# 46-926 Homework 5, Part 1

*Jingyi Guo, Pittsburgh Campus*

*2/18/2017*

## Set Up

```
#set up
library(mgcv)
bonddata = read.table("http://www.stat.cmu.edu/~cschafer/MSCF/bonddata.txt",sep=",",header=T)
bonddatasub = bonddata[,-c(1,2,17:61)]
#convert the factors
bonddatasub$is_callable = factor(bonddatasub$is_callable)
bonddatasub$trade_type = factor(bonddatasub$trade_type)
bonddatasub$trade_type_last1 = factor(bonddatasub$trade_type_last1)
```

## 1 Transformation

```
bonddatasub$weight=log(bonddatasub$weight)
bonddatasub$time_to_maturity=log(bonddatasub$time_to_maturity)
bonddatasub$trade_size=log(bonddatasub$trade_size)
bonddatasub$trade_size_last1=log(bonddatasub$trade_size_last1)
#transform to categorical variable
bonddatasub$reporting_delay=cut(bonddatasub$reporting_delay,c(-Inf,2,10,100,Inf))
bonddatasub$received_time_diff_last1=cut(bonddatasub$received_time_diff_last1,c(-Inf,500,
                                                                                75000,4000000,Inf))
```

## 2 GAM Model

```
holdgam=gam(trade_price ~ s(weight)+s(current_coupon)+s(time_to_maturity)+s(trade_size)
            +s(curve_based_price)+s(trade_price_last1)+s(trade_size_last1)+s(curve_based_price_last1)
+is_callable+reporting_delay+trade_type+received_time_diff_last1+trade_type_last1,
data=bonddatasub)
summary(holdgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## trade_price ~ s(weight) + s(current_coupon) + s(time_to_maturity) +
##     s(trade_size) + s(curve_based_price) + s(trade_price_last1) +
##     s(trade_size_last1) + s(curve_based_price_last1) + is_callable +
##     reporting_delay + trade_type + received_time_diff_last1 +
##     trade_type_last1
##
## Parametric coefficients:
##                                         Estimate Std. Error t value
```

```
## (Intercept)                                105.6099     0.2591 407.584
## is_callable1                                -0.2242     0.1475  -1.520
## reporting_delay(2,10]                       -0.2480     0.1102  -2.250
## reporting_delay(10,100]                     -0.3111     0.1097  -2.836
## reporting_delay(100, Inf]                   -0.6049     0.1491  -4.058
## trade_type3                                  1.4828     0.1119  13.249
## trade_type4                                  0.7326     0.1046   7.005
## received_time_diff_last1(500,7.5e+04]       -0.3660     0.2435  -1.503
## received_time_diff_last1(7.5e+04,4e+06]     -0.4637     0.3139  -1.477
## received_time_diff_last1(4e+06, Inf]        -1.3156     0.5263  -2.500
## trade_type_last13                           -0.9486     0.1122  -8.456
## trade_type_last14                           -0.5029     0.1080  -4.656
##                                            Pr(>|t|)
## (Intercept)                                 < 2e-16 ***
## is_callable1                                0.12865
## reporting_delay(2,10]                       0.02460 *
## reporting_delay(10,100]                     0.00462 **
## reporting_delay(100, Inf]                  5.19e-05 ***
## trade_type3                                 < 2e-16 ***
## trade_type4                                3.66e-12 ***
## received_time_diff_last1(500,7.5e+04]       0.13293
## received_time_diff_last1(7.5e+04,4e+06]     0.13983
## received_time_diff_last1(4e+06, Inf]        0.01254 *
## trade_type_last13                           < 2e-16 ***
## trade_type_last14                          3.49e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                               edf Ref.df       F  p-value
## s(weight)                   1.000  1.000   2.144   0.1434
## s(current_coupon)           2.076  2.658   1.641   0.1431
## s(time_to_maturity)         2.221  2.814   1.652   0.1496
## s(trade_size)               4.757  5.776   8.533 9.35e-09 ***
## s(curve_based_price)        9.000  9.000  15.537  < 2e-16 ***
## s(trade_price_last1)        5.194  6.570 191.058  < 2e-16 ***
## s(trade_size_last1)         5.204  6.247   3.177   0.0038 **
## s(curve_based_price_last1) 7.974  8.773   5.497 5.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.98   Deviance explained =   98%
## GCV = 2.7747  Scale est. = 2.6901    n = 1620
```

