Assignment 4- ggplot with Splines

Andrew Abisha Hu, Elaine Su, Eric Xiong, Tianye Wang

9/6/2018

Data on projects raising funds on Kickstarter.com are recorded. Download kickstarter.csv on Canvas and answer the following questions. Here is the description of variables:

```
blurb: word counts of the blurb on the project page
content: word counts of the content on the project page
pledged: $ pledged at the end of the fund-raising period
category: the category to which the project belongs
Use only the Art category data to answer the following questions
```

Loading libraries and subset for art only:

```
library(ggplot2)
setwd("/Users/andrewhu/Documents/GitHub/Machine-Learning/Class Practice/Assignment 4- ggplot with Splin
ksart <- read.csv("KickStarter.csv")
ksart <- subset(ksart, category== "Art")
```

1. Create dummy variables c66-130, c131-225, c226up to indicate the four regions of content: 0- 65, 66-130. 131-225, and 226 up. Regress pledged on these dummy variables. What is the estimated regression equation? (provide R codes, 0.5%) Hint: you can use ifelse() to create dummy variables

```
attach(ksart)
ksart$c66_130 <- ifelse(content %in% c(66:130),1,0)
ksart$c131_225 <- ifelse(content %in% c(131:225),1,0)
ksart$c266up <- ifelse(content >= 226,1,0)
fit= lm(pledged~c66_130 + c131_225 +c266up, ksart)
coef(fit)
```

```
## (Intercept)     c66_130    c131_225      c266up
##     832.805    1007.638    2003.430    2991.924
```

2. On average, what is the difference in pledged between projects with content in 0-65 words and projects with content in 131-225 words? (0.5%)

```
2003.430
```

3. Now use cut() to break content into four bins as described in question (1). Regress pledged on the bins. (provide R codes 0.5%) Hint: you should get the same estimation results from question (1).

```
ksart$bin = cut(content, breaks= c(0,65,130,225,Inf))
fit2 = lm(pledged~bin, ksart)
coef(fit2)
```

```
##  (Intercept)  bin(65,130] bin(130,225] bin(225,Inf]
##      832.805     1007.638     2003.430     2991.924
```

4. Now use I() to specify the range of each bin without creating new variables nor using cut().Regress pledged on the bins. (provide R codes, 0.5%) Hint: You should get the same estimation results from question (1)

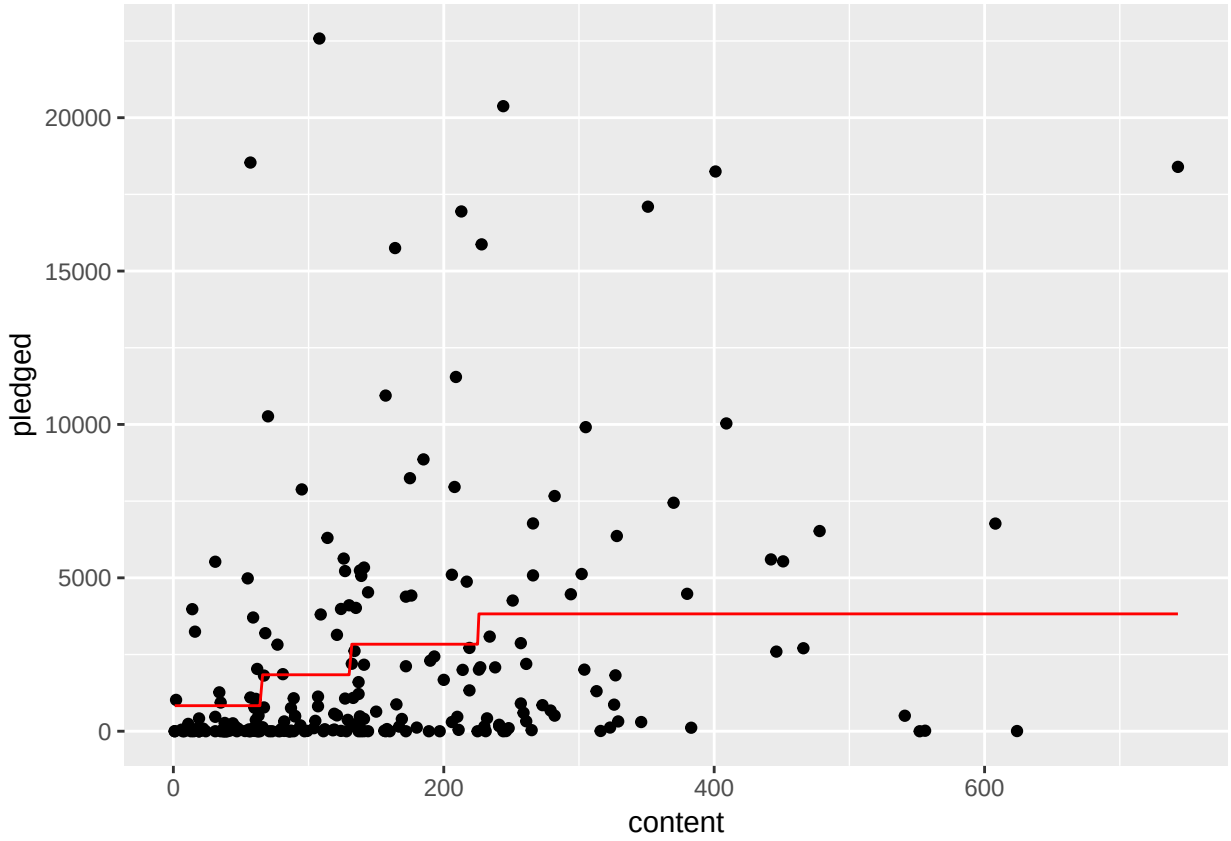
```
fit3= lm(pledged~I(content %in% c(66:130))+ I(content %in% c(131:225)) + I(content>=226))
coef(fit3)
```

```
##                     (Intercept)  I(content %in% c(66:130))TRUE
```

```
##                          832.805                          1007.638
## I(content %in% c(131:225))TRUE          I(content >= 226)TRUE
##                         2003.430                          2991.924
```

