Lab 2

Daniel Tshiani

2025-05-14

```
library(ggplot2)
library(splines)
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
```

```
load("../data/Auto-3.rda")
```

Exercise 1

a

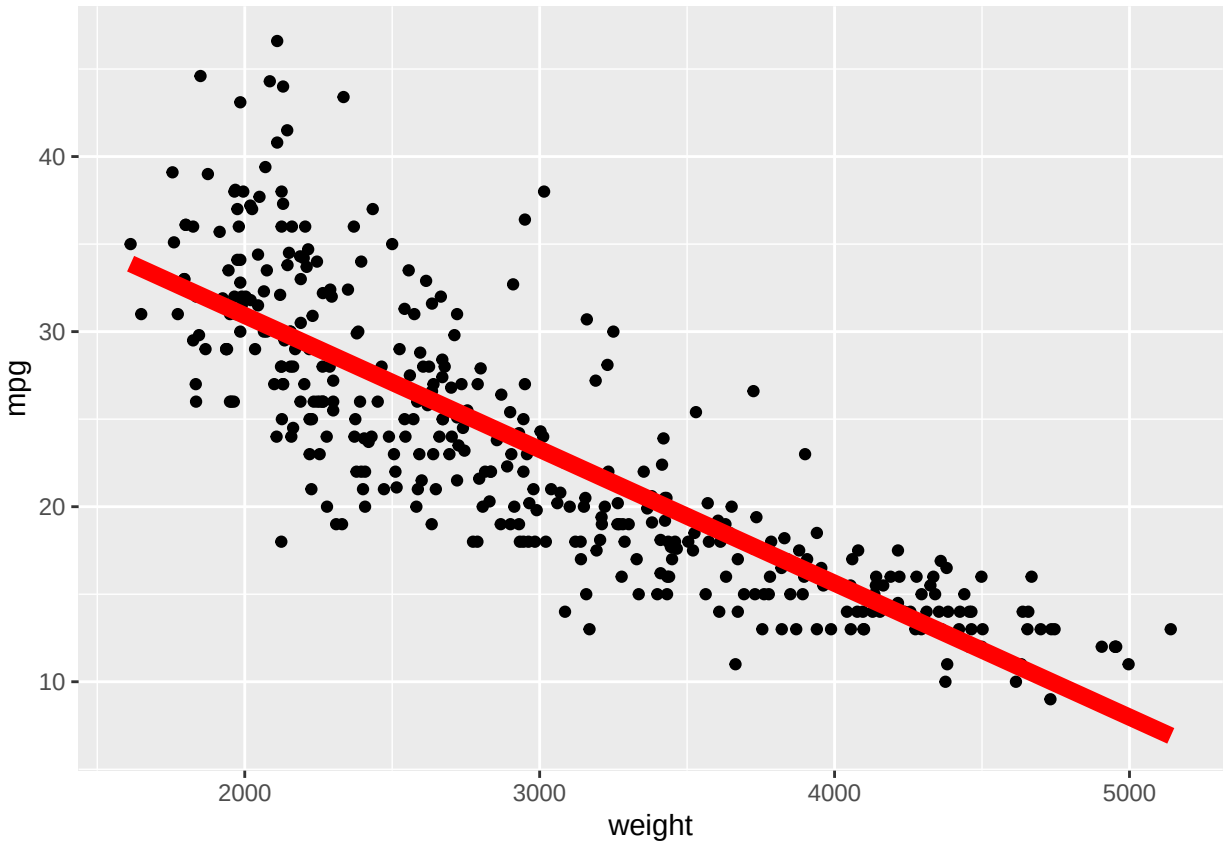
```
model <- lm(mpg ~ weight, data = Auto)
model
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = Auto)
##
## Coefficients:
## (Intercept)       weight
##   46.216525    -0.007647
```

```
ggplot(data = Auto, mapping = aes(x = weight, y = mpg))+
  geom_point()+
  geom_smooth(method = "lm", se = F, color = "red", size = 3)
```

```
## 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.
```