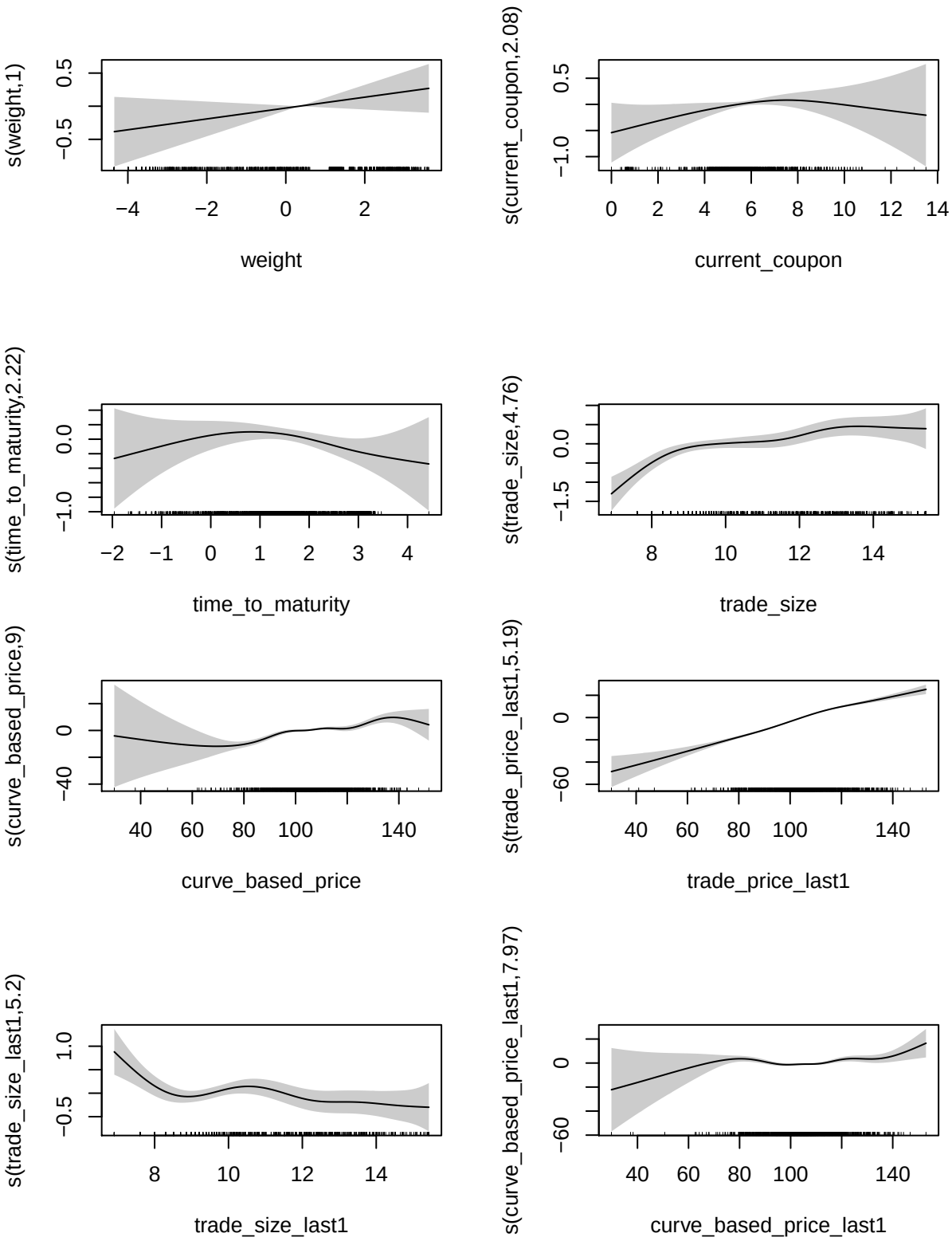
## 3

From the summary, we can see that the model predicts the mean difference in trade price between bonds whose current trade is of type "3" and a bond whose current trade is of type "4" to be 1.4828-0.7326=0.7502.

Therefore, trade price of type 3 is 0.7502 higher than type 4 on average.

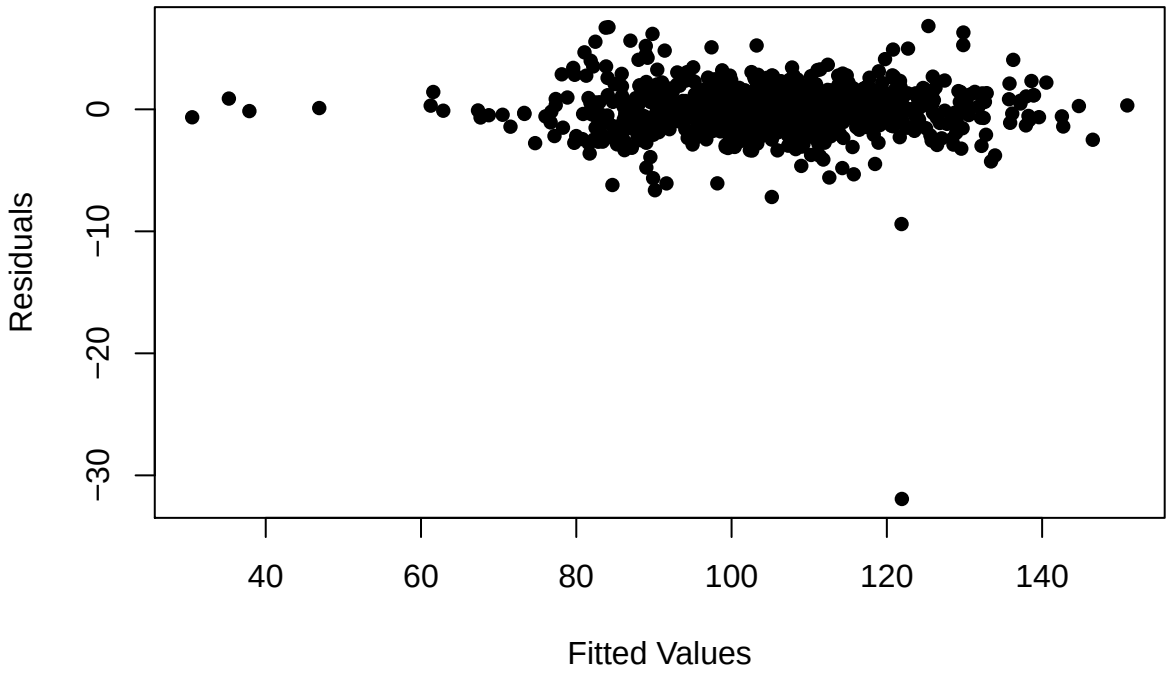
4

```
plot(holdgam, pages=2,scale=0,scheme=1)
```