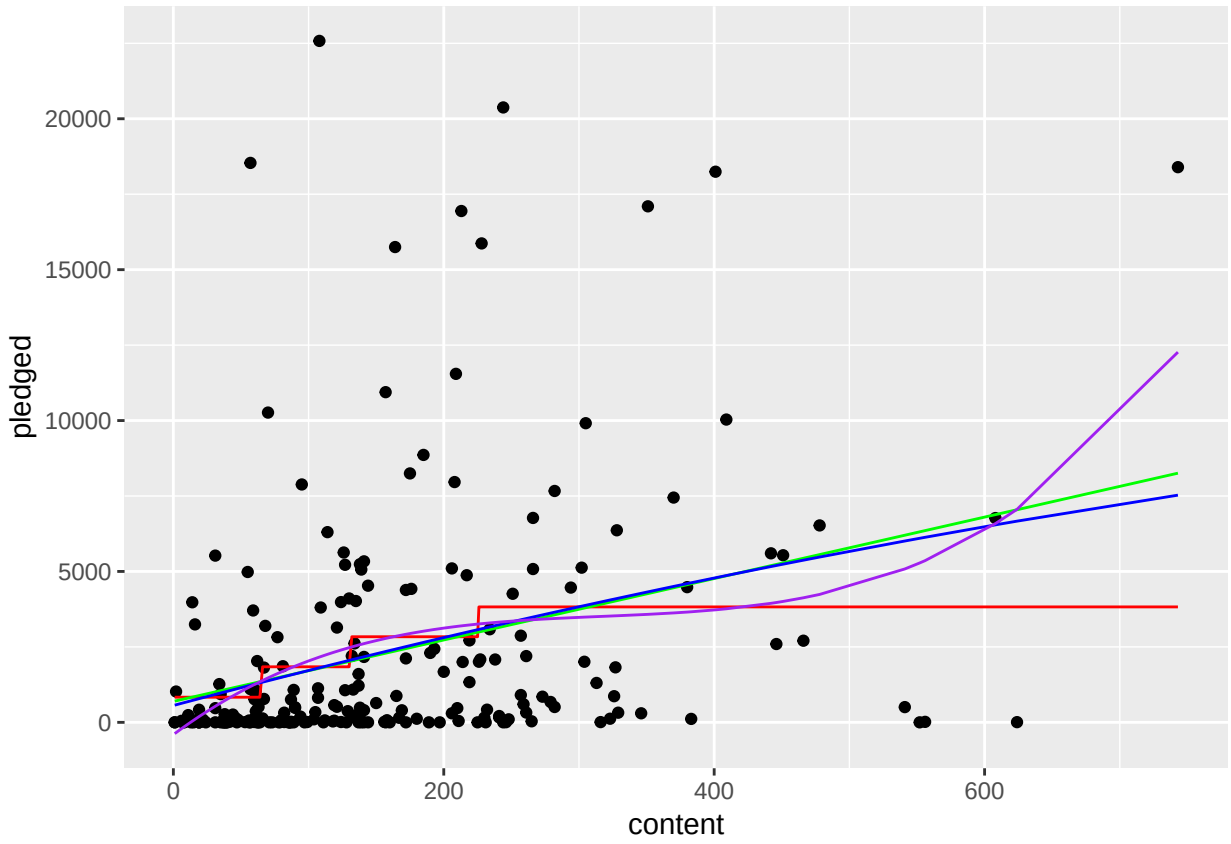
5. Create a scatter plot of content and pledged. Add a "red" line that shows predicted pledged for projects with content in 0 to 750 words by the model in question (4).

```
predict = predict(fit3, ksart[content<=750,])
g<- ggplot(ksart, aes(x=content, y=pledged))
g + geom_point() + geom_line(aes(x=content, y=predict),col=2)
```



