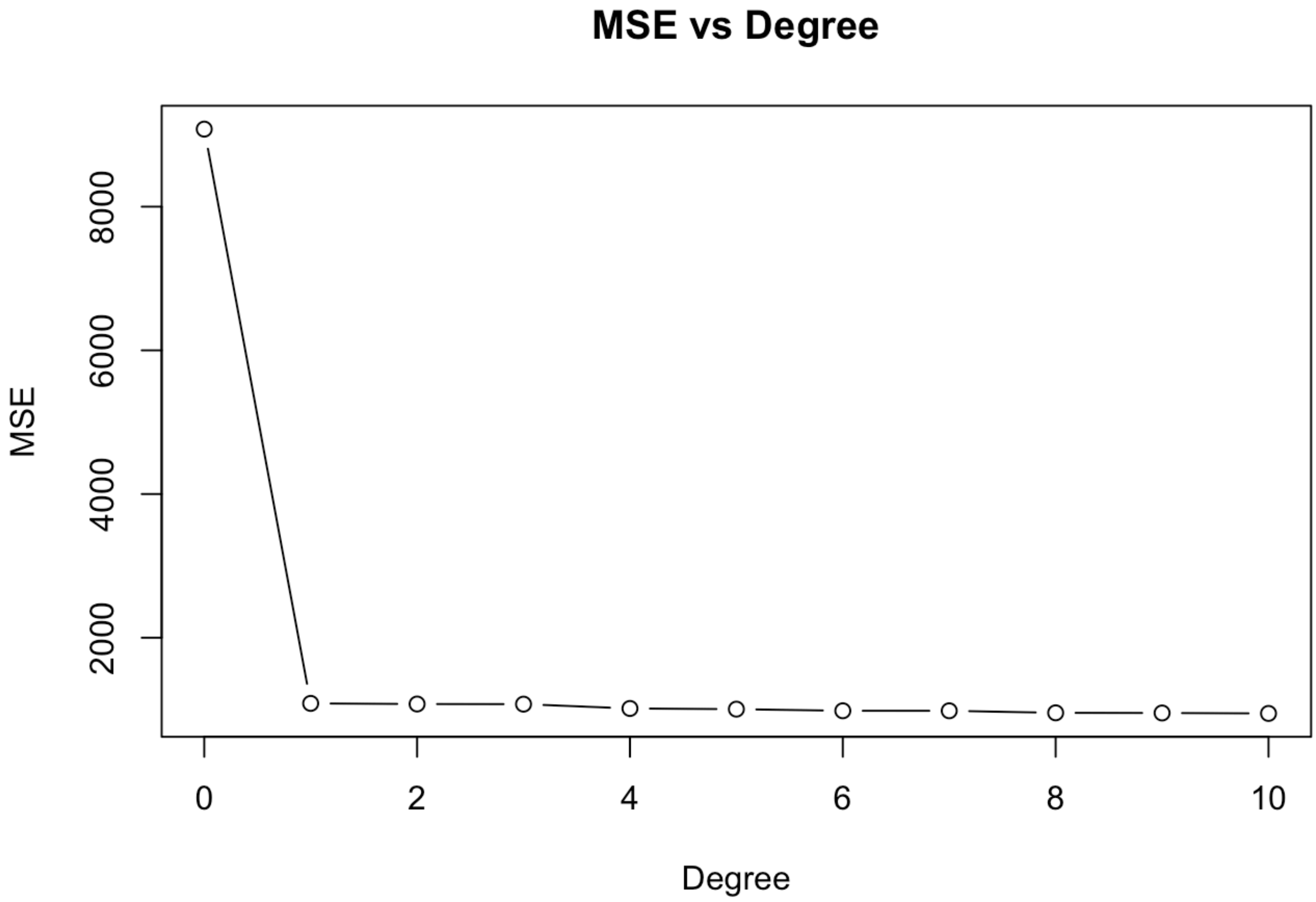
```
par(mfrow = c(2,5), mai = c(0.3, 0.3, 0.5, 0.1))
for (d in 1:10) {
    fit <- lm(horsepower ~ poly(torque, d))
    fhat <- predict(fit, data.frame(torque = x.grid))
    plot(torque, horsepower, main = sprintf("degree d = %i", d), xlab = "", ylab = ""
)
    lines(x.grid, fhat, col = cc[d], lwd = 2)
    }
```



