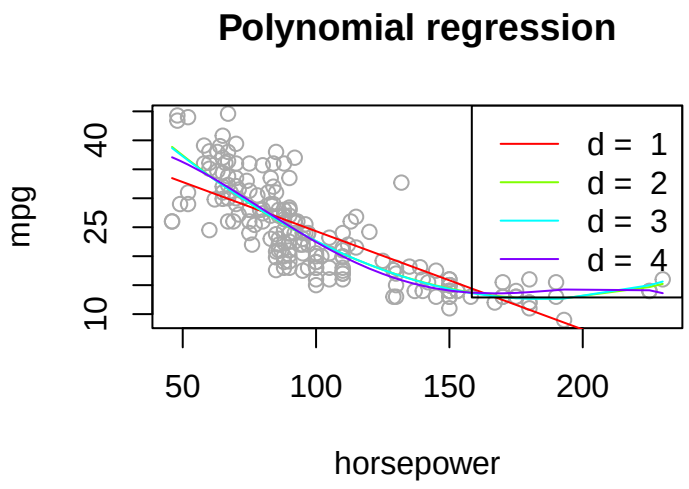
Øving 6

2022-05-30

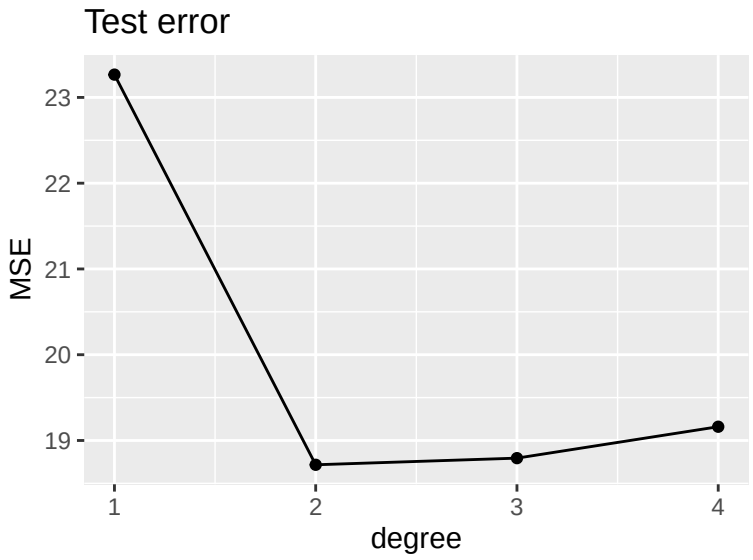
Problem 1

Perform polynomial regression of degree 1,2,3 and 4. Use lines() to add the fitted values in the plot. Also plot the test error depending on polynomial degree.

```
library(ISLR)

ds = Auto[c("horsepower", "mpg")]
n = nrow(ds)

degrees = 1:4

set.seed(1)

train_id = sample.int(n,n/2)

plot(ds[train_id,], col = "darkgrey", main = "Polynomial regression")

co = rainbow(length(degrees))


MSE = sapply(degrees,function(d){
  model = lm(mpg ~ poly(horsepower, degree = d), data = ds[train_id,])
  lines(cbind(ds[train_id,1],model$fitted.values)[order(ds[train_id,1]),], col=co[d])
  mean((predict(model, ds[-train_id,])-ds[-train_id,2])^2)
})

legend("topright", legend = paste("d = ", degrees), lty = 1, col = co)
```




```
MSE = data.frame(MSE = MSE, degree = 1:4)

ggplot(data = MSE, aes(x = degree, y = MSE)) + geom_line() + geom_point() + labs(title = "Test error")
```

