# ID 5059 Practical 1

180025784 MSc Applied Statistics and Data mining

## 1 Introduction

This report seeks to analyses the pairwise relationship of MPG against four attributes, including displacement, horsepower, weight and acceleration. During the analysis, there are 3 different kinds of model for predicting these four parameters, which are linear models, bin smooth models and b-spline basis. The data set using in this report is called Auto MPG, which is from UCI Machine Learning repository. Data cleaning, model fitting and the calculation of results including AIC, RSS, MSE etc. were conducted in the R programming language using R-Studio.

## 2 Methods

### 2.1 Data Cleaning

By observing the data set, there are unknown values in the observations. The unknown values should be cleaned at the beginning the analysis in case of the inaccurate results. Before the data set being cleaned, there are 398 observations. After doing the data cleaning, 6 unknown values were removed and there are 392 observations.

### 2.2 Model Fitting

The report used the code from P01-code-stats-driver.R to perform linear regression, bin smooths regression and b-spline regression. The assumption that made on all the models is that all four attributes are significant in making predictions of the response variable MPG.

### 2.3 MSE

In this analysis, I use Mean Square Error (MSE) as the measure of generalisation error. In machine learning, generalisation error tests whether the algorithm can accurately predict data that was previously unseen. In this report, due to the fact that the function of three models give the output of residual sum of squares (RSS), I calculate the average of MSE using the equation below:

$$\textbf{MSE} = \textbf{RSS} / \boldsymbol{n}$$

**where *n* refers to the length of data**

The larger the MSE is, the worse the model is.

## 3 Results

### 3.1 RSS of linear models

| Linear Regression | AIC | RSS | MSE |
|---|---|---|---|
| displacement | 2318.83 | 8378.822 | 21.37455 |
| horsepower | 2363.324 | 9385.916 | 23.94366 |
| weight | 2265.939 | 7321.234 | 18.67662 |
| acceleration | 2650.969 | 19550.46 | 49.87362 |

From the results of linear regression model, we can see that the attribute weight has the smallest AIC score and RSS. The predictor of displacement and horsepower is a little bit worse than weight, but much better the acceleration.

**3.2 RSS of bin smooths**

| Bin Smooths | AIC | RSS | MSE |
|---|---|---|---|
| displacement | 2258.165 | 6547.693 | 16.70330 |
| horsepower | 2284.614 | 7004.712 | 17.86916 |
| weight | 2257.155 | 6530.837 | 16.66030 |
| acceleration | 2649.26 | 17757.41 | 45.29952 |

From the results of bin smooths model, it is clear that displacement, horsepower and weight all show comparatively small AIC, RSS and mean MSE. Among these three predictors, weight is still the best one. Acceleration still have large RSS and MSE.

**3.3 RSS of b-spline bases**

| B-spline | AIC | RSS | MSE |
|---|---|---|---|
| displacement | 2243.813 | 6608.903 | 16.85945 |
| horsepower | 2269.845 | 7062.692 | 18.01707 |
| weight | 2247.391 | 6669.5 | 17.01403 |
| acceleration | 2643.998 | 18626.84 | 47.51745 |

The results of b-spline base model illustrate that displacement has the smallest AIC, RSS and MSE, which is different from the above two models. However, acceleration stays at the worst position which is same as before.

## 4 Discussion

**4.1 Which attribute / predictor has the best predictive ability and why?**

RSS and AIC both measure how well the model fits. The smaller RSS and AIC is, the better the model fits. For all four covariates, the AIC and RSS of bin smooths model shows the smallest value, which makes bin smooths model is the best among these three models. For linear regression model and bin smooths model, weight has the smallest RSS and AIC, which makes it the best predictor. However, in b-spline base model, displacement has the smallest RSS and AIC. Thus, we can say displacement shows better predictive ability than others in b-spline model. Because in three models, the difference of RSS and AIC of weight and displacement is very small, I believe these two attributes both can show the best predictive ability.

**4.2 What size of bins did you choose for the bin smoothing and why? What effect would decreasing or increasing the number of bins have on the residual sum of squares?**

The size of bins that I chose is 20 because it gives me the suitable RSS that the model needs. Also, it prevents the overfit of the model and fits quite well. If I decrease the number of bins, the RSS will be decreased; On the contrary, if I increase the number of bins, the RSS will also be increased.

**4.3 What knots and degree of polynomial did you pick for the b-splines and why?**

With regards to the knots, I picked 2 knots for all four covariates. By plotting the scatter diagram between mpg and the four attributes, I can see that there are about three different slopes in the plot. Therefore, the knots should be 2 for all the covariates. When choosing the degree, I considered it should be between 1 to 10, so I tried them one by one. Thus, I realised that when the degree is very small (e.g. 3), the curve is quite smooth, but when the parameters become larger (e.g. 0.8), the model does not fit well. Nevertheless, when the degree is quite high (e.g. 10), the prediction is much better than with smaller degree, but when the parameters are small (e.g. 0.1), the model shows a sharp drop. Thus, with the displacement, horsepower and weight attributes, 8 is a good degree for smooth model and avoid overfitting. And with the acceleration attribute, 5 is a better degree.

# 5 References


Raudys, R. (2001) *Performance and the Generalisation Error*. London: Springer.

