Exercise 8 (2025) — Advanced Methods for Regression and Classification

Olesia Galynskaia 12321492

2025-12-01

Loading and observing data

```
str(cars)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
##  $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

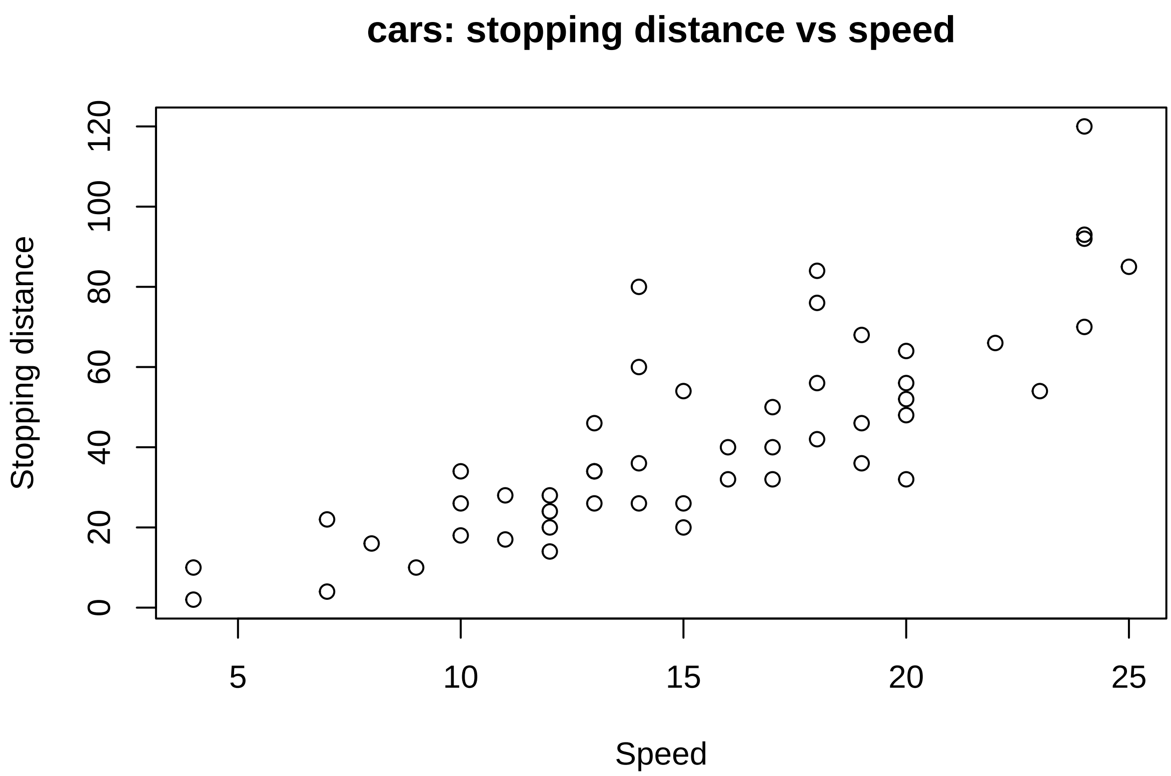
```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

```
cor(cars$speed, cars$dist)
```

```
## [1] 0.8068949
```

```
plot(dist ~ speed, data = cars,
     main = "cars: stopping distance vs speed",
     xlab = "Speed",
     ylab = "Stopping distance")
```



Train/test split

```
n <- nrow(cars)
n_train <- floor(2 * n / 3)
train_idx <- sample(1:n, size = n_train)

cars_train <- cars[train_idx, ]
cars_test  <- cars[-train_idx, ]

nrow(cars_train)
```

```
## [1] 33
```

```
nrow(cars_test)
```

```
## [1] 17
```

```
head(cars_train)
```

```
##    speed dist
## 16    13   26
## 29    17   32
## 9     10   34
## 17    13   34
## 22    14   60
## 19    13   46
```