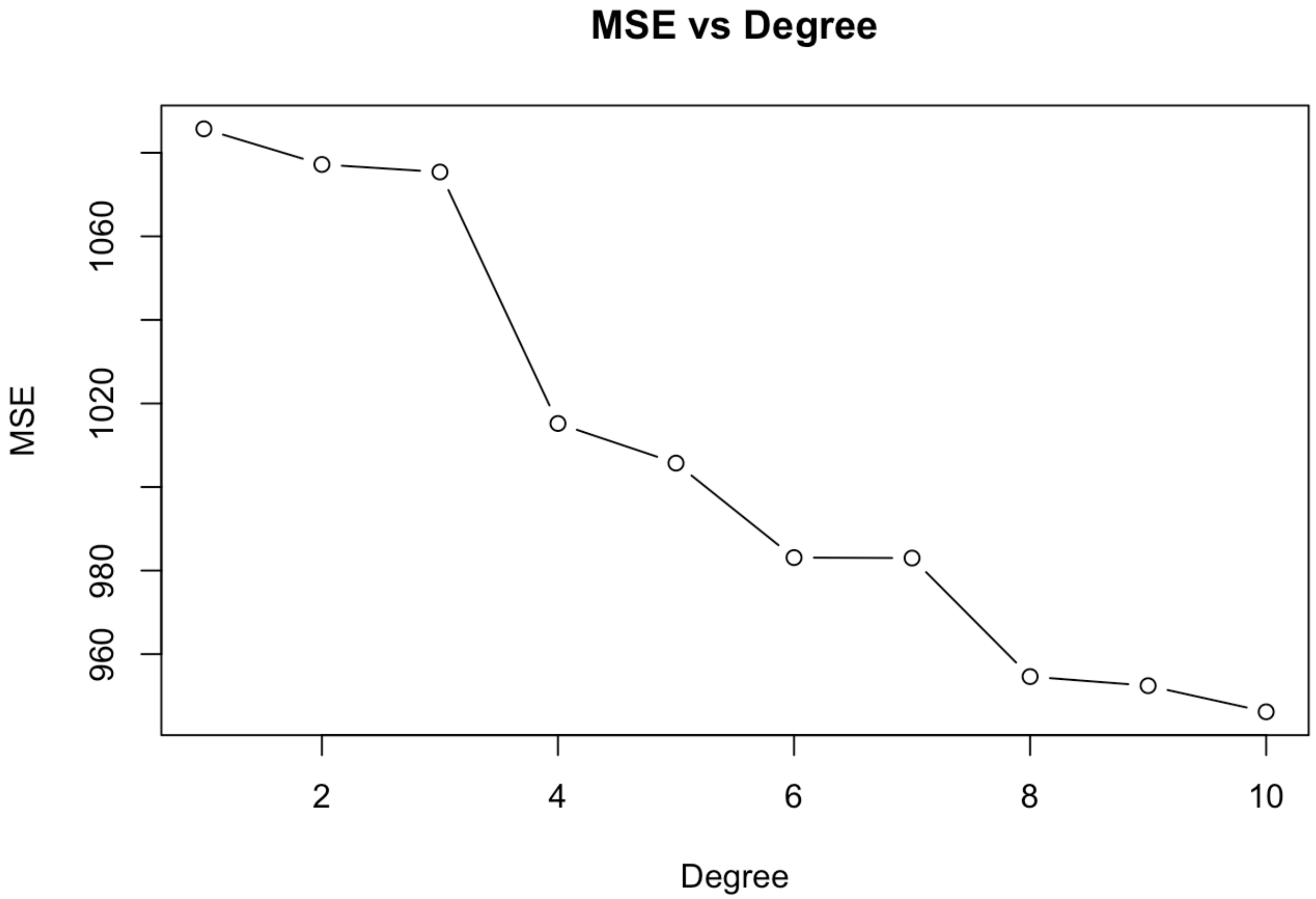
Based on above plot, I would choose d = 5 as it has the best balanced fit in my opinion. I think degree less than 5 are underfit while degree more than 5 are overfit.

