Ad Analysis

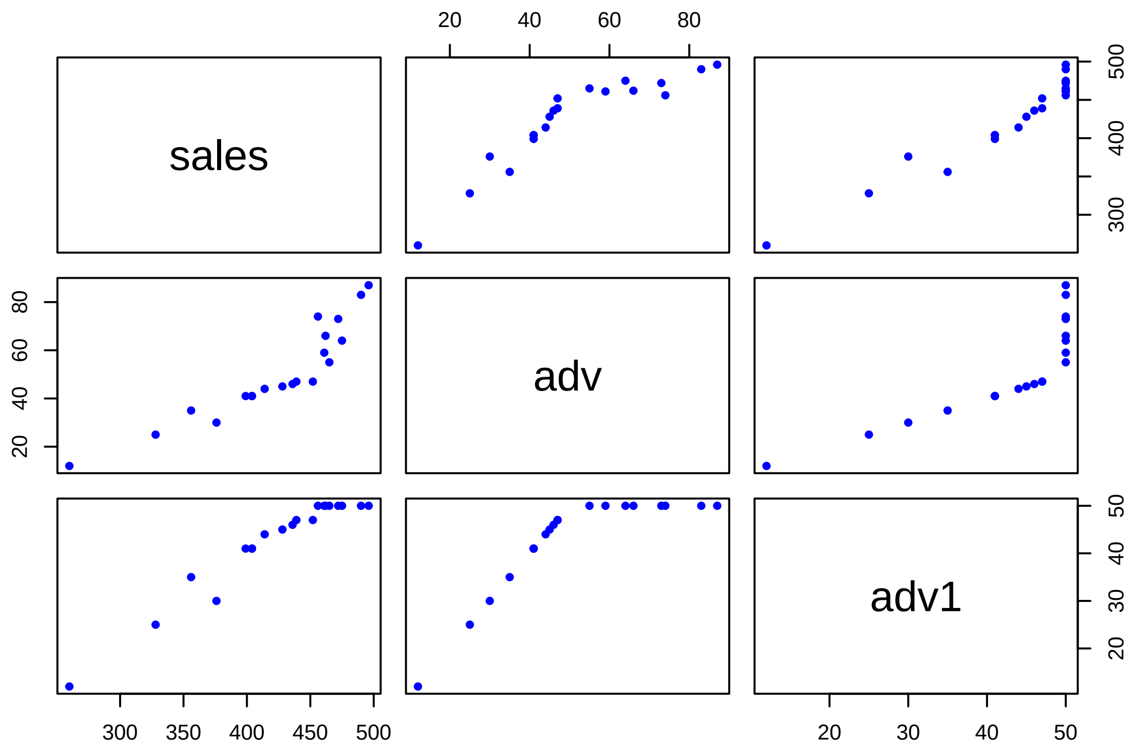
Gustavo Cepparo and Milica Cudina

First, we download the data set and take a look at it.

```
data<-read.csv("salesadv.csv",header=TRUE)
attach(data)
data[,-4]
```

```
##    sales adv adv1
## 1    260  12   12
## 2    328  25   25
## 3    376  30   30
## 4    356  35   35
## 5    404  41   41
## 6    399  41   41
## 7    404  41   41
## 8    414  44   44
## 9    428  45   45
## 10   436  46   46
## 11   439  47   47
## 12   452  47   47
## 13   465  55   50
## 14   461  59   50
## 15   475  64   50
## 16   462  66   50
## 17   472  73   50
## 18   456  74   50
## 19   490  83   50
## 20   496  87   50
```

```
plot(data[,-4],
     col="blue", pch=20)
```