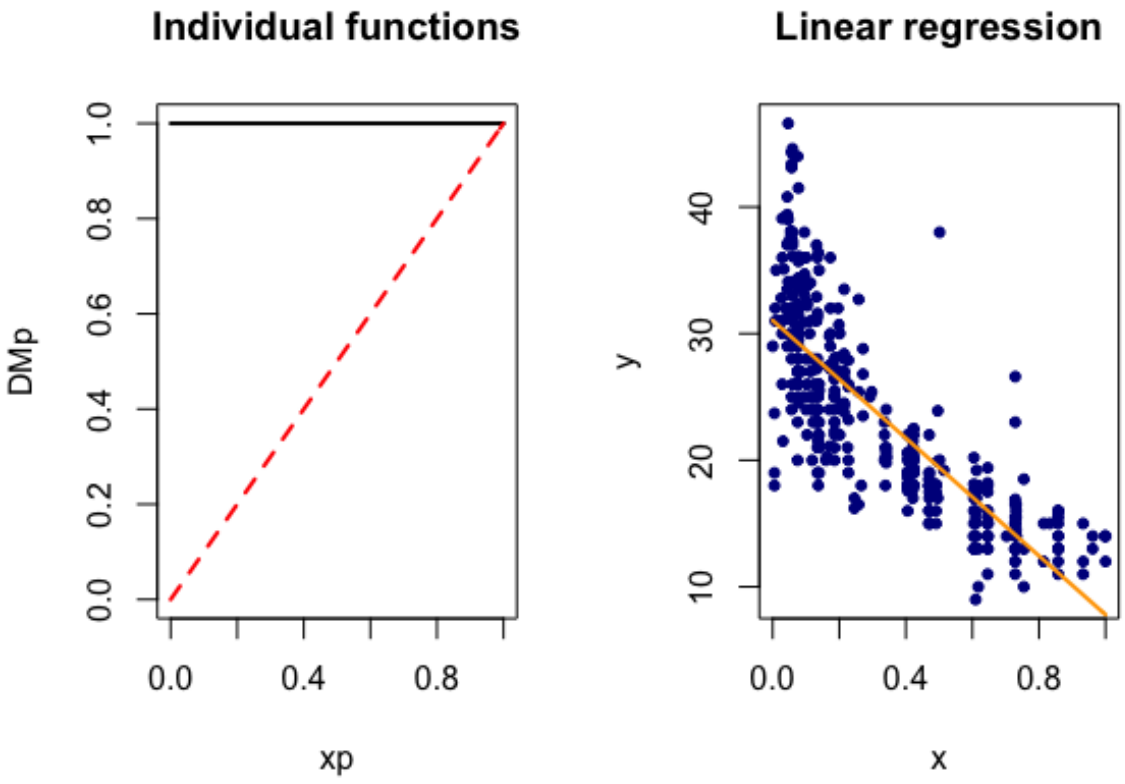
R Core Team (2018) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. Available at: https://www.R-project.org/ (Accessed: 2 Mar 2019)

UCI Machine Learning repository (1993) *Auto MPG Data Set*. Available at: https://archive.ics.uci.edu/ml/datasets/auto+mpg/ (Accessed: 2 Mar 2019)