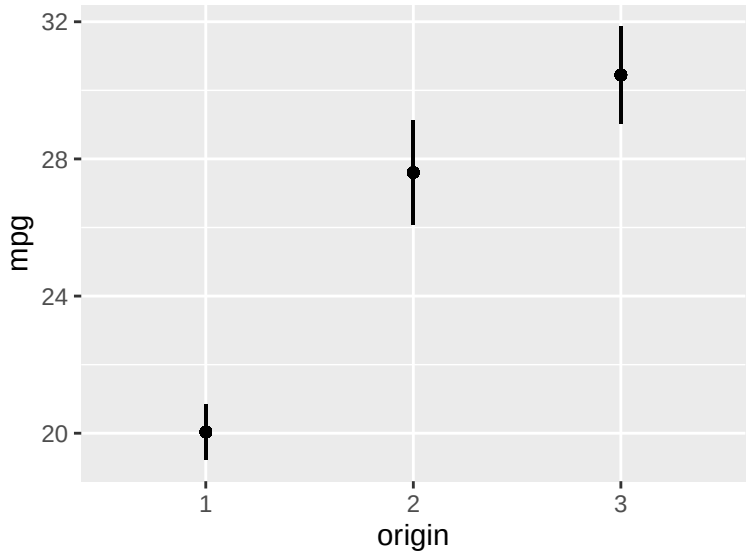


Problem 2

Use factor(origin) for conversion to factor variable for the origin variable. Predict mpg by origin with a linear model. Plot the fitted values and approximate 95% confidence intervals. Selecting se = T in predict() gives standard errors for prediction.

```
l_model = lm(mpg ~ factor(origin), data = Auto)

new_data = data.frame(origin = factor(Auto$origin))

predictions = predict(l_model, new_data, se = T)
```

```
data = data.frame(origin = new_data, mpg = predictions$fit, lwr = predictions$fit - 1.96*predictions$se

ggplot(data, aes(x=origin, y = mpg)) + geom_point() + geom_segment(aes(x=origin, y = lwr, xend = origin
```



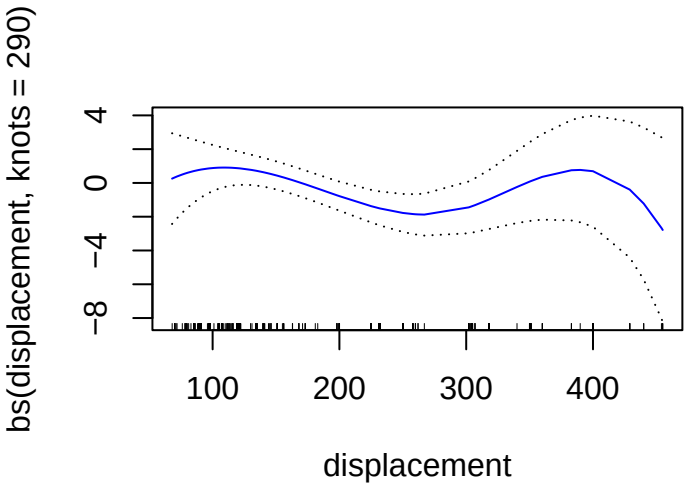
Problem 5

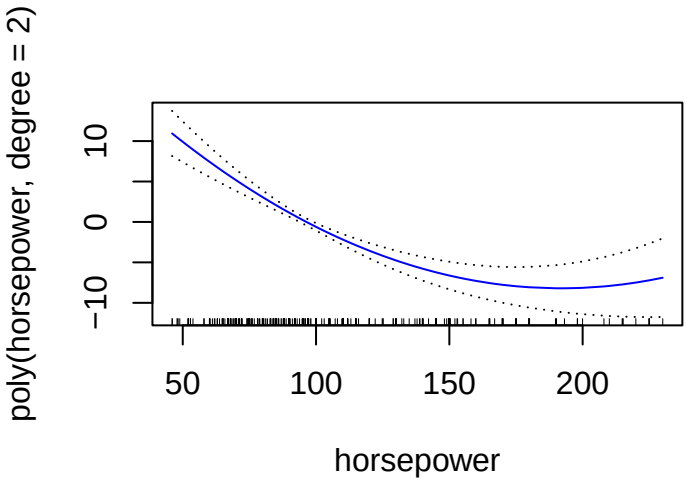
Fit an additive model using the function gam from package gam. Call the result gamobject.