Comment

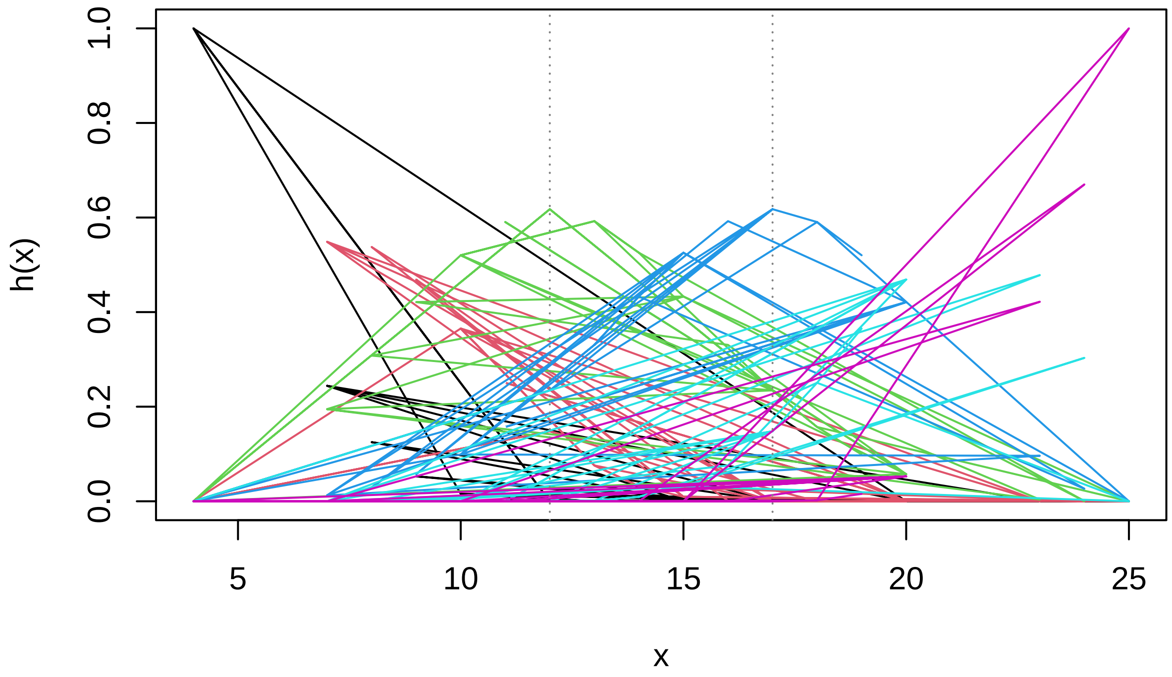
The plot shows a clear upward trend: higher speed comes with longer stopping distance.
The correlation is about 0.81, so the relationship is strong.
The spread gets wider at higher speeds, but the overall pattern stays the same.

Function lecturespl()

```r
## spline basis constructor used in the lecture
lecturespl <- function(x, q = 2, M = 4) {
  ## ensure numeric vector
  x <- as.numeric(x)

  ## internal knots chosen as empirical quantiles
  xquantiles <- as.numeric(
    quantile(x, probs = seq(0, 1, length.out = q + 2))
  )[2:(q + 1)]   # drop 0% and 100%

  ## build B-spline basis of order M (degree = M - 1)
  X <- splines::bs(
    x,
    degree    = M - 1,
    knots     = xquantiles,
    intercept = TRUE
  )

  ## return list in the format expected by plotspl()
  list(
    x          = x,
    X          = X,
    xquantiles = xquantiles
  )
}
```

Ex-1 Basis functions

```
plotspl <- function(splobj, ...) {
  matplot(splobj$x, splobj$X, type = "l", lty = 1,
          xlab = "x", ylab = "h(x)", ...)
  abline(v = splobj$xquantiles, lty = 3, col = gray(0.5))
}

# construct spline basis on the training set
spl_train <- lecturespl(
  x = cars_train$speed,
  q = 2,       # number of knots
  M = 4        # spline order
)

# visualize basis functions
plotspl(spl_train)
```