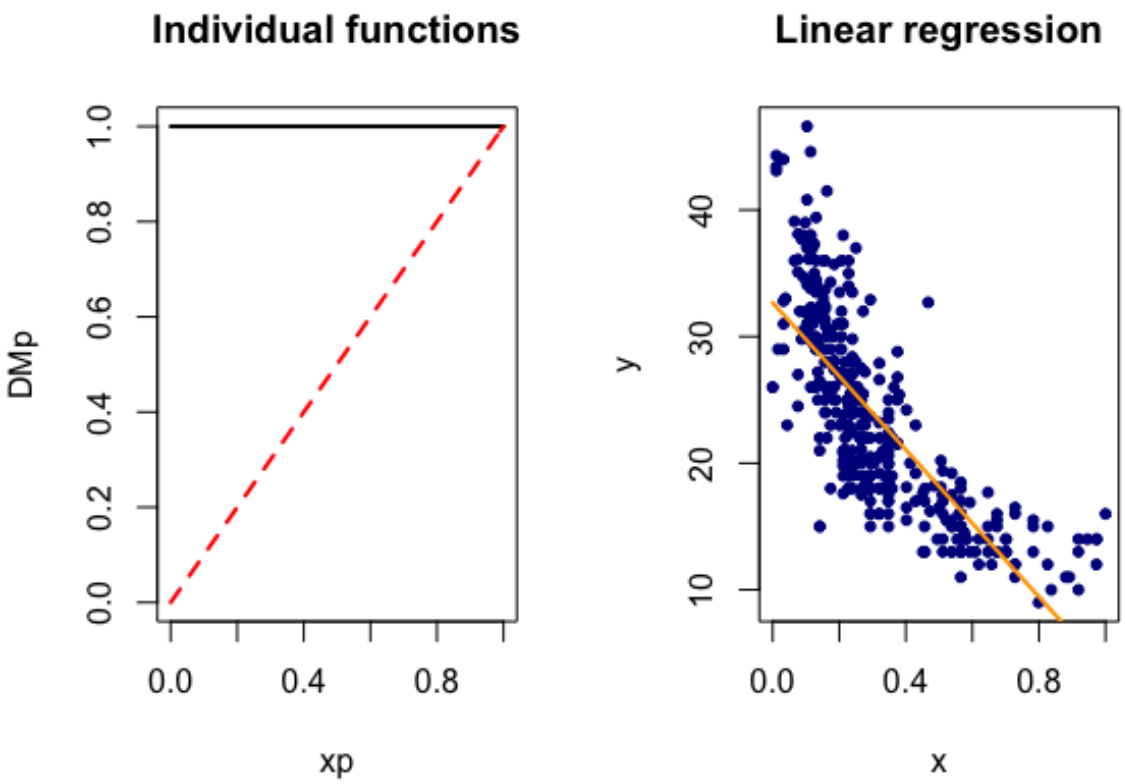

# 6 Appendix

## 6.1 Linear Models

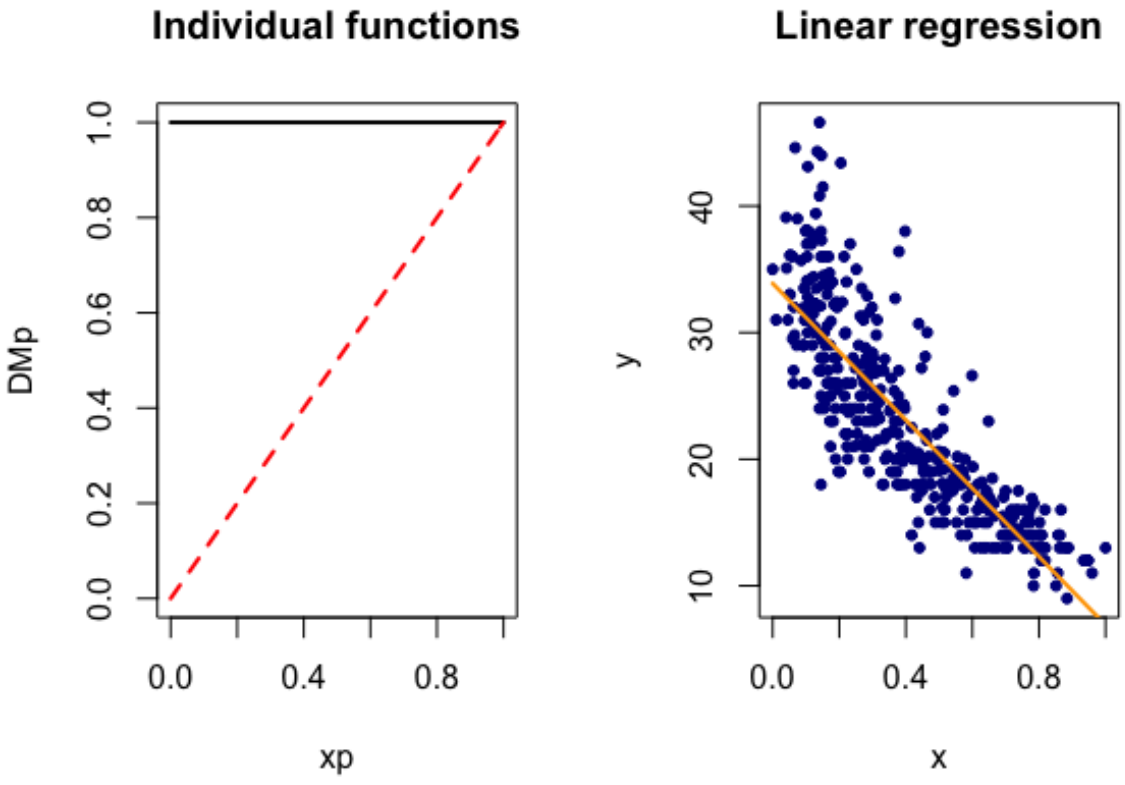
### 6.1.1 mpg vs displacement



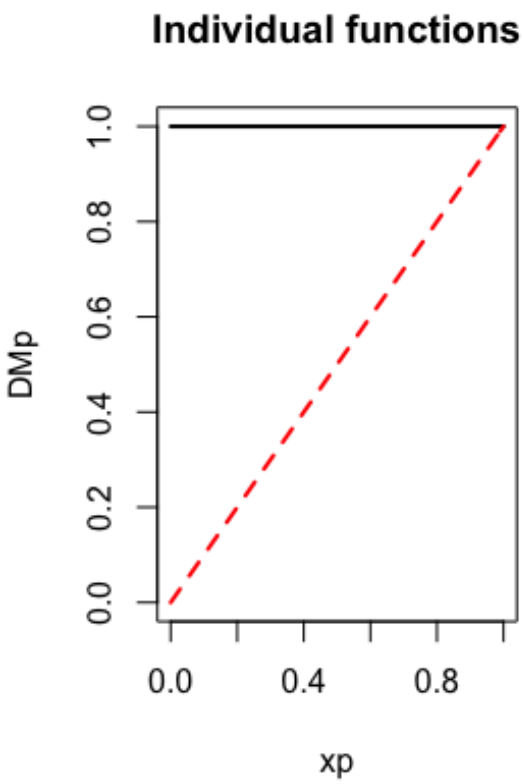
### 6.1.2 mpg vs horsepower

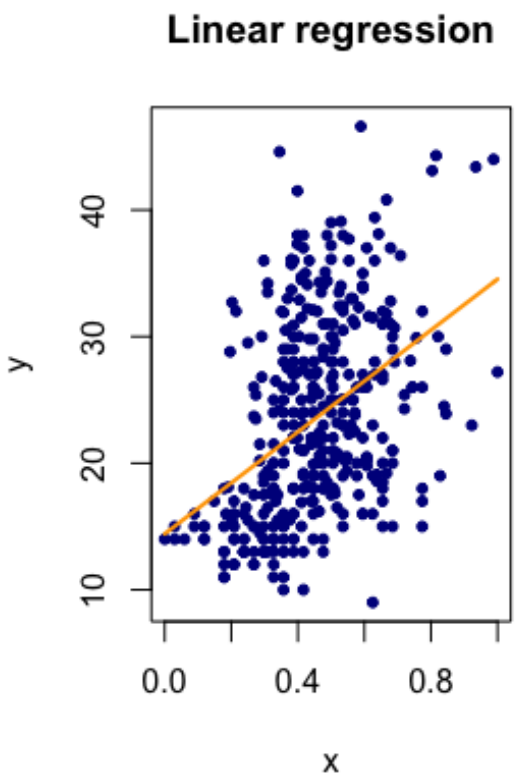



### 6.1.3 mpg vs weight



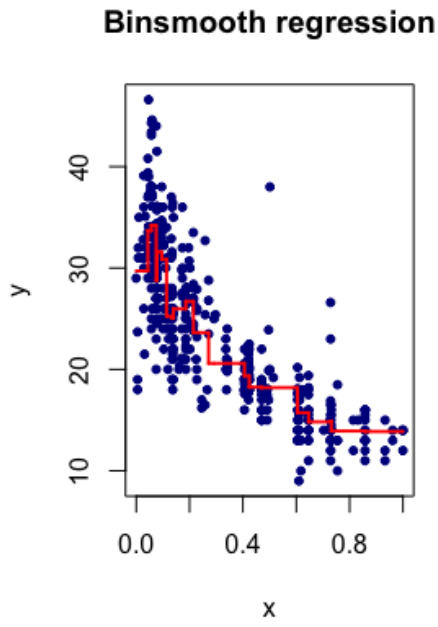