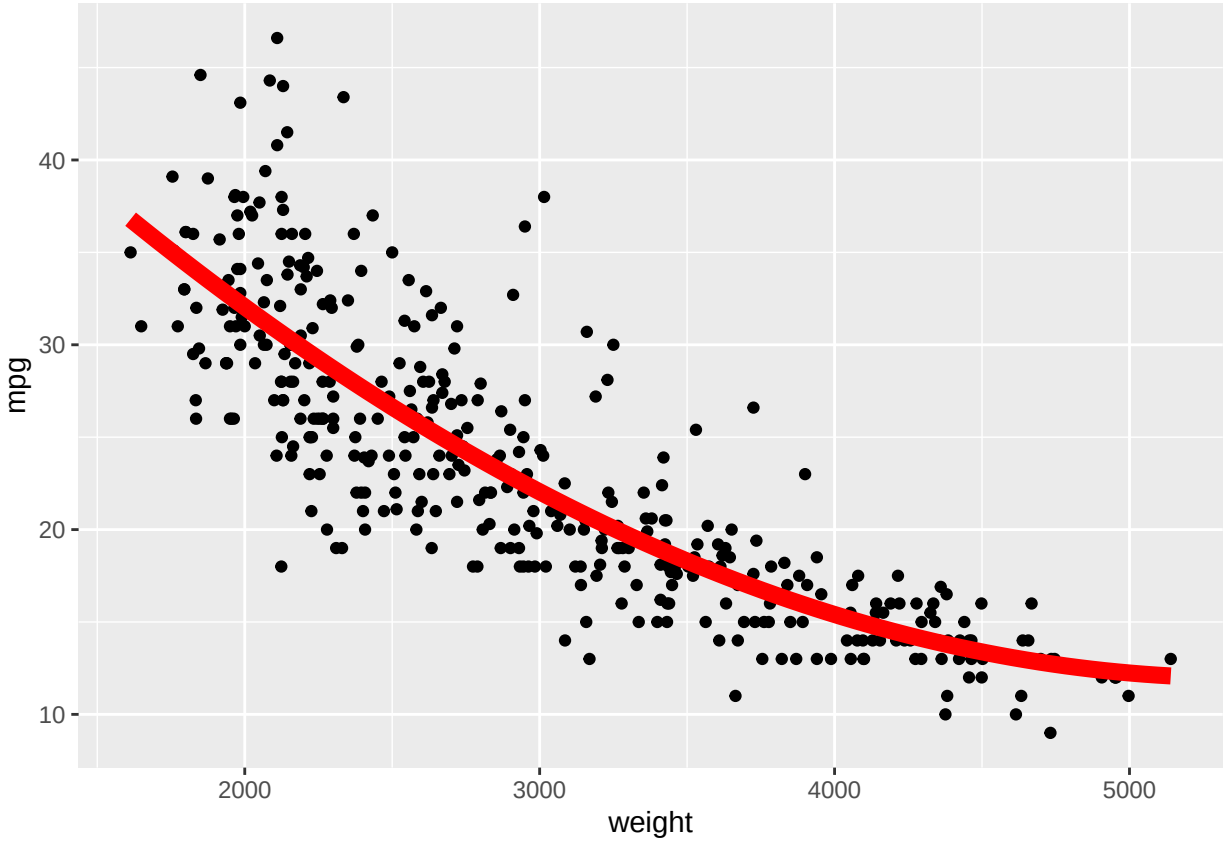
```
## `geom_smooth()` using formula = 'y ~ x'
```



b

```
ggplot(data = Auto, mapping = aes(x = weight, y = mpg))+
  geom_point()+
  geom_smooth(method = "lm", se = F, color = "red", size = 3, formula = y ~ bs(x, df = 2))
```