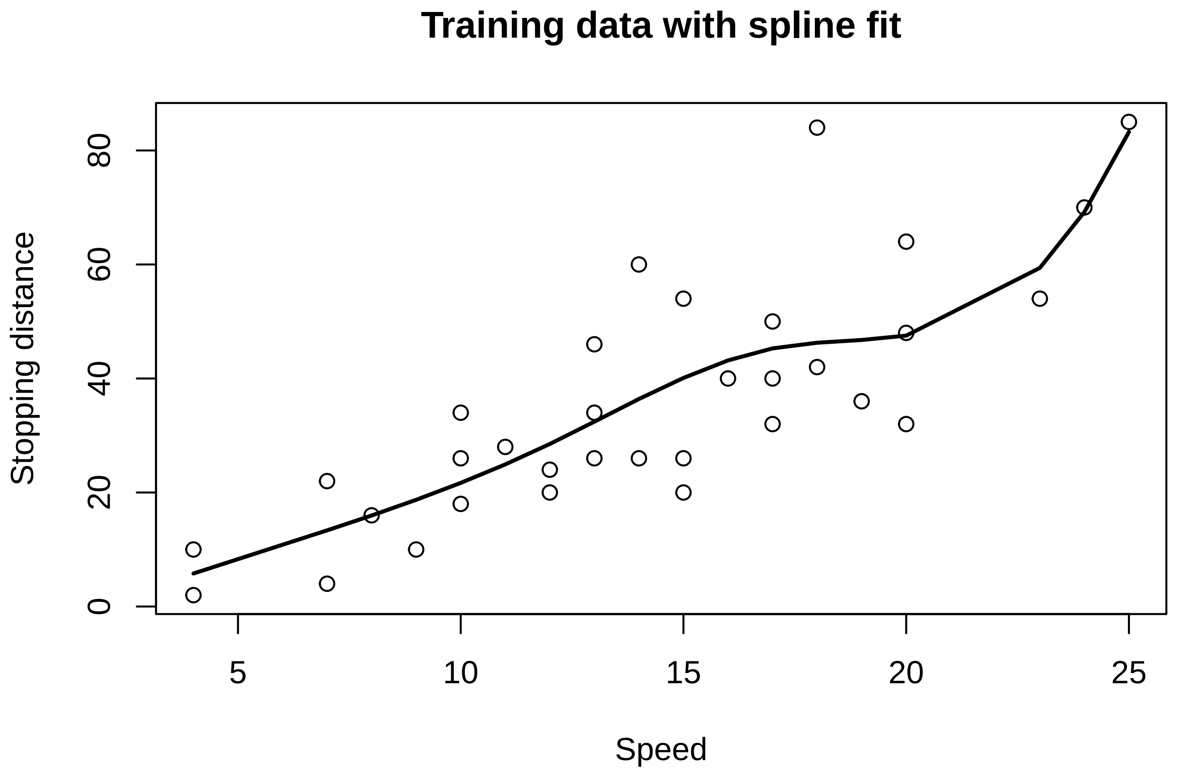
Comment

The plot shows all spline basis functions evaluated at the training x-values.
Each colored line is one basis function. The dotted vertical lines mark the knots.
The basis functions overlap and peak in different regions, which is exactly what splines are supposed to do.

The functions rise and fall smoothly around the knots, they overlap in the right regions, and none of them do anything weird like exploding or staying flat.
This means the spline construction worked, the knots were placed correctly, and the model will have enough flexibility to capture curvature in the speed-distance relationship.

Ex-2 Predict the training data of the response

```r
# spline basis
X_train <- spl_train$X

# fit linear model on the training data using the spline basis
fit_spl <- lm(cars_train$dist ~ X_train)

# predictions on the training set
yhat_train <- predict(fit_spl)

# sort training data by speed for a smooth prediction curve
ord <- order(cars_train$speed)
x_sorted    <- cars_train$speed[ord]
y_sorted    <- cars_train$dist[ord]
yhat_sorted <- yhat_train[ord]

# plot training data and spline fit
plot(cars_train$speed, cars_train$dist,
     xlab = "Speed",
     ylab = "Stopping distance",
     main = "Training data with spline fit")
lines(x_sorted, yhat_sorted, lwd = 2)
```