### 6.1.4 mpg vs acceleration



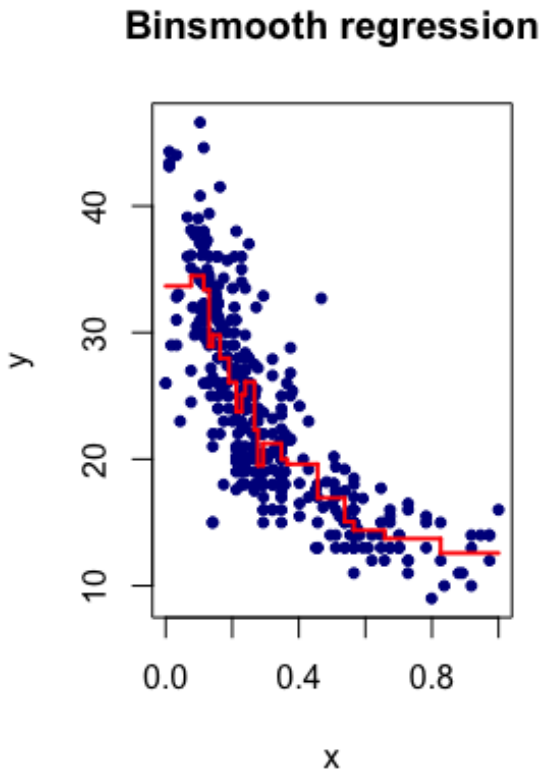




## 6.2 Bin smooths

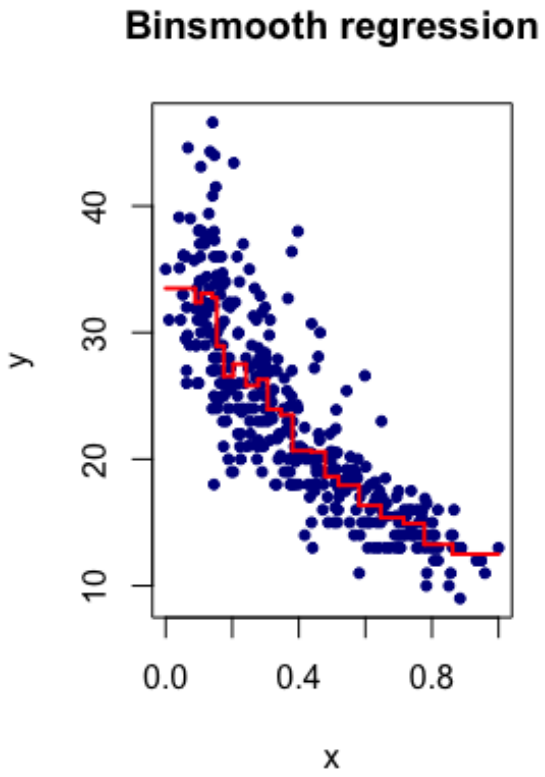
### 6.2.1 mpg vs displacement



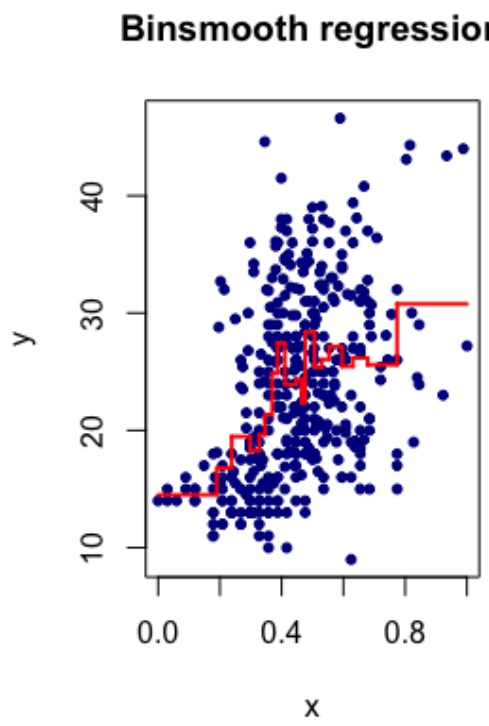

### 6.2.2 mpg vs horsepower




**6.2.3 mpg vs weight**

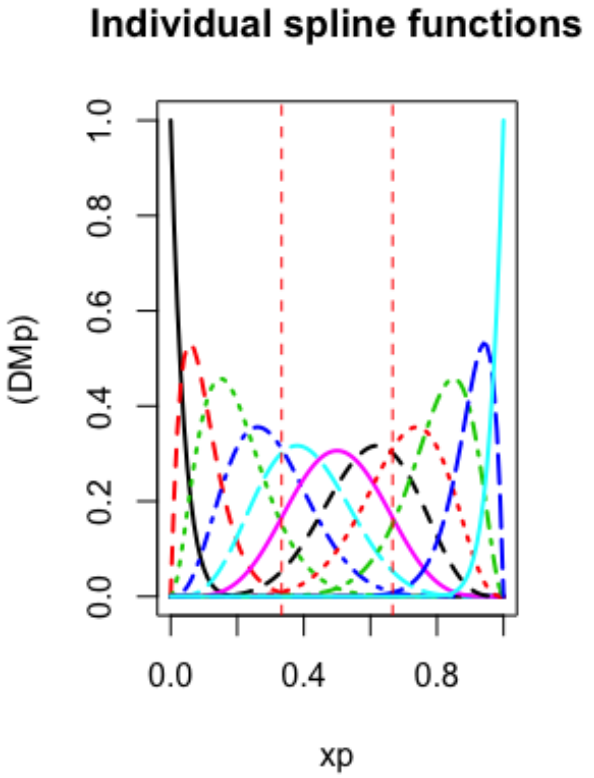