Just for laughs, let's fit a multiple linear regression.

```
lm.fit.m=lm(sales ~ adv + adv1, data=data)
summary(lm.fit.m)
```

```
##
## Call:
## lm(formula = sales ~ adv + adv1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.2078  -4.0006   0.1739   6.1054  23.9011
##
## Coefficients:
##             Estimate Std. Error t value        Pr(>|t|)
## (Intercept) 201.4454    11.6992  17.219 0.0000000000034 ***
## adv           0.9658     0.2398   4.027        0.000875 ***
## adv1          4.0560     0.4571   8.874 0.0000000865618 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.05 on 17 degrees of freedom
## Multiple R-squared:  0.9684, Adjusted R-squared:  0.9647
## F-statistic: 260.6 on 2 and 17 DF,  p-value: 0.0000000000001763
```

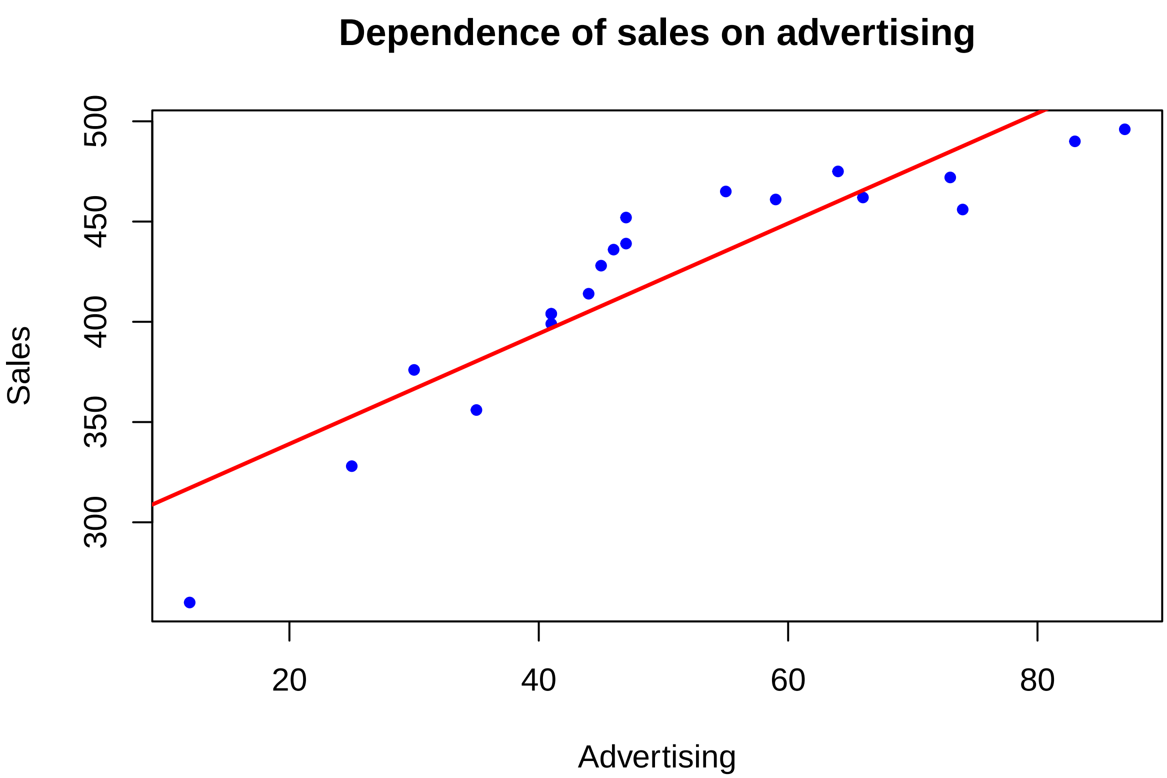
```
lm.fit.mi=lm(sales ~ adv*adv1, data=data)
summary(lm.fit.mi)
```

```
##
## Call:
## lm(formula = sales ~ adv * adv1, data = data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.9117  -4.5606  -0.2768   6.1266  25.2126
##
## Coefficients:
##               Estimate Std. Error t value    Pr(>|t|)
## (Intercept) 209.115091  25.982946   8.048 0.000000513 ***
## adv           0.481233   1.476049   0.326       0.749
## adv1          3.981928   0.519587   7.664 0.000000964 ***
## adv:adv1      0.008642   0.025956   0.333       0.744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.35 on 16 degrees of freedom
## Multiple R-squared:  0.9686, Adjusted R-squared:  0.9627
## F-statistic: 164.7 on 3 and 16 DF,  p-value: 0.000000000003089
```