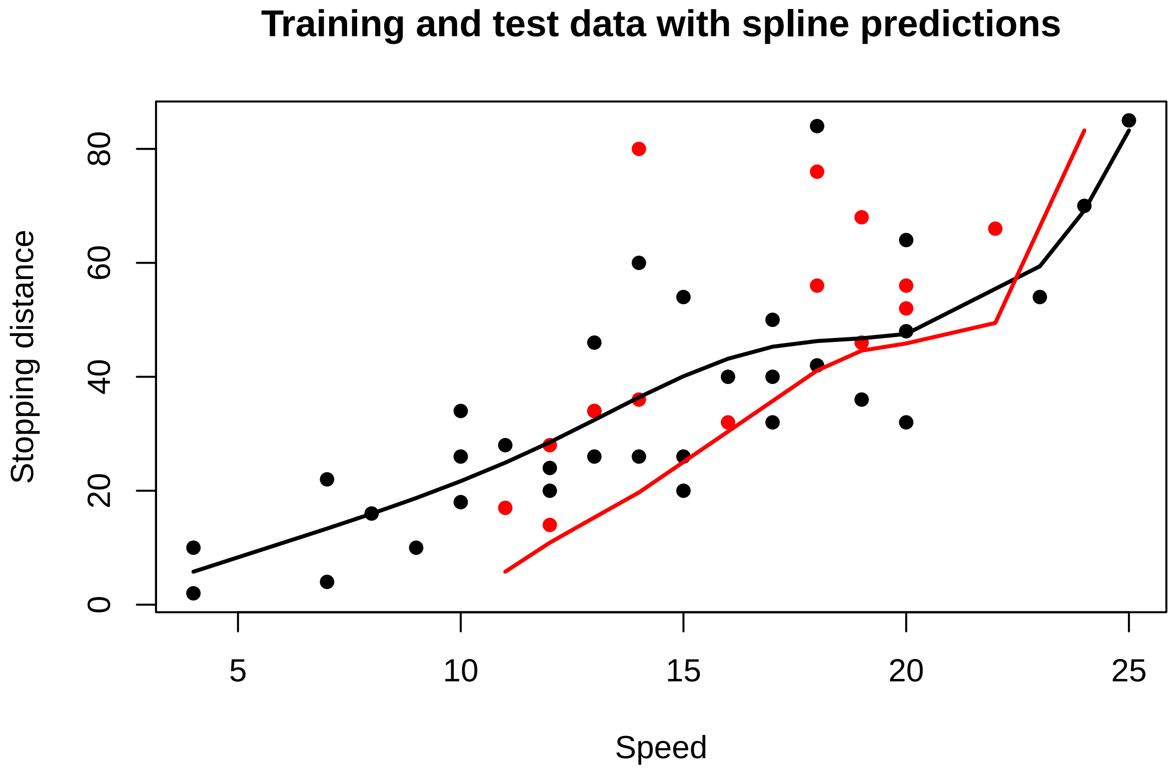
Comment

The spline fit forms a smooth curve that follows the general rise in stopping distance as speed increases.
It bends slightly where the data change direction and stays close to the main trend without trying to chase every single point.
The jump near higher speeds comes from the data themselves, not from the model.
Overall, the predictions look smooth and reasonable for this dataset.

Ex-3 Predict the response from the test set observations

```
# build the same spline basis for the test set
spl_test <- lecturespl(
  x = cars_test$speed,
  q = 2,
  M = 4
)

# predictions for test set
yhat_test <- predict(fit_spl, newdata = list(X_train = spl_test$X))
```

```r
# sort both sets for smooth curves
ord_train <- order(cars_train$speed)
ord_test  <- order(cars_test$speed)

# plot training and test points
plot(cars_train$speed, cars_train$dist,
     col = "black", pch = 16,
     xlab = "Speed",
     ylab = "Stopping distance",
     main = "Training and test data with spline predictions")
points(cars_test$speed, cars_test$dist,
       col = "red", pch = 16)

# add prediction curves
lines(cars_train$speed[ord_train],
      yhat_train[ord_train], col = "black", lwd = 2)
lines(cars_test$speed[ord_test],
      yhat_test[ord_test], col = "red", lwd = 2)
```