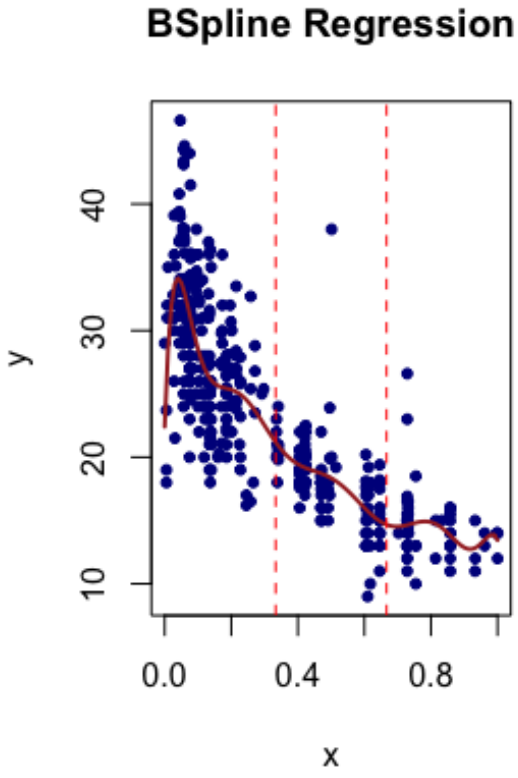

**6.2.4 mpg vs acceleration**



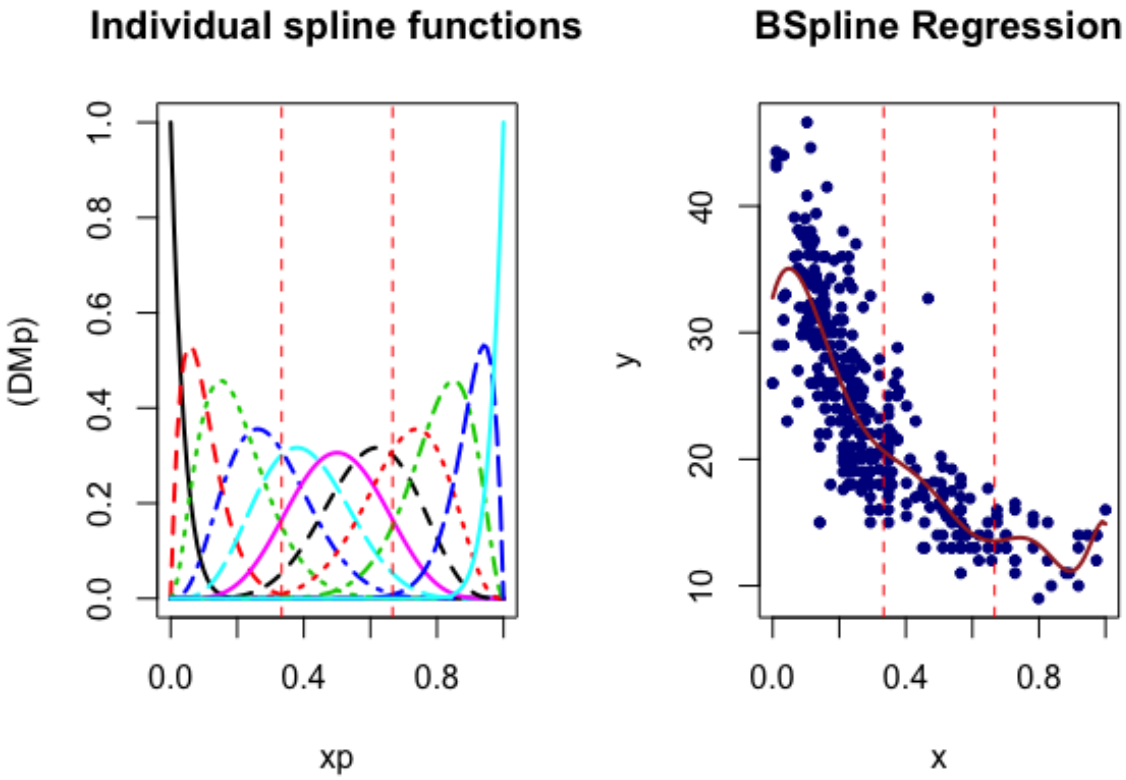

**6.3 b-spline**

**6.3.1 mpg vs displacement**

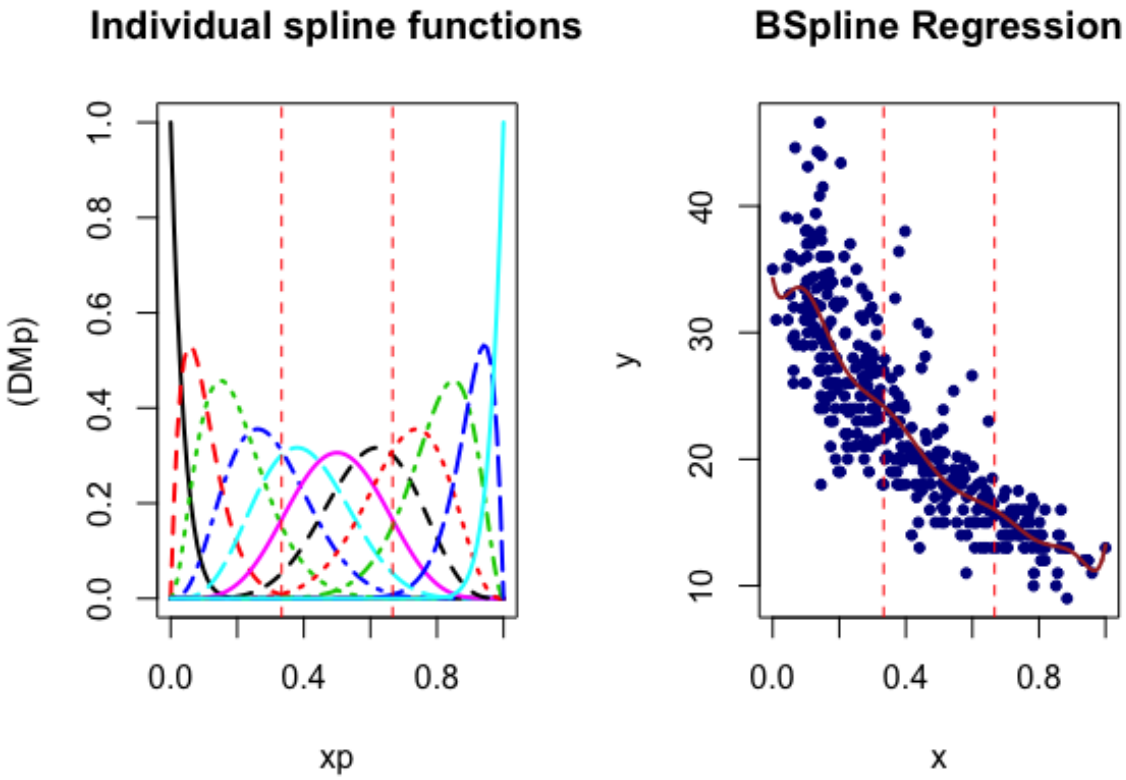


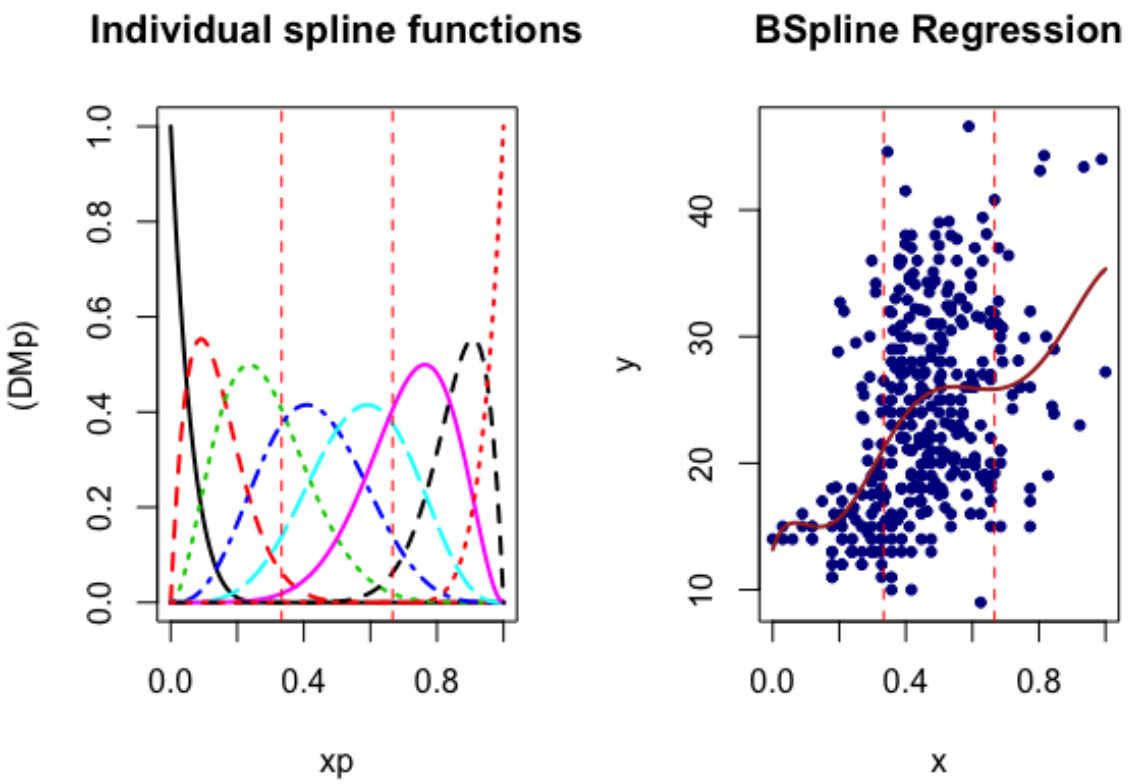

**6.3.2 mpg vs horsepower**




### 6.3.3 mpg vs weight

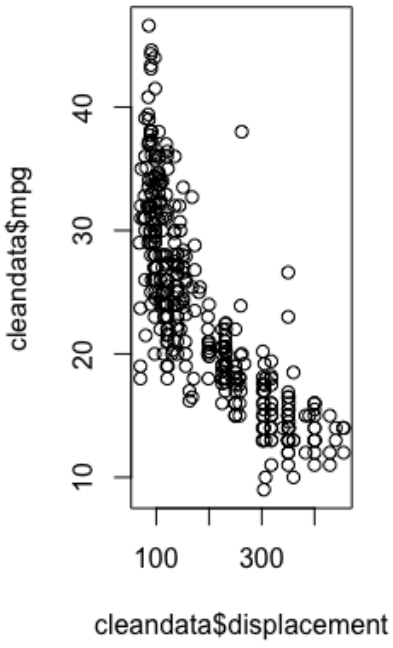