We are going to focus on just the fit with `adv` as a single predictor. So, let's look at the simple linear regression.

```r
plot(adv, sales,
     pch=20, col="blue",
     main="Dependence of sales on advertising",
     xlab="Advertising",
     ylab="Sales")
lm.fit<-lm(sales ~ adv)
abline(lm.fit, col="red", lwd=2)
```




```r
summary(lm.fit)
```

```
##
```

```
## Call:
## lm(formula = sales ~ adv)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.091 -22.836   7.162  16.226  38.662
##
## Coefficients:
##             Estimate Std. Error t value         Pr(>|t|)
## (Intercept) 284.0915    16.3292   17.40 0.00000000000105 ***
## adv           2.7499     0.3015    9.12 0.00000003615574 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.48 on 18 degrees of freedom
## Multiple R-squared:  0.8221, Adjusted R-squared:  0.8122
## F-statistic: 83.17 on 1 and 18 DF,  p-value: 0.00000003616
```

Since there appears to be a difference in the "signal" in the vicinity of 50 in `adv`, it's time to consider linear splines. First, we import the library `splines`.

```r
library(splines)
```

Then, we do create a fit with `degree=1` - meaning that it's a linear spline. Note, that we specify a single knot at 50.

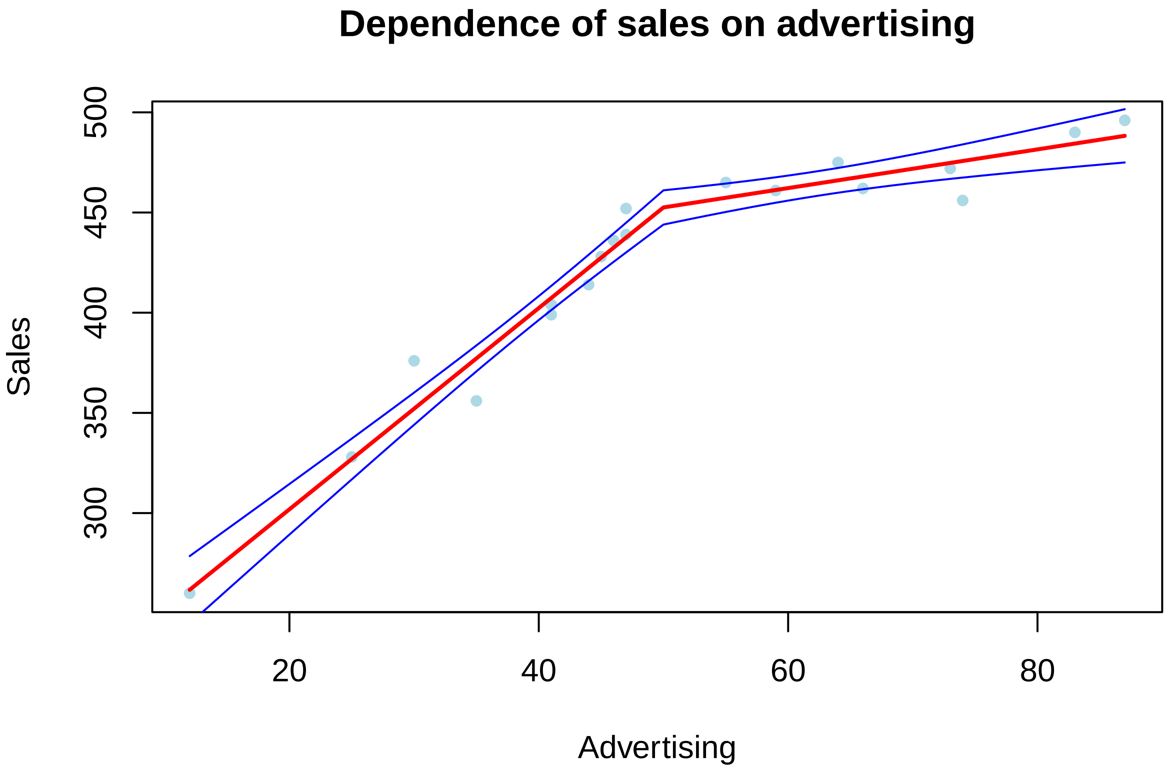
```r
lm.fit.ls<-lm(sales~bs(adv,knots=c(50),degree=1),data=data)
summary(lm.fit.ls)
```

```
##
## Call:
## lm(formula = sales ~ bs(adv, knots = c(50), degree = 1), data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.2078  -4.0006   0.1739   6.1054  23.9011
##
## Coefficients:
##                                     Estimate Std. Error t value
## (Intercept)                           261.71       8.41   31.12
## bs(adv, knots = c(50), degree = 1)1   190.83      10.92   17.47
## bs(adv, knots = c(50), degree = 1)2   226.56      10.07   22.50
##                                               Pr(>|t|)
## (Intercept)                                    < 2e-16 ***
## bs(adv, knots = c(50), degree = 1)1 0.0000000000026980 ***
## bs(adv, knots = c(50), degree = 1)2 0.0000000000000433 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.05 on 17 degrees of freedom
## Multiple R-squared:  0.9684, Adjusted R-squared:  0.9647
## F-statistic: 260.6 on 2 and 17 DF,  p-value: 0.0000000000001763
```