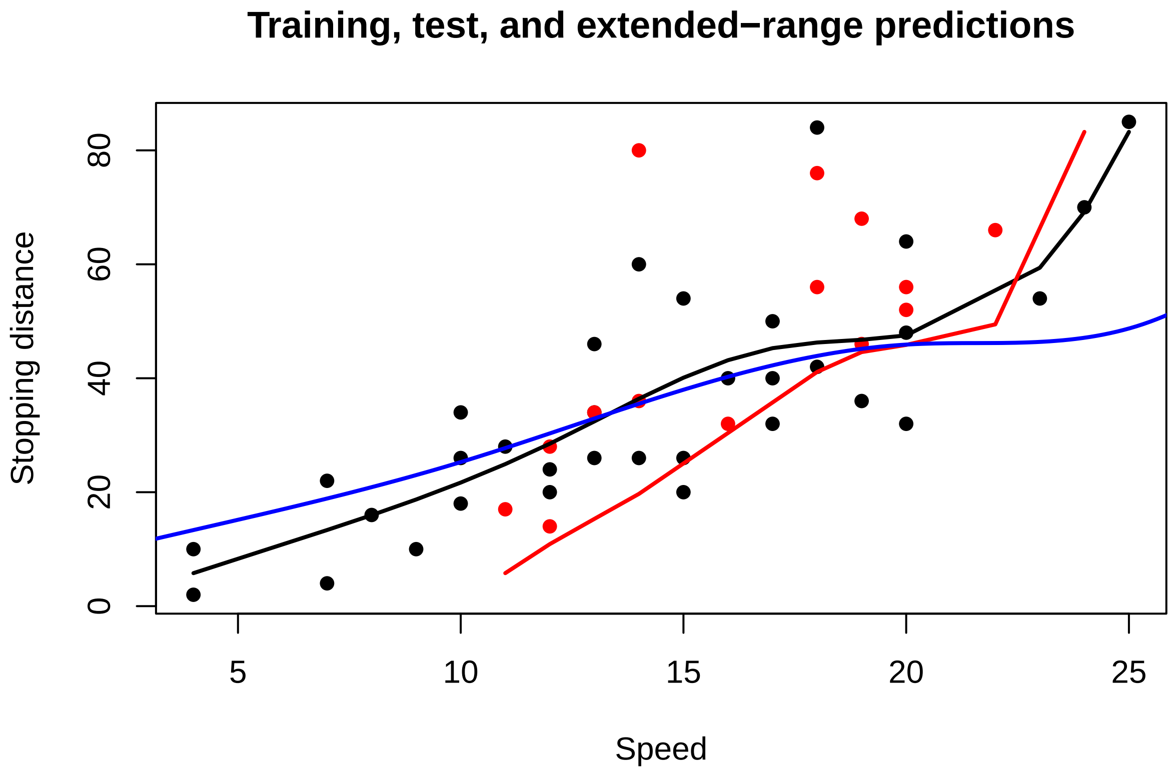


Comment

The black points and curve show the training data and its predictions.
The red points and curve show the test set.
Both curves follow nearly the same shape: they rise smoothly with speed and change in the same places.
The small gap between the black and red lines is normal, since the test set is just a different sample.
Overall, the model behaves consistently on both sets, which means the spline fit is working as expected.

Ex-4 seq(0,30)

```r
# create new speed values (extended range)
new_x <- seq(0, 30, length.out = 200)

# build spline basis for the new x using the same knots and order
spl_new <- lecturespl(
  x = new_x,
  q = 2,
  M = 4
)

# predictions on extended range
yhat_new <- predict(fit_spl, newdata = list(X_train = spl_new$X))

# plot everything together
plot(cars_train$speed, cars_train$dist,
     col = "black", pch = 16,
     xlab = "Speed",
     ylab = "Stopping distance",
     main = "Training, test, and extended-range predictions")

points(cars_test$speed, cars_test$dist,
       col = "red", pch = 16)

# lines for train and test predictions (as before)
lines(cars_train$speed[ord_train],
      yhat_train[ord_train], col = "black", lwd = 2)
lines(cars_test$speed[ord_test],
      yhat_test[ord_test], col = "red", lwd = 2)

# add extended prediction curve
lines(new_x, yhat_new, col = "blue", lwd = 2)
```



Comment

The blue line shows the model's predictions for speeds outside the original data range.
It stays smooth and keeps the same general trend as in the training region.
Once the curve moves far away from the observed speeds, it becomes flatter and then slightly bends, which is typical when a spline has no data to guide it.
Overall, the extended predictions look reasonable and behave consistently with the fitted model.