6. Now create polynomial regression models to regress pledged on content from degree 1 to degree 3. Add lines that show predicted pledged for projects with content in 0 to 750 words by the three models to the scatter plot. Choose "green" for degree 1 model, "blue" for degree 2, and "purple" for degree 3.

```
fit4 = lm(pledged~content, ksart)
fit5 = lm(pledged~poly(content,2), ksart)
fit6 = lm(pledged~poly(content,3), ksart)

predict4 =predict(fit4,ksart[content<=750,])
predict5 =predict(fit5,ksart[content<=750,])
predict6 =predict(fit6,ksart[content<=750,])

g+ geom_point() + geom_line(aes(x=content, y=predict),col=2) + geom_line(aes(x=content, y=predict4),col=
```



7. Comment on how the four lines/models visually fit the data. Which one do you prefer? Why? (0.5%)

The purple line, because it is more flexible. The purple line starts from 0 at the beginning and is also the closest to outliers. Overall the purple line fits the dots better than any other model

8. Create a new variable h so that : h=(content – 130)^3 if content > 130 or h=0 if content <=130. Run a polynomial regression model of degree 3 to regress pledged on content with the predictor h. (provide R codes, 0.5%) Hint: check ifelse() for creating h.

```
ksart$h = ifelse(content>130, (content-130)^3,0)
fit7= lm(pledged~content + I(content^2) + I(content^3)+ h, ksart)
```

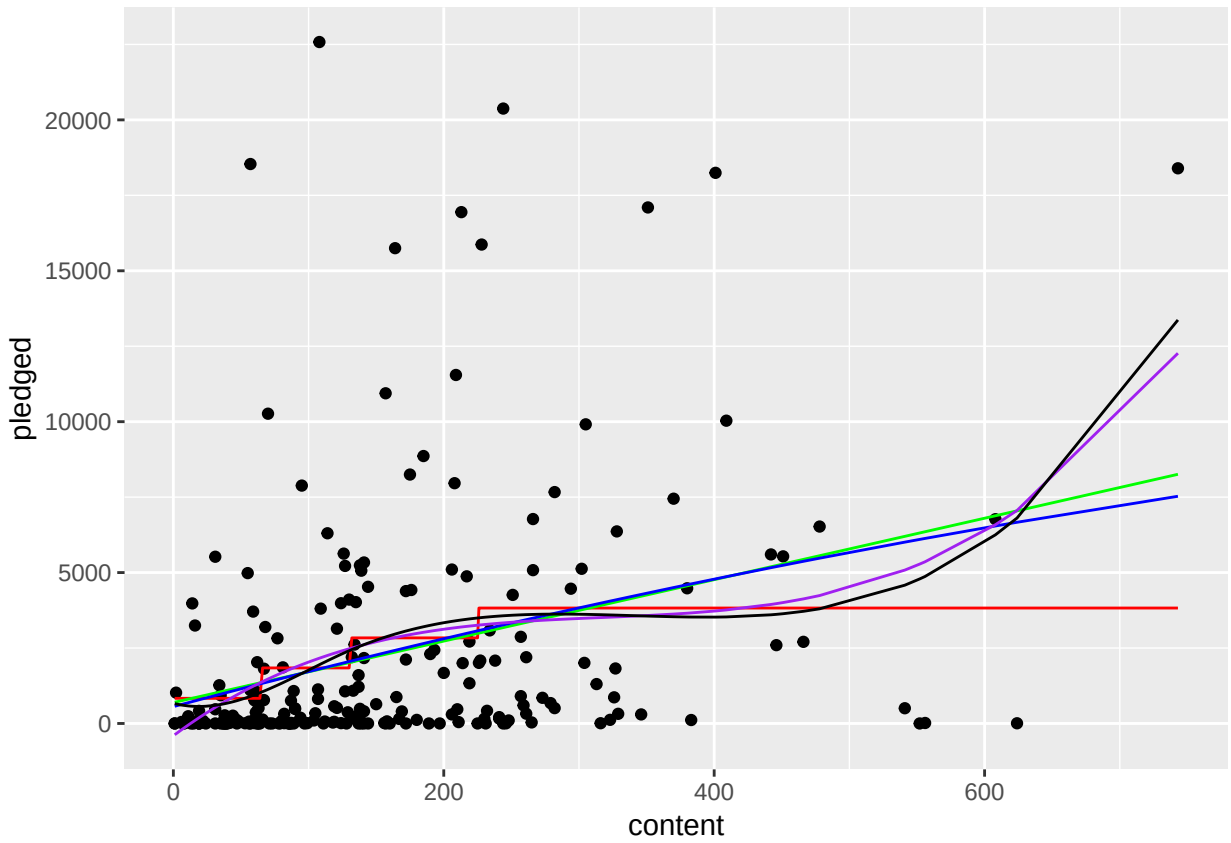
9. h in question (8) is in fact a truncated power basis function with a knot at 130, and the model is in fact a cubic spline with one knot. Rather than creating a new variable h, you can also use I() to add the truncated power basis function in your model. Try using only I() to create all predictors and run the model again. Add a "Black" line that shows predicted pledged for projects with content in 0 to 750 words by the model to the scatter plot. (provide R codes, 0.5%)

```
fit8= lm(pledged~ content + I(content^2) + I(content^3)+ I((content>130)*(content-130)^3),ksart)
summary(fit8)
```

```
##
## Call:
## lm(formula = pledged ~ content + I(content^2) + I(content^3) +
##     I((content > 130) * (content - 130)^3), data = ksart)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6801.0 -2087.6  -822.2   581.8 20632.9
```

```
##
## Coefficients:
##                                       Estimate Std. Error t value
## (Intercept)                          6.692e+02  1.202e+03   0.557
## content                             -1.158e+01  4.148e+01  -0.279
## I(content^2)                         3.379e-01  3.909e-01   0.864
## I(content^3)                        -1.121e-03  1.090e-03  -1.029
## I((content > 130) * (content - 130)^3)  1.279e-03  1.137e-03   1.125
##                                     Pr(>|t|)
## (Intercept)                            0.578
## content                                0.780
## I(content^2)                           0.388
## I(content^3)                           0.305
## I((content > 130) * (content - 130)^3)    0.262
##
## Residual standard error: 3874 on 218 degrees of freedom
## Multiple R-squared:  0.1256, Adjusted R-squared:  0.1096
## F-statistic: 7.828 on 4 and 218 DF,  p-value: 6.509e-06
```