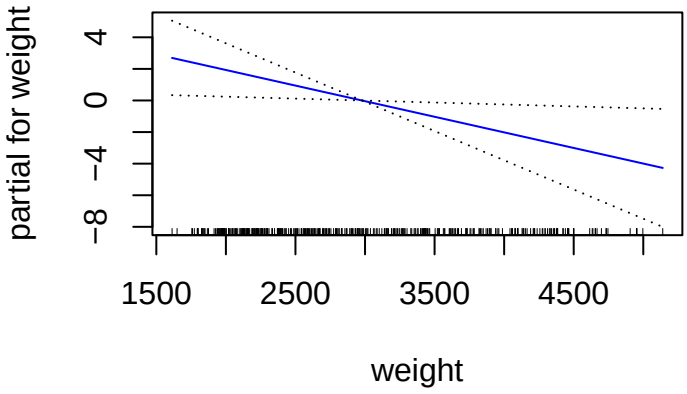
mpg is the response, displace is a cubic spline with knit at 290, horsepower is a polynomial of degree 2, weight is a linear function, acceleration is a smoothing spline with df = 3, origin is a categorical variable.

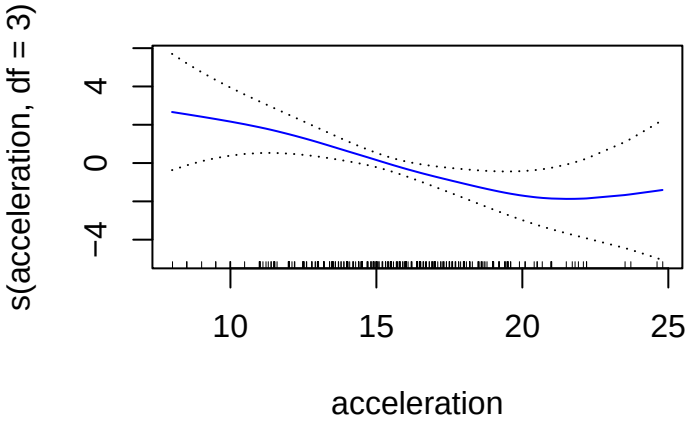
Plot the resulting curves comment on what you see.

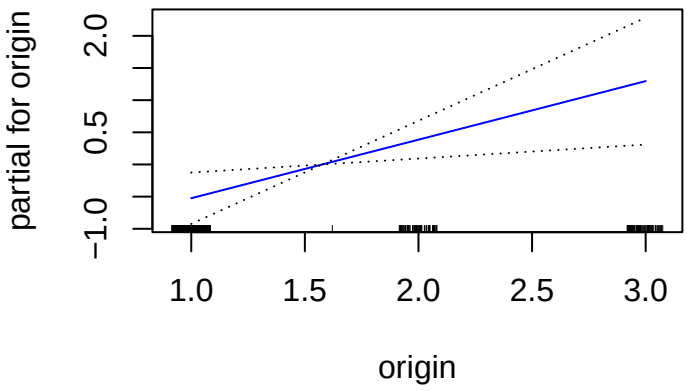
```
library(gam)

gamobject <- gam(mpg  ~ bs(displacement,knots = 290) + poly(horsepower, degree=2) + weight + s(accelera

plot(gamobject, se = TRUE, col = "blue")
```


bs(displacement, knots = 290)
−8
−4
0
4
100
200
300
400
displacement


poly(horsepower, degree = 2)
−10
0
10
50
100
150
200
horsepower


partial for weight
−8
−4
0
4
1500
2500
3500
4500
weight


s(acceleration, df = 3)
−4
0
4
10
15
20
25
acceleration



We observe that displacement has a bigger CI for higher values, the same for the rest of the variables. Horsepower has a low CI for smaller values. weight varies a lot for very small and very high values. Origin has a huge effect on the response and acceleration varies a lot for high and low values.