### 6.3.4 mpg vs acceleration

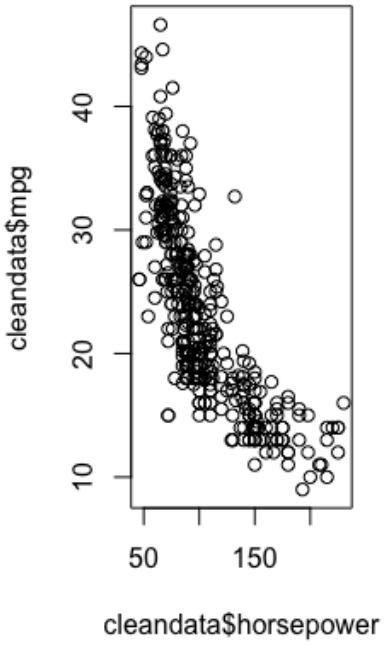



## 6.4 Scatter Plots

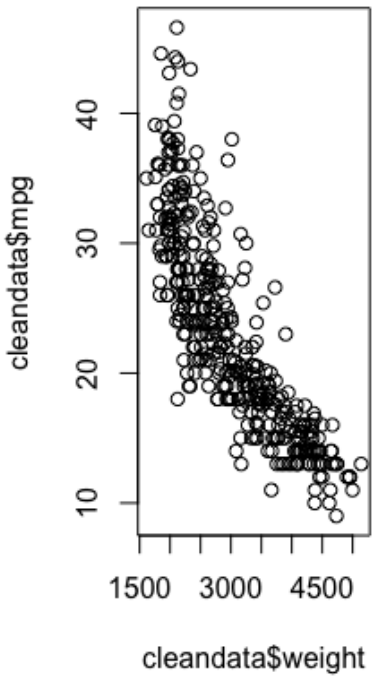
### 6.4.1 Scatter plot between mpg and displacement



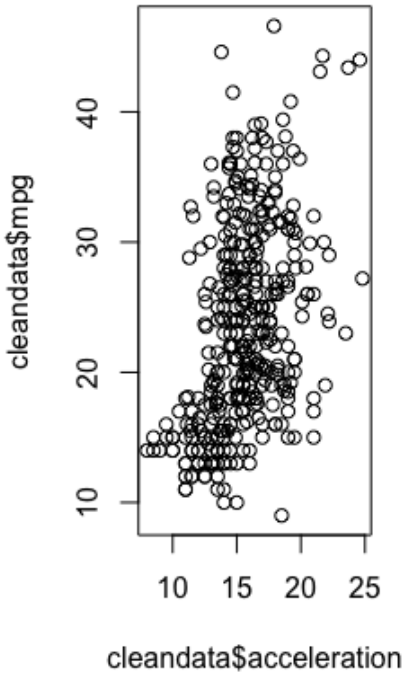
### 6.4.2 Scatter plot between mpg and horsepower