Ex-5 modifying

new lecturespl()

I added the option to pass fixed knots, because test and new x-values must use the same knots as the training set.
If xquantiles is provided, the function uses them instead of recomputing knots.

```r
# modified spline basis constructor
lecturespl <- function(x, q = 2, M = 4, xquantiles = NULL) {
  # ensure numeric vector
  x <- as.numeric(x)
  # if no knots are supplied, compute them from x (training case)
  if (is.null(xquantiles)) {
    xquantiles <- as.numeric(
      quantile(x, probs = seq(0, 1, length.out = q + 2))
    )[2:(q + 1)]  # drop 0% and 100%
  }

  # build B-spline basis with given knots and order
  X <- splines::bs(
    x,
    degree   = M - 1,
    knots    = xquantiles,
    intercept = TRUE
  )

  # return list in the format expected by plotspl()
  list(
    x          = x,
    X          = X,
    xquantiles = xquantiles
  )
}
```

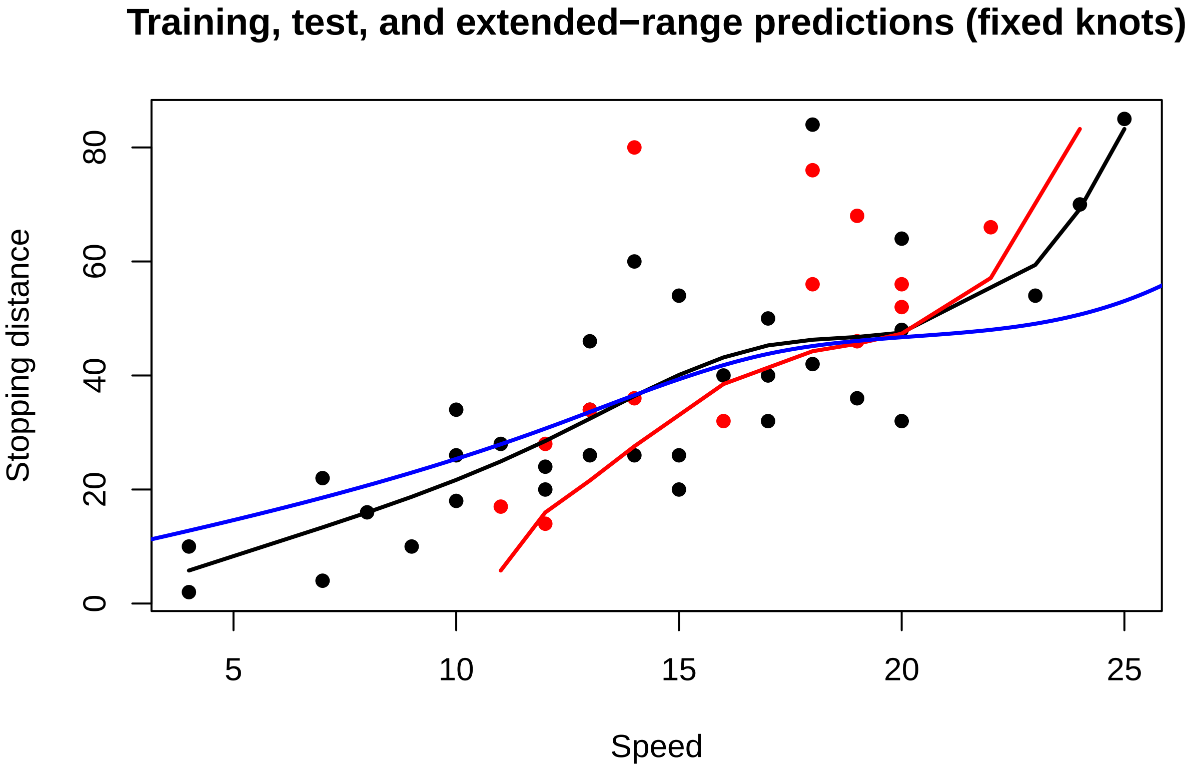
new basis and model

```r
# training basis (knots computed from training x)
spl_train <- lecturespl(cars_train$speed, q = 2, M = 4)
X_train   <- spl_train$X

# fit model on training data
fit_spl <- lm(cars_train$dist ~ X_train)

# predictions on training set
yhat_train <- predict(fit_spl)

# test basis using the SAME knots as in training
spl_test <- lecturespl(
  x          = cars_test$speed,
  q          = 2,
  M          = 4,
  xquantiles = spl_train$xquantiles
)
X_test   <- spl_test$X
yhat_test <- predict(fit_spl, newdata = list(X_train = X_test))
```

```r
# extended x-range
new_x <- seq(0, 30, length.out = 200)

spl_new <- lecturespl(
  x          = new_x,
  q          = 2,
  M          = 4,
  xquantiles = spl_train$xquantiles
)
X_new    <- spl_new$X
yhat_new <- predict(fit_spl, newdata = list(X_train = X_new))
```