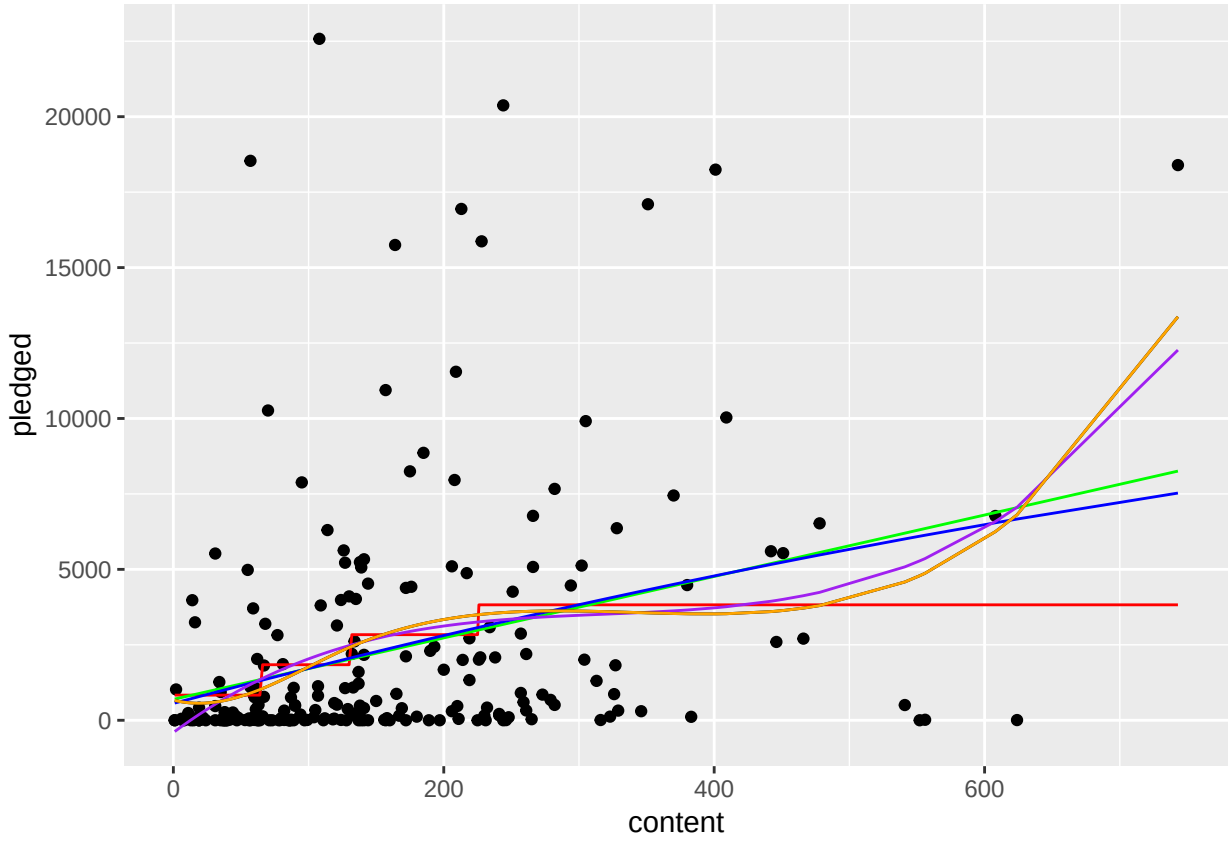
```
predict8 = predict(fit8,ksart[content<=750,])

g+ geom_point() + geom_line(aes(x=content, y=predict),col=2) + geom_line(aes(x=content, y=predict4),col=
```