### 6.4.3 Scatter plot between mpg and weight



### 6.4.4 Scatter plot between mpg and acceleration




### 6.5 R Code

```
#read data
data <- read.table("/Users/apple/Desktop/ID5059/Practicals/P01/auto-mpg.data", sep=" ",
header=T)

attach(data)

#remove rows with NAs
cleandata <- na.omit(data)

# convert to comma separated values for use in Excel
write.csv(cleandata,"auto-data.csv")

#####1 Linear Models
#The function linreg performs a linear regression
#Input Arguments:
#       x - vector containing the explonatory variable
#       y - vector containing the dependent variable
#       ouptut - 1: delivers some output, 0: no output
#       opt   - 1: returns adj R-squared, 0: returns nothing
#       ploto - 1: Create new plot, 0: no new plot
```

```
linreg <- function(x, y, output=1, ploto=1, opt=0)
{
 #Scale data to [0,1]
 x<-x-min(x)
 x<-x/max(x)
 #Create a Design Matrix DM
 n <- length(x)
 q <- 2
 DM = matrix(1,n,q)
 DM[,2] <- x

 #Perform regression
 # beta <- solve(t(DM)%*%DM)%*%t(DM)%*%y
 reg <- lm(y~0+DM)

 #Calculate goodness of fit measures
 #Residual sum of squares
 rss <-  sum(sapply(residuals(reg), function(x) { x^2 }))
 #Coefficient of determination: R^2
 R2 <- 1 - (rss/ (t(y)%*%y-(mean(y)**2*n)))
 #Adjusted Coefficient of determination: R^2
 R2adj <- 1 - ( (n-1)/(n-q) ) * (1-R2)
 #AIC
 aic <- AIC(reg)

 if(output==1)
 {
  #Summary output
  cat("RSS: ", rss, "\n")
  cat("TSS: ", t(y)%*%y-(mean(y)**2/n), "\n")
  cat("R-squared: ", R2, "\n")
  cat("Adjusted R-squared: ", R2adj, "\n")
  cat("AIC: ", aic, "\n")
  #cat("Coefficents: \n")
  #print(coef(reg))
  #print(summary(reg))
  #print(anova(reg))

  #Graphic
  xp <- 0:100/100
  n <- length(xp)
  DMp = matrix(1,n,q)
  DMp[,2] <- xp
```

```
   if(ploto==1) par(mfrow=c(1,2))
   if(ploto==1) matplot(xp, DMp, type="l", lwd=2, main="Individual functions")
   if(ploto==1) plot(x,y, main="Linear regression", pch=20, col="darkblue")
   lines(xp, DMp%*%coef(reg), col="orange", type="l", lwd=2)
  }

  if(opt==1) {return(c(R2adj,aic))}
}

#show the graphs and RSS of linear regression and calculate the MSE
##1 displacement
linreg(cleandata$displacement, cleandata$mpg)
mse_displacement1 <- 8378.822/length(cleandata$mpg)

##2 horsepower
linreg(cleandata$horsepower, cleandata$mpg)
mse_horsepower1 <- 9385.916/length(cleandata$mpg)

##3 weight
linreg(cleandata$weight, cleandata$mpg)
mse_weight1 <- 7321.234/length(cleandata$mpg)

##4 acceleration
linreg(cleandata$acceleration, cleandata$mpg)
mse_acceleration1 <- 19550.46/length(cleandata$mpg)



#mse of all four attributes
mse_linear <- c(mse_displacement1, mse_horsepower1, mse_weight1, mse_acceleration1)
mse_linear



#####2 Bin Smooths
#The function binsmoothREG performs a binsmooth regression with a user defined binlength
#Input Arguments:
#       x - vector containing the explonatory variable
#       y - vector containing the dependent variable
#       binlength - amount of x values per bin
#       ouptut - 1: delivers some output, 0: no output
#       opt   - 1: returns adj R-squared, 0: returns nothing
#       ploto - 1: Create new plot, 0: no new plot
```

```
binsmoothREG <- function(x, y, binlength=20, knotsdef=NULL, output=1, ploto=1, opt=0)
{
 #Scale data to [0,1]
 x<-x-min(x)
 x<-x/max(x)
 #Sort x values in ascending order
 y <- y[order(x)]
 x <- sort(x)
 n <- length(x)
 #Devide data into bins
 if(is.vector(knotsdef)) bins = knotsdef
 else bins = ceiling(length(x) / binlength)
 #Create Design Matrix without intercept
 DM <- matrix(1, length(x), bins)
 #Set all elements not corresponding to region j equal 0
 for(i in 1:bins)
 {
  if(i==1) { xstart = 1 }
  if(i>1) { xstart = (i-1)*binlength+1 }
  xend = min(xstart + binlength-1, length(x))
  binelements <- xstart:xend
  elements <- 1:length(x)
  elements[binelements] <- 0
  DM[elements,i] <- 0
 }

 #Perform Linear Regreesion
 reg <- lm(y~0+DM)

 #Calculate goodness of fit measures
 q <- bins
 #Residual sum of squares
 rss <-  sum(sapply(residuals(reg), function(x) { x^2 }))
 #Coefficient of determination: R^2
 R2 <- 1 - (rss/ (t(y)%*%y-(mean(y)**2*n)))
 #Adjusted Coefficient of determination: R^2
 R2adj <- 1 - ( (n-1)/(n-q) ) * (1-R2)
 #AIC
 aic <- AIC(reg)

 if(output==1)
 {
```

```
   #Summary output
   cat("Elements per bin: ", binlength, "\n")
   cat("Number of bins: ", bins, "\n")
   cat("RSS: ", rss, "\n")
   cat("TSS: ", t(y)%*%y-(mean(y)**2/n), "\n")
   cat("R-squared: ", R2, "\n")
   cat("Adjusted R-squared: ", R2adj, "\n")
   cat("AIC: ", aic, "\n")
   #cat("Coefficents: \n")
   #print(coef(reg))
   #print(summary(reg))
   #print(anova(reg))

   #Graphic
   if(ploto==1) plot(x,y, main="Binsmooth regression", pch=20, col="darkblue")
   j<-1
   for(i in 1:length(coef(reg)))
   {
    if(i>1) lines(c(x[xend],x[xend]), c(as.numeric(coef(reg)[i-1]), as.numeric(coef(reg)[i])),
col="red", lwd=2)
    xstart = j
    if(i>1) lines(c(x[xend],x[xstart]), c(as.numeric(coef(reg)[i]), as.numeric(coef(reg)[i])),
col="red", lwd=2)
    xend = min(j+binlength-1, length(x))
    lines(c(x[xstart],x[xend]), rep(as.numeric(coef(reg)[i]), 2), col="red", lwd=2)