C.

```r
# for d = 0
mse <- list()
ave <- mean(horsepower)
mse[1] <- mean((horsepower-ave)^2)

for (d in 1:10) {
    fit <- lm(horsepower ~ poly(torque, d))
    fhat <- predict(fit, data.frame(torque = torque))
    mse[d+1] <- mean((horsepower-fhat)^2)
}
plot(0:10, mse, type = "b", xlab ="Degree", ylab = "MSE", main="MSE vs Degree")
```



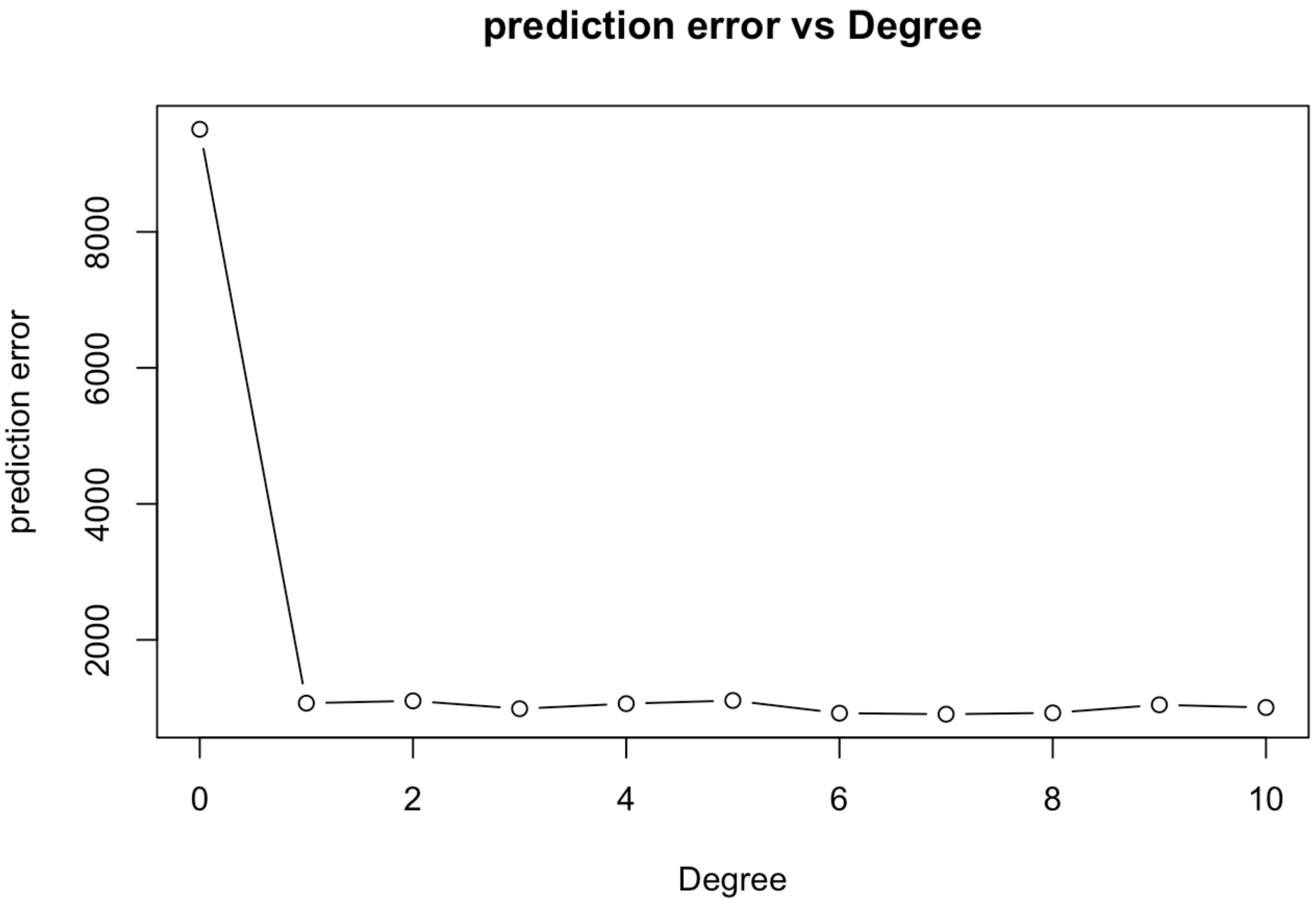
```r
plot(1:10, mse[2:11], type = "b", xlab ="Degree", ylab = "MSE", main="MSE vs Degree")
```



From above plot, since the first one can’t see the difference clearly, so I plot another one with d = 1 to 10. And it is clear that d = 10 has the smallest number of MSE. Hence, I will choose d = 10 based on the plot.