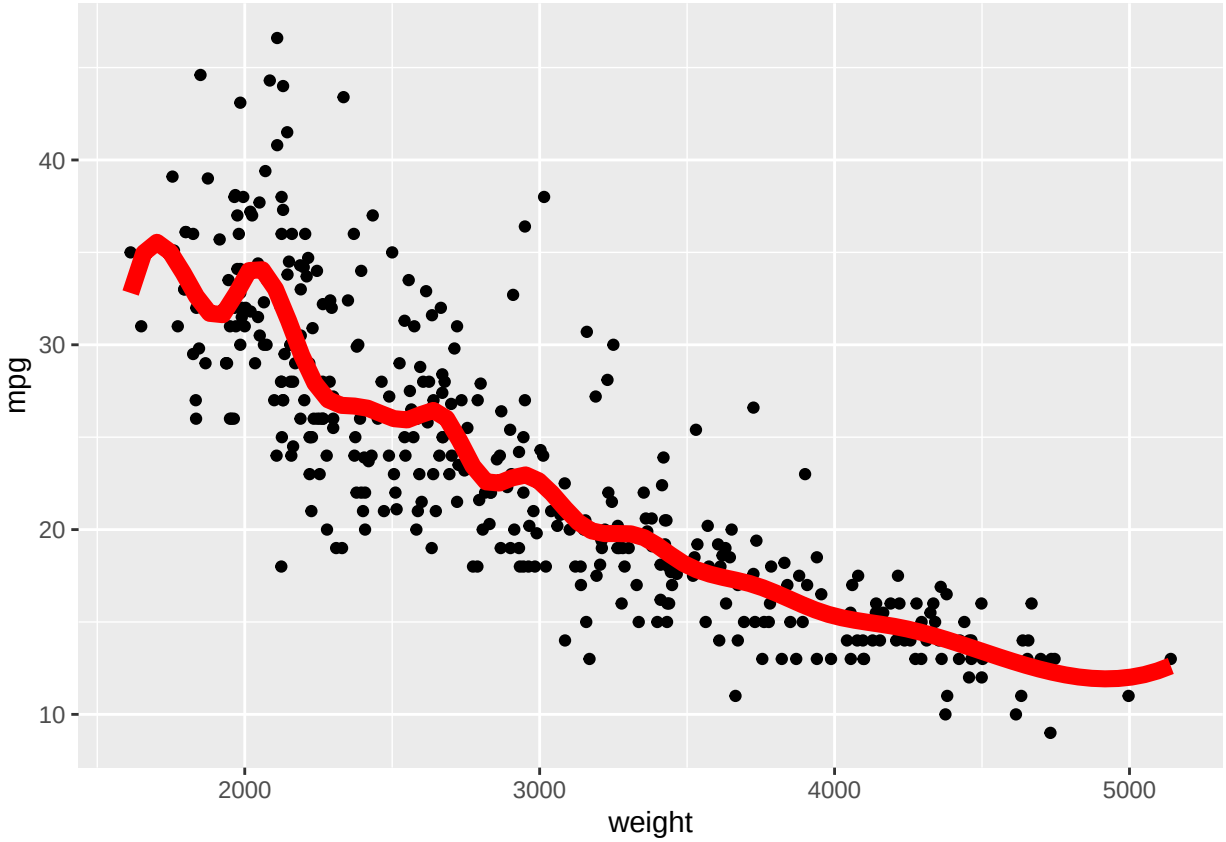
```
## Warning in bs(x, df = 2): 'df' was too small; have used 3
```



this one defaults to 3 splines but i noticed the curve more accurately then the curve in question 1.