    j<-j+binlength

   }
  }

  if(opt==1) return(c(R2adj,aic))
}


#show the graphs and RSS of bin smooths model and calculate the MSE
##1 displacement
binsmoothREG(cleandata$displacement, cleandata$mpg)
mse_displacement2 <- 6547.693/length(cleandata$mpg)

##2 horsepower
binsmoothREG(cleandata$horsepower, cleandata$mpg)
mse_horsepower2 <- 7004.712/length(cleandata$mpg)

##3 weight
```

```
binsmoothREG(cleandata$weight, cleandata$mpg)
mse_weight2 <- 6530.837/length(cleandata$mpg)

##4 acceleration
binsmoothREG(cleandata$acceleration, cleandata$mpg)
mse_acceleration2 <- 17757.41/length(cleandata$mpg)


#mse of all four attributes
mse_binsmooths <- c(mse_displacement2, mse_horsepower2, mse_weight2,
mse_acceleration2)
mse_binsmooths



#####3 b-spline bases
#The function bsplinereg performs a bspline regression with user defined knots
#Input Arguments:
#Input Arguments bsplinereg(...):
#       x - vector containing the explonatory variable
#       y - vector containing the dependent variable
#       knots - number of knots in [0,1]
#       ouptut - 1: delivers some output, 0: no output
#       opt   - 1: returns adj R-squared, 0: returns nothing
#       ploto - 1: Create new plot, 0: no new plot

#Calculate basis (rekursiv)
basis <- function(x, degree, i, knots)
{

 if(degree == 0)
 { B <- ifelse((x >= knots[i]) & (x < knots[i+1]), 1, 0)
 } else {
  if((knots[degree+i] - knots[i]) == 0)
  { alpha1 <- 0
  } else {
    alpha1 <- (x - knots[i])/(knots[degree+i] - knots[i]) }

  if((knots[i+degree+1] - knots[i+1]) == 0)
  { alpha2 <- 0
  } else { alpha2 <- (knots[i+degree+1] - x)/(knots[i+degree+1] - knots[i+1]) }
  B <- alpha1*basis(x, (degree-1), i, knots) + alpha2*basis(x, (degree-1), (i+1), knots)
 }
```

```
  return(B)
}

#Create bspline Desin Matrix
bspline <- function(x, degree, knotpos)
{
  #Number of basis
  K <- length(knotpos) + degree + 1
  #Number of observations
  n <- length(x)
  #Set Boundary knots
  Boundary.knots = c(0,1)
  #create new vector with knot positons
  knotpos <- c(rep(Boundary.knots[1], (degree+1)), knotpos, rep(Boundary.knots[2],
(degree+1)))


  #Create design matrix
  DM <- matrix(0,n,K)
  for(j in 1:K) DM[,j] <- basis(x, degree, j, knotpos)
  if(any(x == Boundary.knots[2])) DM[x == Boundary.knots[2], K] <- 1
  #Return DM
  return(DM)
}


bsplinereg <- function(x, y, knots=0, knotsdef=NULL,  degree, output=1, ploto=1, opt=0)
{
  #Scale data to [0,1]
  x<-x-min(x)
  x<-x/max(x)
  #Sort x values in ascending order
  y <- y[order(x)]
  x <- sort(x)
  n <- length(x)

  #Calculate knot postions
  if(knots == 0) knotpos <- NULL
  if(knots != 0) knotpos <- 1:knots / (knots+1)
  if(length(knotsdef)>0) knotpos <- knotsdef
  #Create Design Matrix
  DM <- bspline(x, degree, knotpos)

  #Perform penalized regression
```

```
reg <- lm(y ~ 0 + DM)
print(summary(reg))

#Calculate goodness of fit measures
q <- length(knotpos) + degree + 1
#Residual sum of squares
rss <-  sum(sapply(residuals(reg), function(x) { x^2 }))
#Coefficient of determination: R^2
R2 <- 1 - (rss/ (t(y)%*%y-(mean(y)**2*n)))
#Adjusted Coefficient of determination: R^2
R2adj <- 1 - ( (n-1)/(n-q) ) * (1-R2)
#AIC
aic <- AIC(reg)

if(output==1)
{
 #Summary output
 cat("Number of knots = ", knots, "\n")
 cat("Knot positions = ", knotpos, "\n")
 cat("RSS: ", rss, "\n")
 cat("TSS: ", t(y)%*%y-(mean(y)**2/n), "\n")
 cat("R-squared: ", R2, "\n")
 cat("Adjusted R-squared: ", R2adj, "\n")
 cat("AIC: ", aic , "\n")
 #cat("Coefficents: \n")
 #print(coef(reg))
 #print(summary(reg))
 #print(anova(reg))

 #Graphics

 #Values for prediction
 xp <- 0:100/100
 DMp <- bspline(xp, degree, knotpos)

 if(ploto==1)par(mfrow=c(1,2))
 if(ploto==1) matplot(xp, (DMp), type="l", lwd=2, main="Individual spline functions")
 if(ploto==1) for(i in 1:length(knotpos)) abline(v=knotpos[i], col="red", lty=2)
 if(ploto==1) plot(x,y, main="BSpline Regression", pch=20, col="darkblue")

 points(xp,DMp%*%coef(reg), type="l", lwd=2, col="brown")
 if(ploto==1) for(i in 1:length(knotpos)) abline(v=knotpos[i], col="red", lty=2)
}
if(opt==1) return(c(R2adj, aic))
```

```
}


#plot the relationship of mpg and x variables for suitable knots and degree
plot(cleandata$mpg~cleandata$displacement)
plot(cleandata$mpg~cleandata$horsepower)
plot(cleandata$mpg~cleandata$weight)
plot(cleandata$mpg~cleandata$acceleration)
#knots should be 2

#show the graphs and RSS of b-spline and calculate the MSE
##1 displacement
bsplinereg(cleandata$displacement, cleandata$mpg, degree = 8, knots = 2)
mse_displacement3 <- 6608.903/length(cleandata$mpg)

##2 horsepower
bsplinereg(cleandata$horsepower, cleandata$mpg, degree = 8, knots = 2)
mse_horsepower3 <- 7062.692/length(cleandata$mpg)

##3 weight
bsplinereg(cleandata$weight, cleandata$mpg, degree = 8, knots = 2)
mse_weight3 <- 6669.5/length(cleandata$mpg)

##4 acceleration
bsplinereg(cleandata$acceleration, cleandata$mpg, degree = 5, knots = 2)
mse_acceleration3 <- 18626.84/length(cleandata$mpg)


#mse of all four attributes
mse_bspline <- c(mse_displacement3, mse_horsepower3, mse_weight3, mse_acceleration3)
mse_bspline
```