D.

```r
B = 1e3
result <- list()
n <- length(torque)
for(d in 1:11){
  dd <- d - 1
  if (dd == 0){
    pred = numeric(B)
    for(b in 1:B){
        ind = sample(n, replace = TRUE)
        # mean(y_boot)  -> prediction
        y_boot = horsepower[ind]
        ave <- mean(y_boot)
        ind = sample(n, 1)
        y_new = horsepower[ind]
        pred[b] = (y_new - ave)^2
        result[d] <- mean(pred)
    }
  }
  else{
    pred = numeric(B)
    for (b in 1:B){
        # generate bootstrap data and fit model
        ind = sample(n, replace = TRUE)
        x_boot = torque[ind]
        y_boot = horsepower[ind]
        fit = lm(y_boot ~ poly(x_boot, dd))
        # generate bootstrap new observation and compute error
        ind = sample(n, 1)
        x_new = torque[ind]
        y_new = horsepower[ind]
        pred_new = predict(fit, data.frame(x_boot = x_new))
        pred[b] = (y_new - pred_new)^2
     }
    result[d] <- mean(pred)
  }
}

plot(0:10, result, type = "b", xlab ="Degree", ylab = "prediction error", main="prediction error vs Degree")
```



Since we estimate the prediction error by bootstrap, so the result we have each time is slighly different. We can see there is a trend that there is a high prediction error when d=0, as the degree going up, the value pf prediction error get decreasing.