The R^2 is better. What about the visuals?

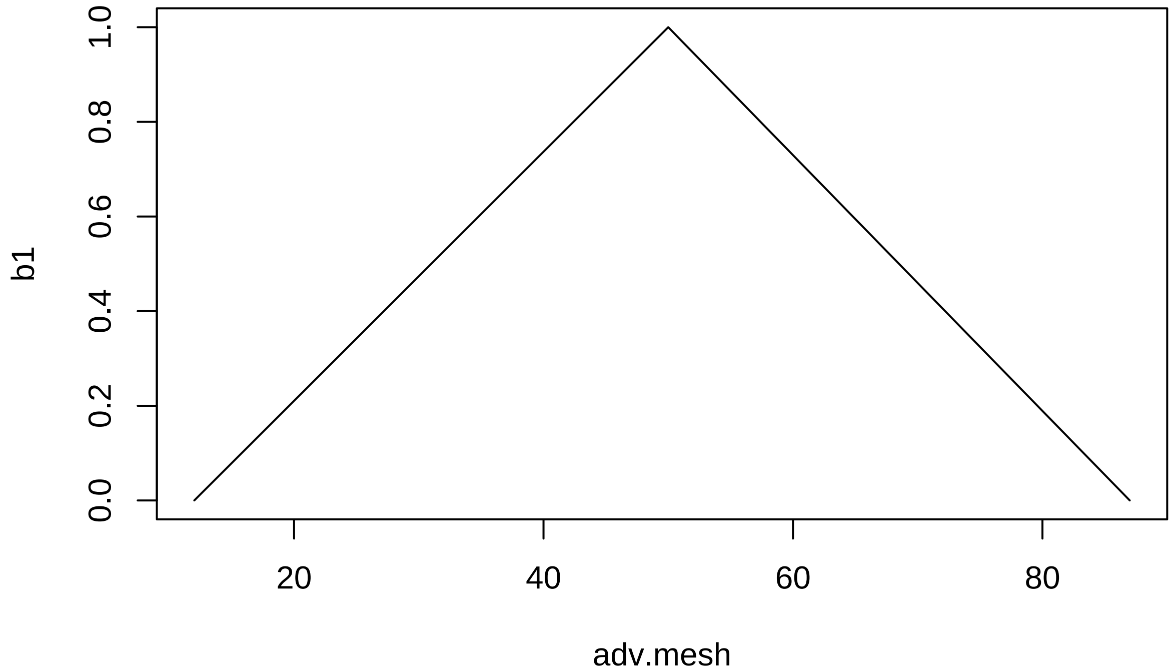
```r
adv.mesh=seq(from=min(adv),to=max(adv), by=0.5)
predictions=predict(lm.fit.ls,newdata=list(adv=adv.mesh),se=T)
```

```r
plot(adv, sales,
     pch=20, col="lightblue",
     main="Dependence of sales on advertising",
     xlab="Advertising",
     ylab="Sales")

lines(adv.mesh,predictions$fit,col="red", lwd=2)
lines(adv.mesh,predictions$fit+2*predictions$se, col="blue")
lines(adv.mesh,predictions$fit-2*predictions$se, col="blue")
```