### 5

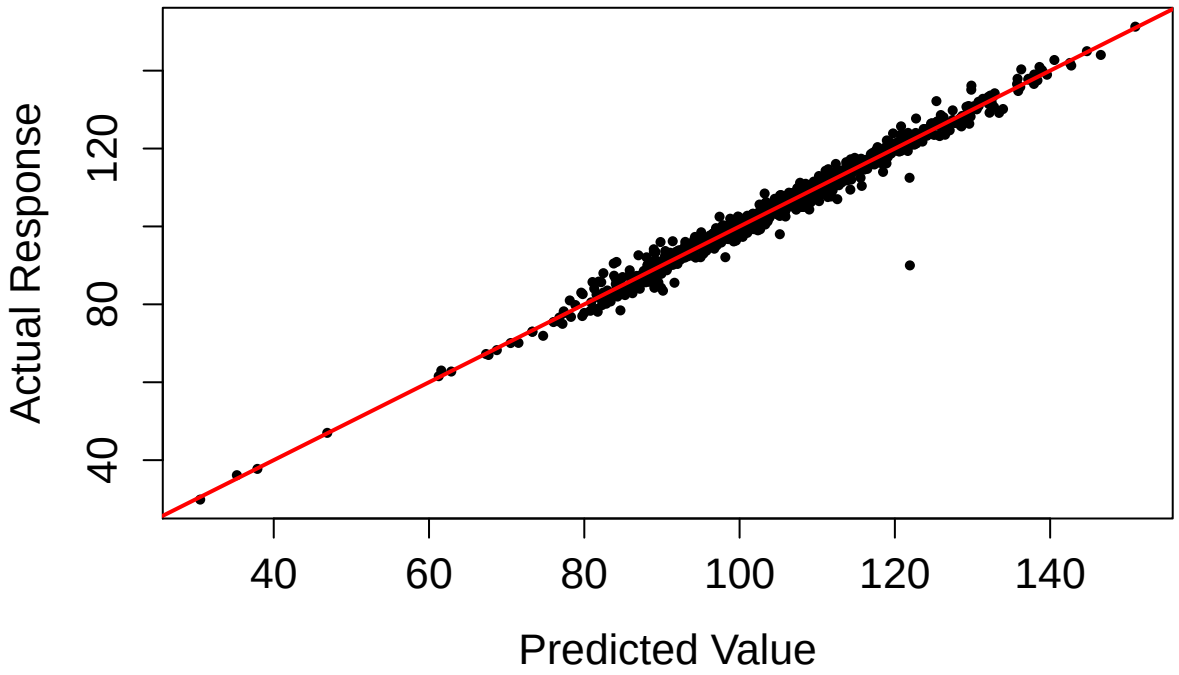
```
plot(holdgam$fit, holdgam$residuals,
     xlab="Fitted Values", ylab="Residuals",pch=16)
```




Quality of the fit: There is no prevalent pattern in the plot of residuals versus fitted values. However, there is one point for which the residual is quite extreme relative to others.

### 6

```
plot(predict(holdgam),bonddatasub$trade_price,pch=16,cex=0.7,xlab="Predicted Value",
     ylab="Actual Response", cex.axis=1.3,cex.lab=1.3)
abline(0,1,lwd=2,col=2)
```




Quality of the fit: almost all the points are close to the 45 degree line and they are evenly distributed on both sides of the line. However there is a point far from the line worth noticing.

**7**

```
# The alternative model
holdlinear = gam(trade_price ~ weight + current_coupon +
                   time_to_maturity + is_callable + reporting_delay +
                   trade_size + trade_type + curve_based_price +
                   received_time_diff_last1 + trade_price_last1 +
                   trade_size_last1 + trade_type_last1 + curve_based_price_last1,
                 data = bonddatasub)
#AIC of linear model
AIC(holdlinear)
```

```
## [1] 6357.764
```

```
# AIC of gam model:
AIC(holdgam)
```

```
## [1] 6251.106
```

AIC of the gam model is smaller than that of linear model. So the extra complexity is justified since it makes AIC smaller.