10. Use the Splines package with bs() function to regress pledged on content with a cubic spline with one knot at 130. Add an "orange" line that shows predicted pledged for projects with content in 0 to 750 words by the model to the scatter plot. (provide R codes, 0.5%) Hint: You should find the orange line is exactly the black line, i.e., orange is the new black, because this model is equivalent to the models in questions (8) and (9). However, the coefficients are not the same because the predictors have been

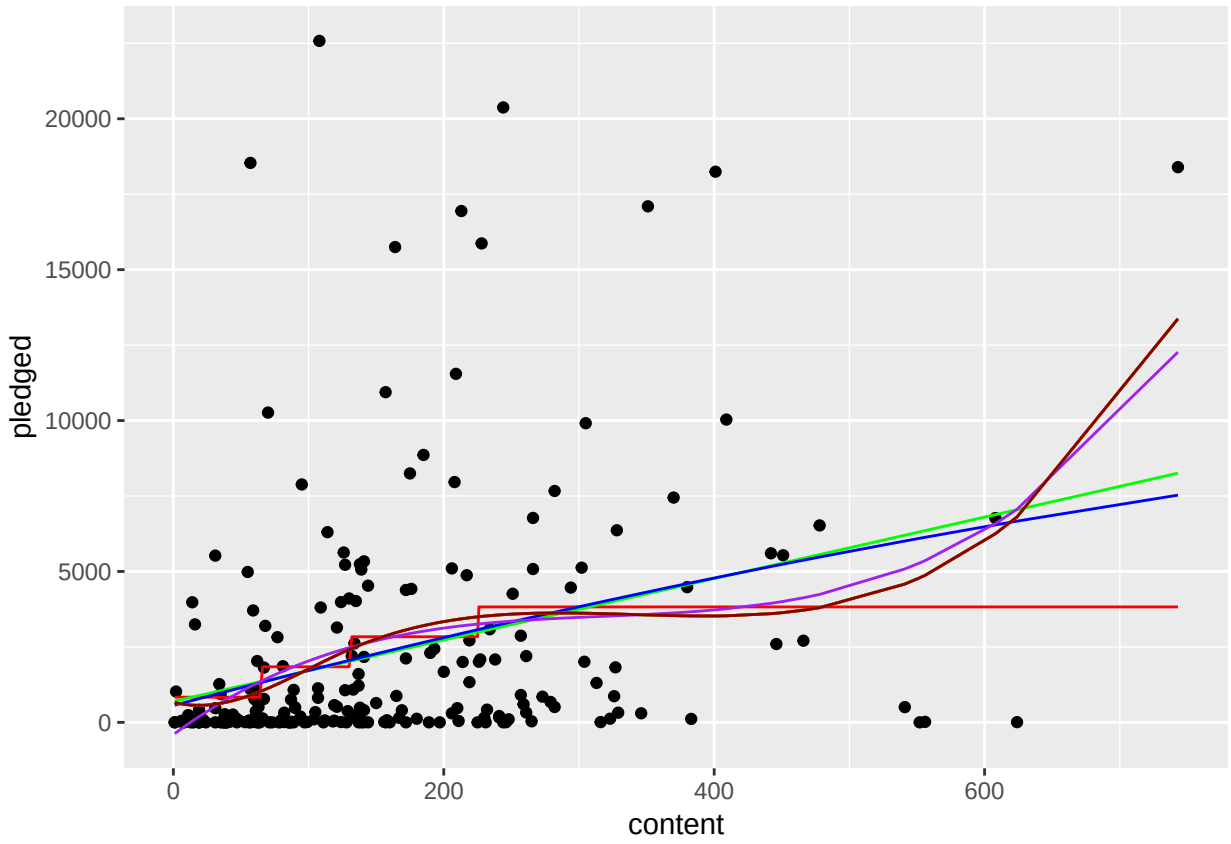
transformed when using bs().

```
library(splines)
fit9 = lm(pledged~ bs(content, knots=130, degree=3),ksart)
predict9 = predict(fit9, ksart[content<=750,])

g+ geom_point() + geom_line(aes(x=content, y=predict),col=2) + geom_line(aes(x=content, y=predict4),col=
```