new plot

```r
# order indices for smooth curves
ord_train <- order(cars_train$speed)
ord_test  <- order(cars_test$speed)

plot(cars_train$speed, cars_train$dist,
     col = "black", pch = 16,
     xlab = "Speed",
     ylab = "Stopping distance",
     main = "Training, test, and extended-range predictions (fixed knots)")

points(cars_test$speed, cars_test$dist,
       col = "red", pch = 16)

# training and test prediction curves
lines(cars_train$speed[ord_train],
      yhat_train[ord_train], col = "black", lwd = 2)
lines(cars_test$speed[ord_test],
      yhat_test[ord_test], col = "red", lwd = 2)

# extended-range predictions
lines(new_x, yhat_new, col = "blue", lwd = 2)
```



Comment

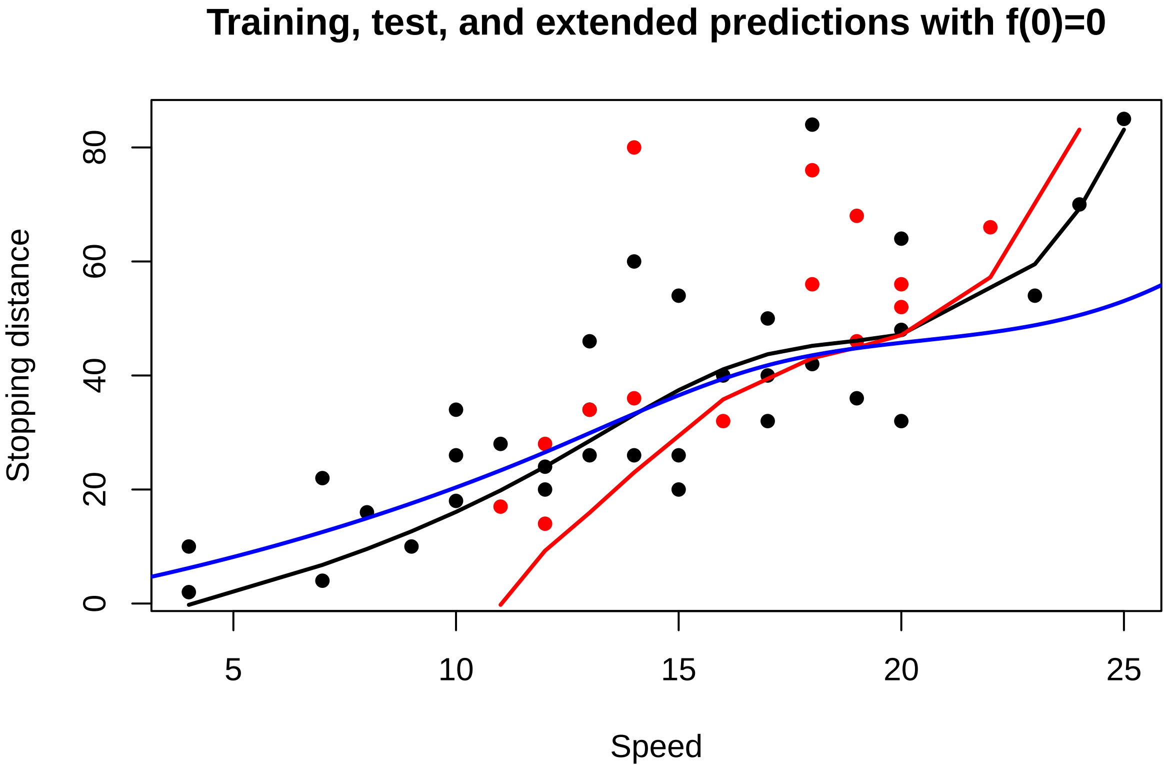
The new plot looks much better.
The train, test, and extended prediction curves now follow the same shape because all of them use the same knots.
The blue line (extended range) no longer bends in a strange way, it's smooth and consistent with the fitted part of the model.
Overall, the predictions make sense now, and the behaviour outside the data range looks more stable.