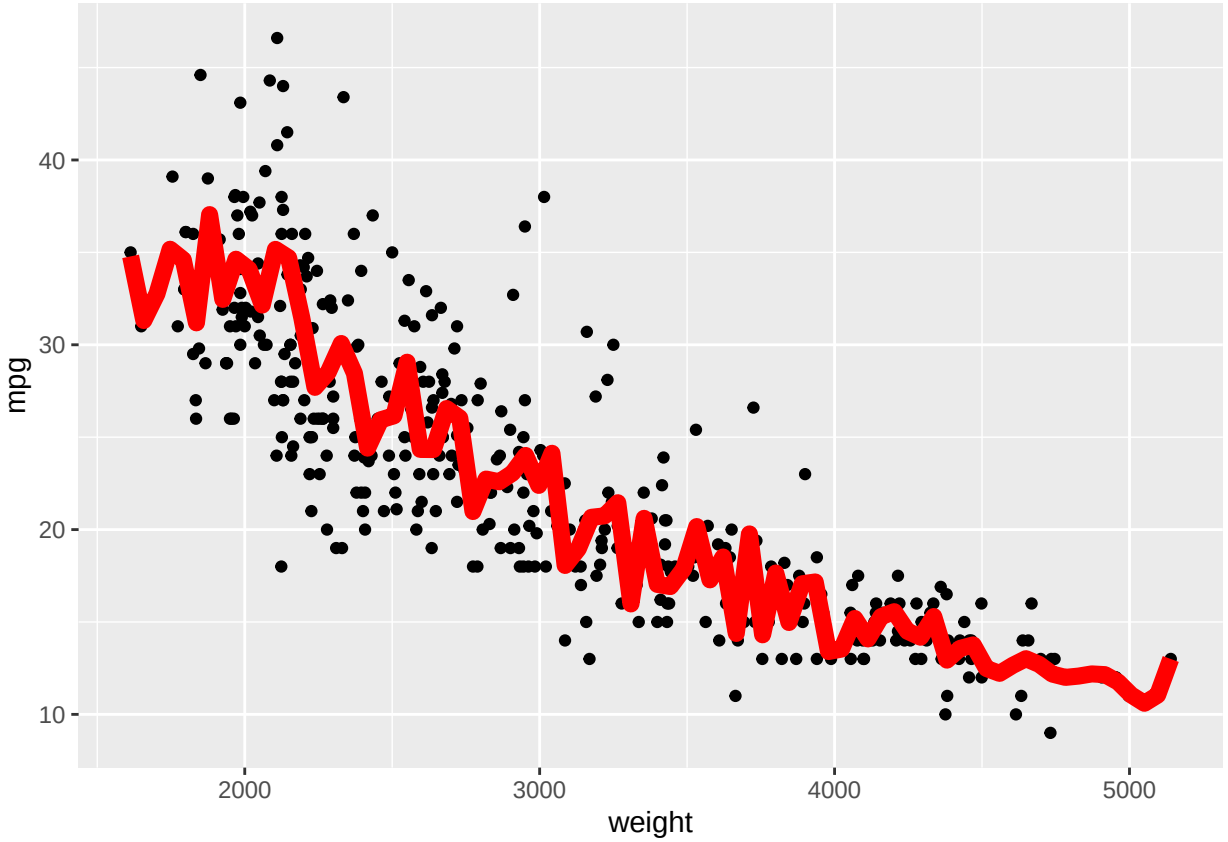
c

```
ggplot(data = Auto, mapping = aes(x = weight, y = mpg))+
  geom_point()+
  geom_smooth(method = "lm", se = F, color = "red", size = 3, formula = y ~ bs(x, df = 20))
```



It looks like we are starting to overfit the model when we fit a spline with 20 DF.

```
ggplot(data = Auto, mapping = aes(x = weight, y = mpg))+
  geom_point()+
  geom_smooth(method = "lm", se = F, color = "red", size = 3, formula = y ~ bs(x, df = 100))
```



I would say with 100 DF the model is way over fitted. And this would be a training sample dataset so the model would follow the training dataset closely and not accurately predict the testing dataset.

d

the last spline produced is flexible due to the high number of DF. I would say yes, it matches the training data well. I don't think this model would be powerful for prediction because its too focused on the training data and it wouldnt be as focused on the testing data.

Exercise 2

##a

```
training <- Auto %>%
  mutate(n = row_number())%>%
  filter(n <= max(row_number()/2))

testing <- Auto %>%
  mutate(n = row_number())%>%
  filter(n <= max(row_number()/2))
```

##b

```
spline_model <- lm(mpg ~ bs(weight, df = 5), data = training)

predictions <- predict(spline_model, newdata = testing)
mse <- mean((testing$mpg - predictions)^2)
print(mse)
```

```
## [1] 4.695487
```