11. Instead of specify the location of the knot, you can specify df (degree of freedom) and let the program select the locations of the knots for you. Use bs() to regress pledged on content by specifying df=4 (i.e., use only one knot because df – degree = 1). Add a "darkred"line that shows predicted pledged for projects with content in 0 to 750 words by the model to the scatter plot. (provide R codes, 0.5%)

```
fit10= lm(pledged~ bs(content, df=4), ksart)
predict10 = predict(fit10, ksart[content<=750,])

g+ geom_point() + geom_line(aes(x=content, y=predict),col=2) + geom_line(aes(x=content, y=predict4),col=
geom_line(aes(x=content, y=predict10),col="darkred")
```

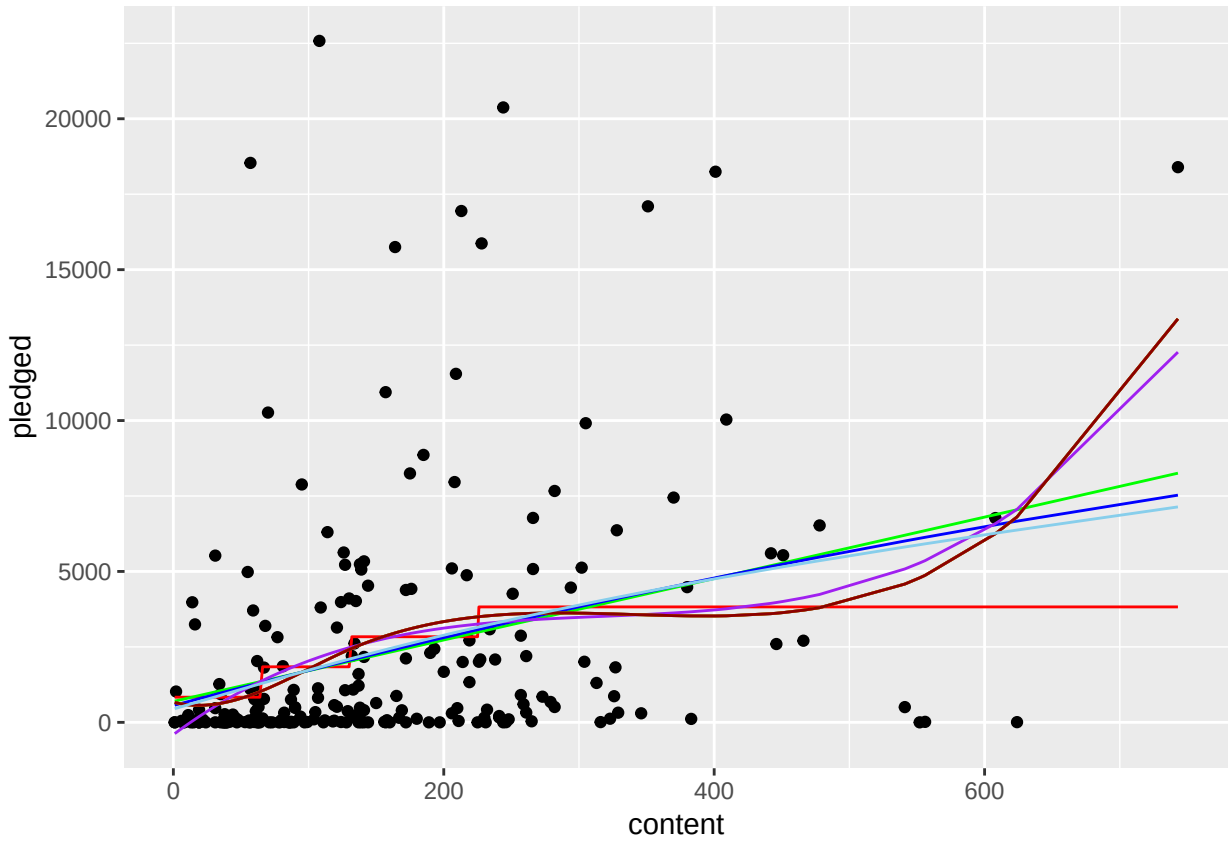


12. Are the dark red line and the orange line the same line? Why? Use bs(art$content, df=4) part to find the location of the knot and answer the question. (0.5%)

    Yes because they are essentially the same function with the same knot. We can see from the result that attr(,"knots")=130, which is the same location as the knot we specified in the previous question.