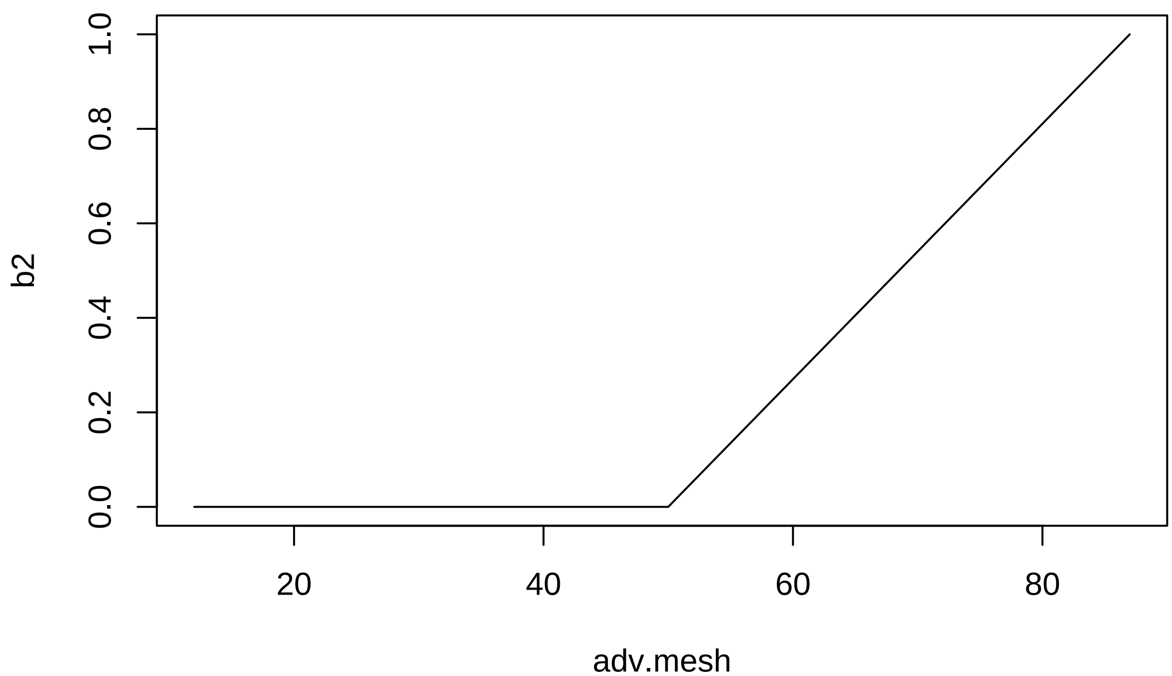


Of course, we should be interested in the basis functions used in this implementation.

```r
## returns a basis matrix
b <- bs(adv.mesh, degree = 1, knots = c(50))

#the functions in the basis
b1 <- b[, 1]
b2 <- b[, 2]
plot(adv.mesh, b1, type = "l")
```




```r
plot(adv.mesh, b2, type = "l")
```




```r
#the coefficients from the summary of `lm`
betas=summary(lm.fit.ls)$coeff[,1]

plot(adv, sales,
     pch=20, col="lightblue",
     main="Dependence of sales on advertising",
     xlab="Advertising",
     ylab="Sales")
```

```
lines(adv.mesh,betas[1]+betas[2]*b1+betas[3]*b2,col="red", lwd=2)
```