Ex-6 Another problem

I added a virtual point (0, 0). I inserted one artificial training observation: speed = 0, distance = 0.Then I refitted the model.
This forces the model to understand the basic physical rule: when speed is zero, the stopping distance must also be zero.
Meanwhile I kept the same knots. All spline bases (for training, test, and extended x) still use the same knots taken from the original training data.
The structure of the spline stayed the same, only the training information changed.

```r
# add a virtual observation (0, 0) to the training data
cars_train_aug <- rbind(
  data.frame(speed = 0, dist = 0),
  cars_train
)

# keep the knots from the previous training fit
knots_train <- spl_train$xquantiles

# build spline basis for the augmented training set using the same knots
spl_train_aug <- lecturespl(
  x          = cars_train_aug$speed,
  q          = 2,
  M          = 4,
  xquantiles = knots_train
)

X <- spl_train_aug$X

# refit the spline model on the augmented training data
fit_spl2 <- lm(cars_train_aug$dist ~ X)

# build bases for original train, test, and new x using the same knots
X_train2 <- lecturespl(
  x          = cars_train$speed,
  q          = 2,
  M          = 4,
  xquantiles = knots_train
)$X

X_test2 <- lecturespl(
  x          = cars_test$speed,
  q          = 2,
  M          = 4,
  xquantiles = knots_train
)$X

new_x <- seq(0, 30, length.out = 200)

X_new2 <- lecturespl(
  x          = new_x,
  q          = 2,
  M          = 4,
  xquantiles = knots_train
)$X

# predictions for train, test and extended range
```

```r
yhat_train2 <- predict(fit_spl2, newdata = list(X = X_train2))
yhat_test2  <- predict(fit_spl2, newdata = list(X = X_test2))
yhat_new2   <- predict(fit_spl2, newdata = list(X = X_new2))

# plot all results in one figure
ord_train <- order(cars_train$speed)
ord_test  <- order(cars_test$speed)

plot(cars_train$speed, cars_train$dist,
     col = "black", pch = 16,
     xlab = "Speed",
     ylab = "Stopping distance",
     main = "Training, test, and extended predictions with f(0)=0")

points(cars_test$speed, cars_test$dist,
       col = "red", pch = 16)

# training and test prediction curves
lines(cars_train$speed[ord_train],
      yhat_train2[ord_train], col = "black", lwd = 2)
lines(cars_test$speed[ord_test],
      yhat_test2[ord_test], col = "red", lwd = 2)

# extended-range prediction
lines(new_x, yhat_new2, col = "blue", lwd = 2)
```



Comment

Black points = training data, Red points = test data, Black curve = training predictions, Red curve = test predictions, Blue curve = predictions for the extended range.

Now around speed = 0, the predicted stopping distance is close to 0. This is physically correct and matches expectations.
The curve in the main data range still looks good, the black and blue curves rise smoothly and follow the data well.
Adding one point did not distort the model.
The test curve remains close to the training curve. This shows the model is stable and consistent across both sets.
The extended prediction looks reasonable.
It starts at zero, grows smoothly, and continues the same overall trend without strange jumps.