13. Now use ns(), i.e., natural smoother, rather than bs() to regress pledged on content with one knot at 130. Add a "skyblue" line that shows predicted pledged for projects with content in 0 to 750 words by the model to the scatter plot. Comment on the difference between the results of bs() and ns(). (provide R codes, 0.5%)

```
fit11 = lm(pledged~ns(content,knots=130), ksart)
predict11 = predict(fit11, ksart[content<= 750,])

g+ geom_point() + geom_line(aes(x=content, y=predict),col=2) + geom_line(aes(x=content, y=predict4),col=
geom_line(aes(x=content, y=predict10),col="darkred")+ geom_line(aes(x=content, y=predict11),col="skyblu
```



14. Now use the gam package and s() to try smoothing splines. Regress pledged on content with four degree of freedom. Based on the F test in the ANOVA tables, is it preferred to use a non-linear specification than a linear one for content given the 0.05 significance level? (provide R codes, 0.5%)

```
library(gam)
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.15
```

```
fit12= gam(pledged ~ s(content,4), data=ksart)
summary(fit12)
```

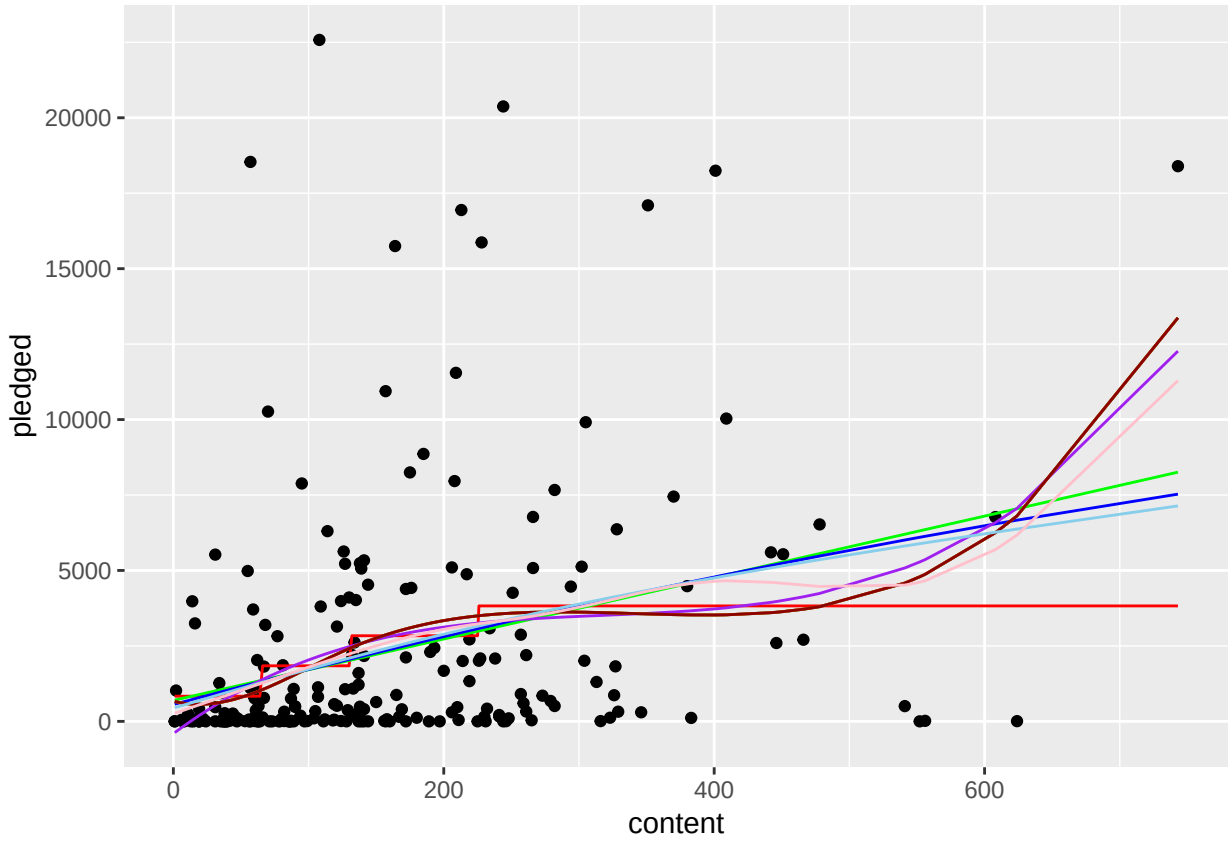
```
##
## Call: gam(formula = pledged ~ s(content, 4), data = ksart)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -6166.8 -1935.6  -914.7   508.0 20663.6
##
## (Dispersion Parameter for gaussian family taken to be 14807496)
##
##     Null Deviance: 3740888539 on 222 degrees of freedom
## Residual Deviance: 3228033760 on 218 degrees of freedom
## AIC: 4321.663
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                Df     Sum Sq   Mean Sq F value    Pr(>F)
```

```
## s(content, 4)   1  383510679 383510679    25.9 7.756e-07 ***
## Residuals     218 3228033760  14807496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##               Npar Df Npar F   Pr(F)
## (Intercept)
## s(content, 4)       3 2.9117 0.03536 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the summary results, we can see that the F-statistic for the parametric effect is 25.9, which is larger than the F-statistic for the nonparametric effect (2.9117). We can thus conclude that the linear specification is preferred over the non-linear one

15. Add a "pink" line that shows predicted pledged for projects with content in 0 to 750 words by the model with a smoothing spline to the scatter plot. Comment on the pattern of the line. (provide R codes, 0.5%)