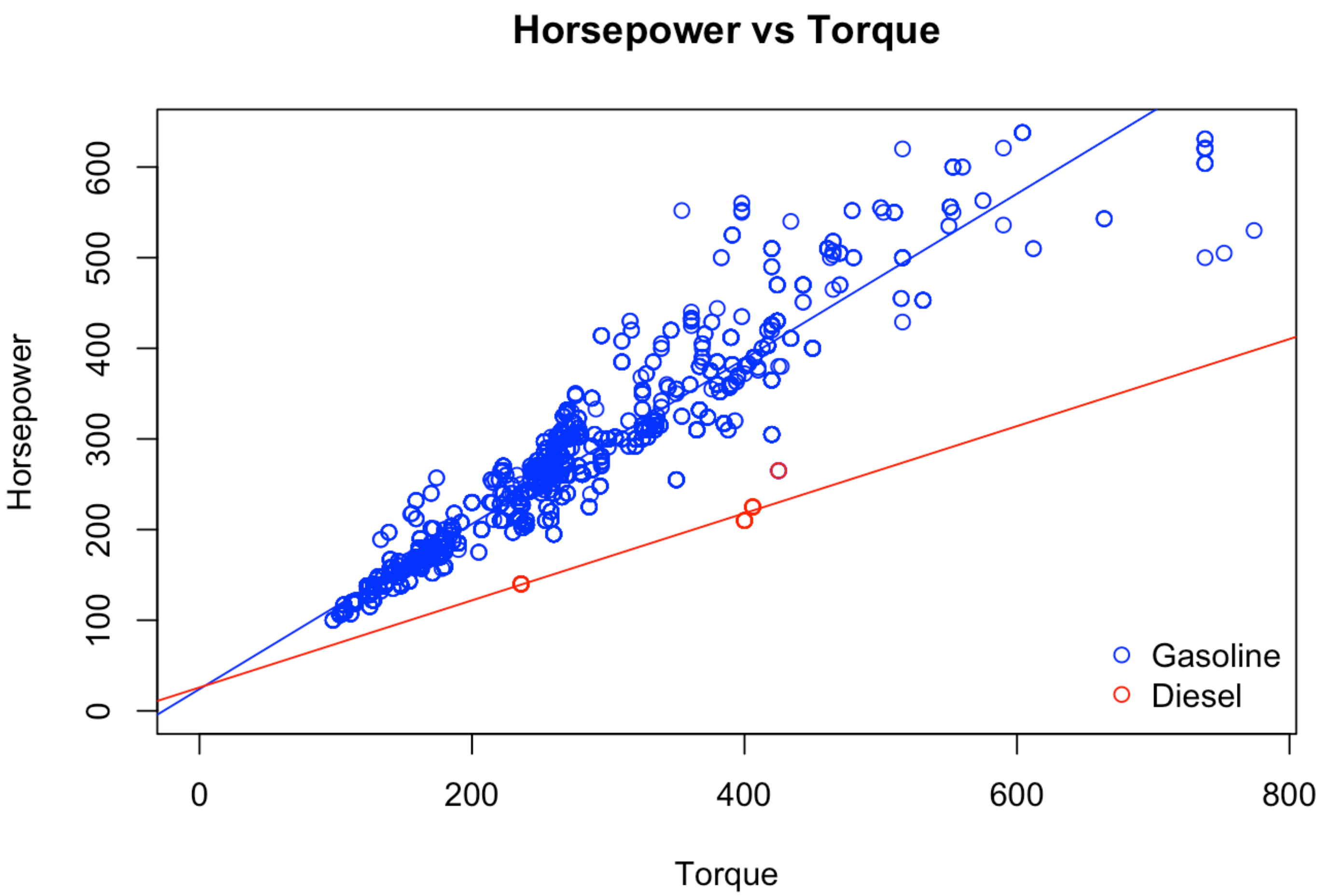
# Problem 2

A. Null hypothesis (H0): There is no difference in linear relationship between horsepower and torque among cars powered by gas and powered by diesel.

Alternative hypothesis (H1): There is a difference in linear relationship between horsepower and torque among cars powerd by gas and powered by diesel.

B.

```
gas <- dataset[dataset$Fuel.Information.Fuel.Type == "Gasoline",]

diesel <- dataset[dataset$Fuel.Information.Fuel.Type == "Diesel fuel",]

lm_gas <- lm(Engine.Information.Engine.Statistics.Horsepower ~ Engine.Information.Eng
ine.Statistics.Torque, data = gas)

lm_die <- lm(Engine.Information.Engine.Statistics.Horsepower ~ Engine.Information.Eng
ine.Statistics.Torque, data = diesel)

plot(gas$Engine.Information.Engine.Statistics.Torque, gas$Engine.Information.Engine.S
tatistics.Horsepower,
     xlab = "Torque", ylab = "Horsepower", col = "blue", main = "Horsepower vs Torque
",
     xlim = c(0, max(dataset$Engine.Information.Engine.Statistics.Torque)),
     ylim = c(0, max(dataset$Engine.Information.Engine.Statistics.Horsepower)))

abline(lm_gas, col = "blue")

points(diesel$Engine.Information.Engine.Statistics.Torque, diesel$Engine.Information.
Engine.Statistics.Horsepower,
       col = "red")
abline(lm_die, col = "red")

legend("bottomright", legend = c("Gasoline", "Diesel"), col = c("blue", "red"), pch =
1, bty = "n")
```



C. We can apply correlation test to see the connection between the horsepower and torque for gas and for diesel. If they get a similar p value, that means that they have similar connection/relationship between those 2 factors.

```
pgas = cor.test(x = gas$Engine.Information.Engine.Statistics.Torque, y = gas$Engine.I
nformation.Engine.Statistics.Horsepower)$p.value
pdiesel = cor.test(x = diesel$Engine.Information.Engine.Statistics.Torque, y = diesel
$Engine.Information.Engine.Statistics.Horsepower)$p.value

print(pgas)
```

```
## [1] 0
```