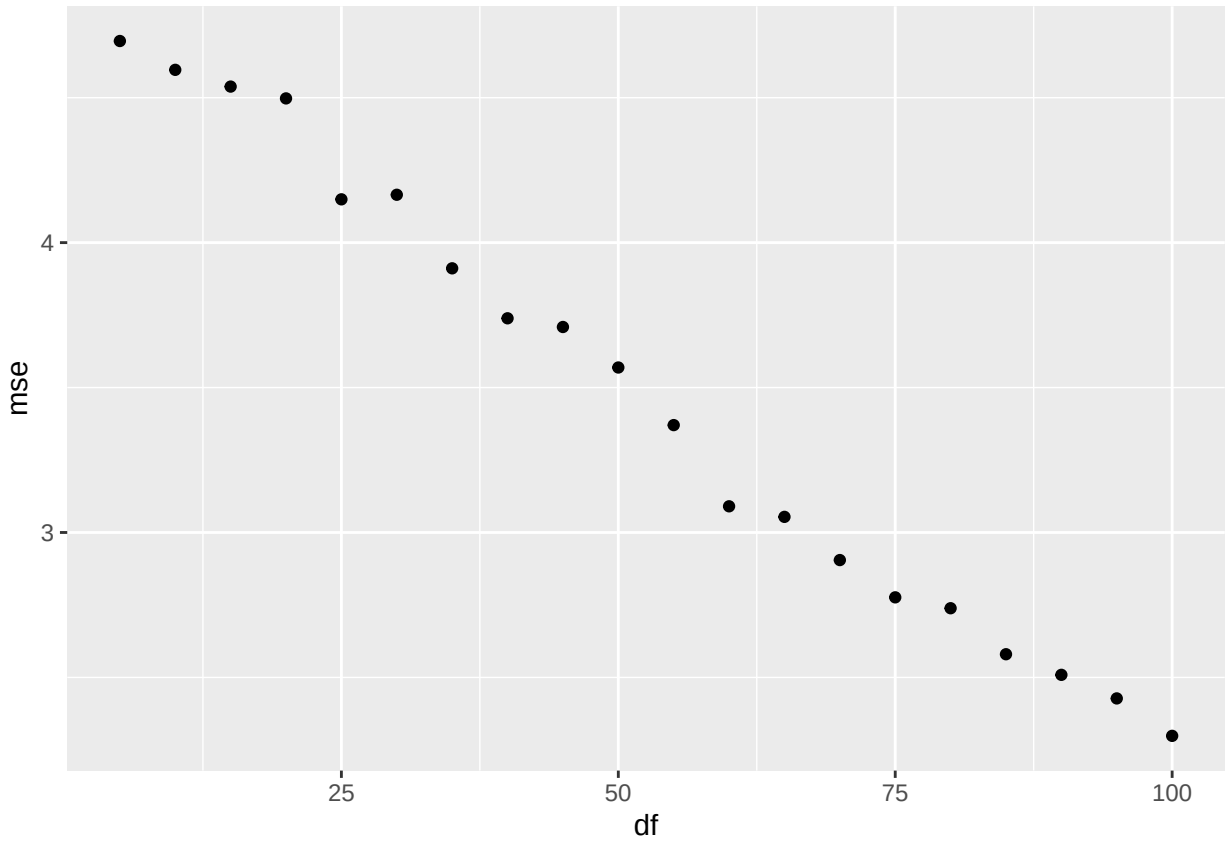
c

```
splines <- seq(5,100,5)
mse_values <- numeric(length(splines))

for(i in seq_along(splines)) {
  df <- splines[i]

  spline_model <- lm(mpg ~ bs(weight, df = df), data = training)
  predictions <- predict(spline_model, newdata = testing)
  mse_values[i] <- mean((testing$mpg - predictions)^2)
}
  results <- data.frame(df = splines, mse = mse_values)

  head(results,10)
```

```
##    df      mse
## 1   5 4.695487
## 2  10 4.596065
## 3  15 4.538325
## 4  20 4.497503
## 5  25 4.149309
## 6  30 4.165096
## 7  35 3.911352
## 8  40 3.738895
## 9  45 3.708906
## 10 50 3.569224
```

d

```
ggplot(data = results, mapping = aes(x = df, y = mse))+
  geom_point()
```


mse
4
3
25
50
75
100
df