```
predict12 = predict(fit12, ksart[content<=750,])

g+ geom_point() + geom_line(aes(x=content, y=predict),col=2) + geom_line(aes(x=content, y=predict4),col=
geom_line(aes(x=content, y=predict10),col="darkred")+ geom_line(aes(x=content, y=predict11),col="skyblu
```



16. Now expand the model in question (14) by adding the linear effect of blurb. Use an F-test to check whether the expand model with blurb fit the data better than the reduced model without blurb, given the 0.05 significance level. What is your conclusion? (provide R codes, 0.5%)

```
fit13= gam(pledged~s(content,4) +blurb , data=ksart)
summary(fit13)
```

```
##
## Call: gam(formula = pledged ~ s(content, 4) + blurb, data = ksart)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -5716.0 -2000.5  -895.5   565.2 20651.4
##
## (Dispersion Parameter for gaussian family taken to be 14741290)
##
##     Null Deviance: 3740888539 on 222 degrees of freedom
## Residual Deviance: 3198859571 on 217 degrees of freedom
## AIC: 4321.639
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                Df     Sum Sq   Mean Sq F value    Pr(>F)
## s(content, 4)   1  383510679 383510679 26.0161 7.374e-07 ***
## blurb           1   26741692  26741692  1.8141    0.1794
## Residuals     217 3198859571  14741290
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##               Npar Df Npar F   Pr(F)
## (Intercept)
## s(content, 4)       3 3.1649 0.02539 *
## blurb
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the F-test, the linear effect of blurb is not significant in the expanded model, including parametric and non-parametric effect. Therefore, the reduced model (without blurb) is better than the expand model.

17. Instead of expanding the model by adding the linear effect of blurb, you decided to try adding smoothing splines for blurb with four degree of freedom. Based on the F test in the ANOVA tables, is it preferred to use a non-linear specification than a linear one for blurb given the 0.05 significance level?

```
fit14= gam(pledged~s(content,4)+s(blurb,4),data=ksart)
summary(fit14)
```

```
##
## Call: gam(formula = pledged ~ s(content, 4) + s(blurb, 4), data = ksart)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -5561.6 -1946.9  -955.6   424.6 20170.3
##
## (Dispersion Parameter for gaussian family taken to be 14736056)
##
##     Null Deviance: 3740888539 on 222 degrees of freedom
## Residual Deviance: 3153515875 on 214 degrees of freedom
## AIC: 4324.455
```

```
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                Df     Sum Sq   Mean Sq F value    Pr(>F)
## s(content, 4)   1  386046130 386046130  26.197 6.851e-07 ***
## s(blurb, 4)     1   26952525  26952525   1.829    0.1777
## Residuals     214 3153515875  14736056
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##               Npar Df Npar F   Pr(F)
## (Intercept)
## s(content, 4)       3 3.1417 0.02621 *
## s(blurb, 4)         3 1.0611 0.36654
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

18. You can also use smooth.spline() to fit a cubic smoothing spline considering only one predictor. Regress pledged on content and set cv=TRUE to use leave-one-out cross validation to select the degree of freedom. What is the degree of freedom selected by the model?

```
fit15= smooth.spline(pledged~content,cv=T)
```

```
## Warning in smooth.spline(pledged ~ content, cv = T): cross-validation with
## non-unique 'x' values seems doubtful
```

```
fit15$df
```

```
## [1] 10.14541
```