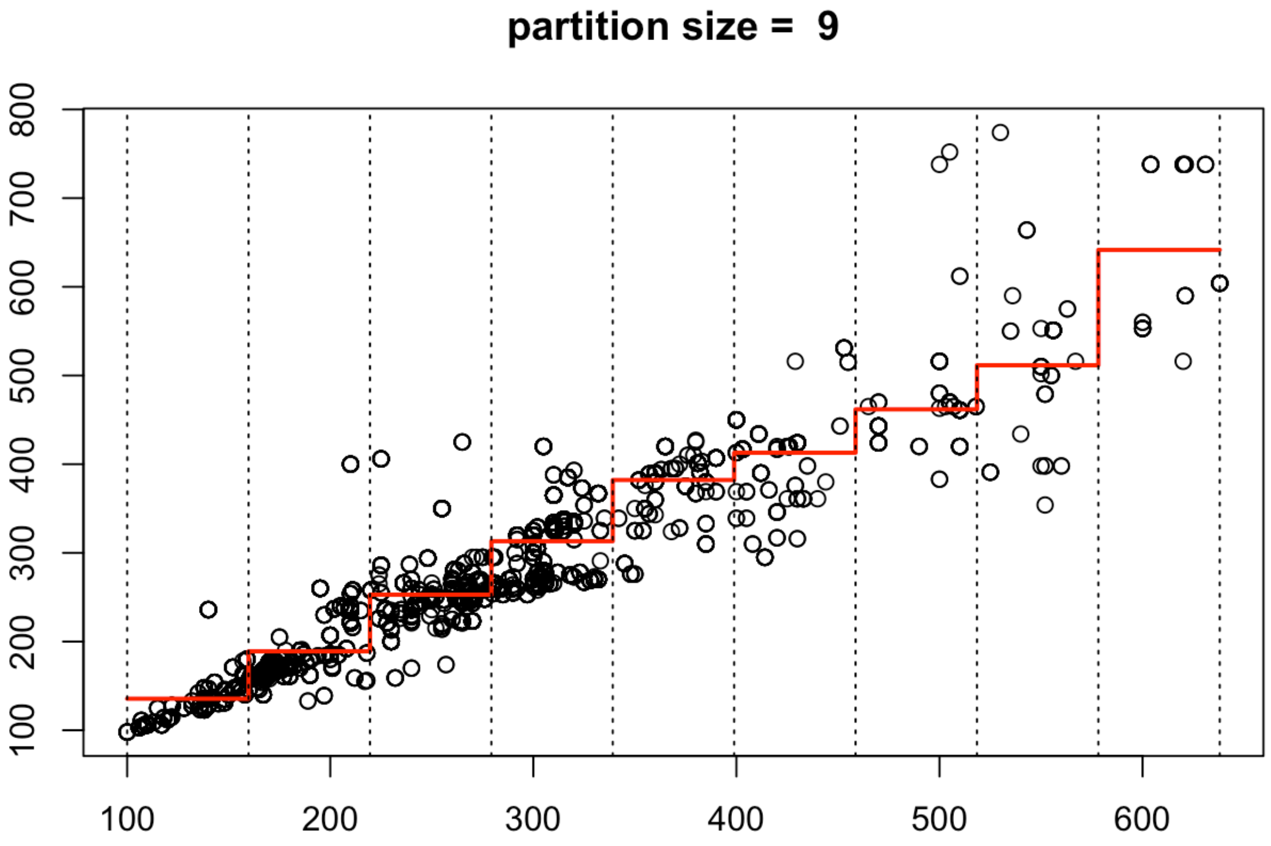
```
print(pdiesel)
```

```
## [1] 4.122982e-18
```

As both p value very similar, this result suggest that there is a closely connection for torque and horsepower under the category gas and category diesel. Hence, here is no difference in linear relationship between horsepower and torque among cars powered by gas and powered by diesel.

# Problem 3

```r
fit_piecewise_constant <- function(x, y, B) {
    calculate_piecewise_constant <- function(x, y, partition) {
        a <- partition
        # piecewise computation in part 9 from le
        m <- length(partition)-1
        t = rep(0, m+1)
        fhat = rep(0, m+1)
        for (j in 1:m){
            I = (a[j] < x)&(x <= a[j+1])
            fit = lm(y[I] ~ 1)
            t[2*j-1] = a[j]
            t[2*j] = a[j+1]
            fhat[2*j-1] = coef(fit)
            fhat[2*j] = coef(fit)
        }
        plot(x, y, main = paste("partition size = ", m), xlab = "", ylab = "")
        lines(t, fhat, col = "red", lwd = 2)
        abline(v = a, lty = 3)
    }

    # Determine optimal partition size
    partition_sizes <- 1:25
    errors <- numeric(length(partition_sizes))

    for (m in partition_sizes) {
        bootstrap_errors <- numeric(B)
        for (b in 1:B) {
            indices <- sample(1:length(x), replace = TRUE)
            x_boot <- x[indices]
            y_boot <- y[indices]

            ind = sample(1:length(x), size = 1)
            x_new = x[ind]
            y_new = y[ind]

            pred_new = mean(y_boot)
            bootstrap_errors[b] = (y_new - pred_new)^2
        }
        errors[m] <- mean(bootstrap_errors)
    }

    best_partition_size <- partition_sizes[which.min(errors)]
    best_partition <- seq(min(x), max(x), length.out = best_partition_size + 1)
    calculate_piecewise_constant(x,y,best_partition)
}
fit_piecewise_constant(horsepower, torque, 1e3)
```


partition size =  9
100
200
300
400
500
600
700
800
100
200
300
400
500
600